Security and Communication Networks

Security and Privacy Protection for 5G-Enabled Internet of Things

Lead Guest Editor: Kwok-Yan Lam
Guest Editors: Xun Yi and Huaxiong Wang

Security and Privacy Protection for 5G-Enabled Internet of Things

Security and Communication Networks

Security and Privacy Protection for 5G-Enabled Internet of Things

Lead Guest Editor: Kwok-Yan Lam
Guest Editors: Xun Yi and Huaxiong Wang

Chief Editor

Associate Editors

Academic Editors

Contents

Hindawi
Security and Communication Networks
Volume 2022, Article ID 7729456, 11 pages
https://doi.org/10.1155/2022/7729456

Research Article

A Threat Intelligence Analysis Method Based on Feature Weighting and BERT-BiGRU for Industrial Internet of Things

Jingchen Yan, Zhe Du, Jifang Li, Shiduo Yang, Jinghao Li, and Jianbin Li

North China Electric Power University, School of Control and Computer Engineering, Beijing. 102206, China

Correspondence should be addressed to Jianbin Li; lijb87@ncepu.edu.cn

Received 9 April 2021; Revised 2 December 2021; Accepted 10 January 2022; Published 25 February 2022

Academic Editor: Xun Yi

The combination of 5G technology and the industrial Internet of things (IIoT) makes it possible to realize the interconnection of all things. Still, it also increases the risk of attacks such as large-scale DDoS attacks and IP spoofing attacks. Threat intelligence is a collection of information causing potential and nonpotential harm to the industrial Internet. Extracting network security entities and their relationships from threat intelligence text and constructing structured threat intelligence information are particularly important for IIoT security protection. However, threat intelligence is mostly text reports, which means the value information needs to be extracted manually by security analysts, and it is highly dependent on personnel experience. Therefore, this study proposes an IIoT threat intelligence analysis method based on feature weighting and BERT-BiGRU. In this method, BERT-BiGRU is used to classify attack behavior and attack strategy. Then, the attack behavior is weighted to make the classified result more accurate according to the relationship between attack strategy and attack behavior in ATT&CK for ICS knowledge. Finally, the possibility of attack and the harm degree of attack are calculated to form the threat value of the attack. The security analysts can judge the emergency response sequence by the threat value to improve the accuracy and efficiency of emergency response. The results indicate that the proposed method in this study is more accurate than the other standard methods and is more suitable for the unstructured threat intelligence analysis of IIoT.

1. Introduction

The developing application of 5G technology [1] improved the communication quality and made it possible to enable the perception and interconnection of infrastructure, personnel, and their environment. However, the interconnection between the network and external devices brought new threats in various angles and forms. Not only does the "threat surface" of external attacks become more extensive, but also the probability of equipment failures, software defects, and user errors increases, all of these will have a tremendous negative impact on the operation of the system. The blackout caused by the BlackEnergy Malware in the Ukrainian power grid [2] and the large-scale blackout in Venezuela [3] are two notable examples. According to the US Securities and Exchange Commission report and European financial report, the loss caused by an industrial infection in 2017 is as high as 1 billion dollars.

Facing the increasingly severe security situation, a new network security defense mechanism driven by threat intelligence has emerged. In 2013, Gartner proposed the concept of threat intelligence (TI) [4], which includes scenarios, tools, indicators, inferences, and feasible suggestions. It is evidence-based knowledge about the existing or emerging threats faced by assets, providing a decision-making basis for threat response [5]. Based on what was mentioned above, the threat intelligence contains detailed information about current or upcoming network security threats, which can help enterprises mine and analyze the attack behavior [6] and implement active network defense against network security threats.

At present, most of the threat intelligence in the industrial field provided by security companies [7, 8] is in an unstructured format, so it is difficult for security analysts and organizations to obtain standardized and structured threat information. Moreover, the extracted attack information does not include a threat value, so it is difficult to provide accurate and effective emergency response countermeasures for IIoT security situational awareness systems or other defense mechanisms. Therefore, effective extraction of the valuable information of threat intelligence and converting it into a standardized and structured form are essential and significant for practical applications and fundamental research on the security of IIoT.

To solve the problems above, this study proposes a threat intelligence analysis method based on feature weighting and BERT-BiGRU for IIoT. Firstly, according to the matrix knowledge of ATT&CK for ICS, a large amount of threat intelligence was collected and standardized [9]. Secondly, a multilabel classification model was built based on the BERT-BiGRU model to identify the attack behavior of threat intelligence. Then, all the attack behaviors were weighted based on the dependence between strategy labels and behavior labels so that more accurate results of attack behavior identification were obtained. Finally, the attack behavior risk index was measured to form the attack behavior threat value.

Through this method, the attack behavior extraction of the threat intelligence of the IIoT is realized. By sorting the threat degree of the attack behavior, it provides a reference for emergency response and disposal, improving the security of the IIoT.

This study is organized as follows: in Section 2, the relevant research works are introduced. In Section 3, we describe the proposed method based on feature weighting and BERT-BiGRU in detail. In Section 4, the experiments are conducted, and the results are analyzed. Finally, we conclude the work and give the prospects for future research.

2. Related Work

2.1. Threat Intelligence Analysis. Threat intelligence analysis extracts unstructured data such as security warning notification, vulnerability notification, and threat notification from threat intelligence using natural language processing technology and helps the attacked analyze the behavior and vulnerabilities exploited by the attackers so that the attacked can make emergency defense decisions promptly.

Gao and Fan [10] used a graph database to analyze threat intelligence, indicated their properties and association relationship of industrial Internet security vulnerability data effectively and intuitively, and realized in-depth analysis and evaluation of vulnerability data. Wu et al. [11] proposed group tracer to automatically extract the TTP curve, to dig out behind the complex attack and potential attackers through the combination of network attack behavior threat intelligence knowledge. Liu et al. [12] analyzed the attack behavior events through threat intelligence and correlated the similar behavior according to the direction of the attack events to investigate the attack stage and protect it. Zhang et al. [13] proposed the EX-Action framework for extracting threat behavior from CTI reports. Ex-Action could detect threat actions using natural language processing (NLP) technology and identify threat actions using a multimodal learning algorithm. Zhang et al. [14] proposed a prediction method SIoT account malicious behavior based on threat intelligence. It used SVMs to obtain the threat intelligence related to the target account's negative behavior. It analyzed the contextual data in the threat intelligence to predict the behavior of the malicious version. Preuveneers et al. [15] proposed the security enhancement framework of TATIS to timely respond to new vulnerabilities and attack forms in network attacks via threat intelligence analysis. Hinne [16] established a joint analysis model in network attack events and threat intelligence to analyze the attacker's motive and exploit the vulnerability, steps, and specific actions. In the process of event response, the attacker status is updated in real time, and decision analysis is provided.

However, the above methods do not provide comprehensive guidance to security analysts. In the face of an attack, it is difficult for security analysts to determine the emergency response sequence, resulting in heavy losses.

2.2. Multilabel Classification Analysis. A noticeable problem exists to automatically extract threat behavior from cyber threat intelligence reports in threat intelligence analysis. The threat intelligence contains various categories of data, such as threat behavior and attack stage. Thus, the threat behavior extraction problem can be abstracted into a multilabel classification problem.

Multilabel classification refers to separately analyzing the task text data with multiple labels. The calculation of multilabel classification tasks is more complicated than that of traditional classification tasks. It is mainly reflected in that the text features of a sample need to be associated with multiple labels, which require more advanced feature extraction and correct mapping to the corresponding labels. However, due to the complexity of data expression and the exponentiality of the label output space, the research on multilabel classification is still limited.

The current research mainly focuses on problem conversion (considering the labels are independent, converting the problem into two (multiple)) and algorithm adaptation (adapting the learning model to cope with the multilabel classification task) of these two aspects. Bernhard et al. [17] proposed a chain binary classification model to model the high-order association between labels. Yen et al. [18] proposed PDSparse to learn a separate linear classifier for each label. In the training process, all the positive labels and a small amount of active negative labels of each training sample can be distinguished by classifier via optimizing the label distribution. Yang et al. [19] proposed a labeled implicit Dirichlet model based on subdividing the data to reduce the time complexity of the multilabel classification algorithm. Tan and Liu [20] used the K-nearest neighbor graph to segment the relationship before the text label as a weakly supervised method. Then, the maximized posterior probability of the label value was utilized to construct a multilabel classification model, resulting in predicting the new label. Prabhu and Varma [21] optimized the nDCG algorithm to learn the structure pattern of the tree in the feature space dimension, then trained a binary classifier for each internal node, and, finally, predicted the label distribution of a given instance. Literature [22] used CNN and RNN to capture the inner relationship of local and global semantic feature modeling labels. Xiao et al. [23] designed the LSAN model to determine the semantic connection between labels and documents using label semantic information. The self-attention mechanism is used to capture label data, and a label-specific document feature representation is constructed. Wehrmann et al. [24] proposed a multilayer output neural network model for multilabel classification; this structure has an output layer at each hierarchical level and provides a global output layer for the entire network to track the label dependency in the hierarchy as a whole by optimizing the sum of the global and each level of a loss function.

3. Threat Intelligence Analysis Method for IIoT Based on Feature

3.1. Weighting and BERT-BiGRU

3.1.1. Basic Notions

(1) Threat Intelligence. Threat intelligence [4] is a collection of information that can cause potential and non-potential harm to an enterprise. Threat intelligence describes attack events and attack behavior.

(2) Attack Event. IIoT attack event refers to a security threat event causing potential harm to the system or damage to system assets through various technical means. Usually, attackers use configuration defects, protocol defects, program defects, or violent attacks to attack the IIoT.

(3) Attack Behavior. Attack behavior refers to an action performed by an attacker to achieve a goal or gain some resources. Any attack event on the IIoT is composed of a series of attack behavior, based on which the whole process of an attack event can be depicted entirely. For example, the attacker can obtain the target system's TCP/IP subnet mask information by "network connection enumeration," and the "network connection enumeration" is recognized as an attack behavior.

(4) ATT&CK for ICS. ATT&CK for ICS [9] is a model and knowledge base reflecting the attack behavior of the industrial control system in each attack life cycle. It consists of three parts: strategy, technology, and process. The design represents what the attacker tries to achieve. Technology and process represent the behavior performed by the attacker to achieve the goal. At present, ATT&CK for ICS covers 11 attack strategies and 81 attack behaviors.

3.1.2. Overview of the Method. This study proposed a threat intelligence analysis method of IIoT based on feature weighting and BERT-BiGRU. The overview of the method is shown in Figure 1. Firstly, the threat intelligence data of IIoT on the open-source threat intelligence platform are collected, and the data preprocessing operations such as cleaning and denoising are completed; secondly, word segmentation and BERT sentence vector acquisition are carried out on the preprocessed data, and a multilabel classification model based on BERT-BiGRU is constructed. The attack strategy and attack behavior of the threat intelligence are classified and identified. Based on the recognition result, all the behavior labels were weighted based on the dependence of the strategy label and its internal behavior labels to obtain more accurate attack behavior recognition results. Finally, the attack risk indicators were measured to obtain the attack behavior threat value. The threat value of attack behavior represents the harm degree of attack behavior, providing a reference for emergency response and disposal.

3.1.3. Threat Intelligence Analysis Method

(1) Threat Intelligence Data Preprocessing. There are usually multiple data sources during the threat intelligence data collection of IIoT, including homogeneous or heterogeneous databases, file systems, and service interfaces. Different data sources generally have complementarity and difference in data integrity, accuracy, and representation format. Different data sources are generally complementary and different in data integrity, accuracy, and presentation format and are vulnerable to noise data, missing data values, data conflicts, etc. Therefore, the collected data sets need to be preprocessed to ensure the accuracy, consistency, and high quality of the data analysis results.

It can be found from Figure 2 that the threat intelligence data preprocessing in this study is divided into three stages: standardization, cleaning, and reduction in threat intelligence data:

> Standardization: the obtained data may have multiple structures and types. By the threat intelligence standard, we standardize the presentation of threat intelligence data from different sources. The concrete operation includes word root processing and morpheme processing. This process contributes to transforming these complex data into a single or manageable structure to achieve the goal of rapid analysis and processing.
>
> Cleaning: not all data in attack events are valuable. There are some negligible data or even some data that are completely wrong distractions. Therefore, it is necessary to use various verification methods to remove inaccurate data (word abbreviations, unusual spacing, nonword characters, and any non-computer-related terms) that hinder classification. This study uses the filtering method to extract valuable data and label the information with confidence.
>
> Reduction: this process is to merge the threat intelligence data. Feature reduction technology can reduce and simplify the size of the data set without compromising the accuracy of analysis results, which contributes to increasing the value density of the threat intelligence data. The feature reduction formulas are shown in

where α and β are, respectively, the set of measured values of two different types of features. n_1 and n_2 are the corresponding sample numbers. $SE(\alpha - \beta)$ is the variance of the

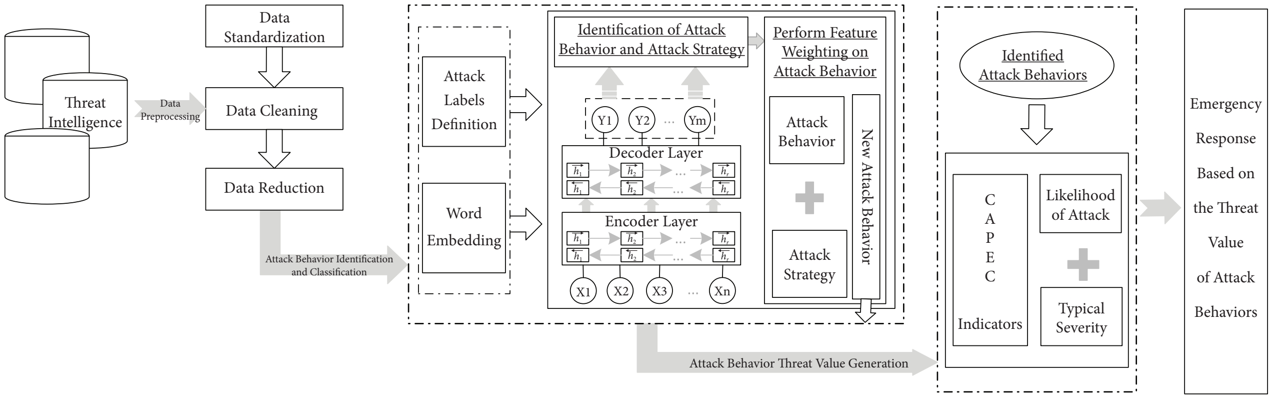

Threat Intelligence
Data Preprocessing
Data Standardization
Data Cleaning
Data Reduction
Attack Behavior Identification and Classification
Attack Labels Definition
Word Embedding
Identification of Attack Behavior and Attack Strategy
Y1
Y2
...
Ym
Decoder Layer
Encoder Layer
X1
X2
X3
...
Xn
Perform Feature Weighting on Attack Behavior
Attack Behavior
Attack Strategy
New Attack Behavior
Attack Behavior Threat Value Generation
Identified Attack Behaviors
C A P E C
Indicators
Likelihood of Attack
Typical Severity
Emergency Response Based on the Threat Value of Attack Behaviors

Figure 1: Overview of the method.

feature. The conflict of the feature is used to normalize the mean of the feature. The TEST function is built for comparison. As the deviation increases severely, the importance of this feature enhances. Otherwise, the importance of this feature decreases.

$$\mathrm{SE}(\alpha-\beta)=\sqrt{\frac{\mathrm{var}(\alpha)}{n_1}+\frac{\mathrm{var}(\beta)}{n_2}}, \qquad (1)$$

$$\text{TEST: } \frac{|\mathrm{mean}(\alpha)-\mathrm{mean}(\beta)|}{\mathrm{SE}(\alpha-\beta)} > \text{theshold\_value}, \qquad (2)$$

3.1.4. Attack Behavior Identification and Classification

(1) Identification of Attack Behavior and Attack Strategy. This study designed a multilabel classification model based on feature weighting and BERT-BiGRU for attack recognition, as shown in Figure 3. The classification model consists of the BERT model and the BiGRU model. The BERT model is only used to extract a sentence representation, whereas the BiGRU model is used to classify attack behavior and attack strategy in threat intelligence. Firstly, the preprocessed threat intelligence content is input into the BERT model, and the vector representation is performed after two pretraining tasks of the model. Subsequently, the vector representation fused with the full-text semantic information is output. Then, the output of the BERT model is input into the BIGRU model. The BIGRU model extracts the abstract features of threat intelligence through the fully connected (FC) layer by word vector mapping. It facilitates feature extraction by adding an attention mechanism before the FC layer to give a higher weight to essential attributes. To complete the multilabel classification task of attack behavior and attack strategy in threat intelligence, an FC layer and softmax need to be connected to the model to classify the deep semantic features of the threat intelligence text.

(2) Feature Weighting of Attack Behavior. According to the ATT&CK for ICS knowledge, an attack strategy connects many different attack behaviors, and there exists a dependency between attack strategy and attack behavior. For example, when the probability of an attack strategy increases, the probability of an attack behavior within the strategy will rise accordingly. The current attack threat can be dealt with more accurately by analyzing and extracting the relationship between attack strategy and attack behavior. As a result, based on the relationship between attack strategy and attack behavior, the attack behavior feature weighting method is designed in this study, and the critical steps of this method are shown in the following formulas:

$$\partial=\left(\frac{1}{\mathrm{EXP}}\left(\left(\mathrm{Labeled}-\mathrm{Tact}_{\Phi}\right)*(-1)\right)*\left(\frac{1}{10}\right)\right), \qquad (3)$$

$$\mathrm{Labeled}-\mathrm{Tech}_{\Delta}=\mathrm{Labeled}-\mathrm{Tech}_{\Delta}*(1+\partial). \qquad (4)$$

When a specific attack behavior occurs, its associated attack strategy must also exist. Analyzing attack strategy is more accessible than analyzing attack behavior, and the attack strategy analysis result is more accurate. The result of attack strategy identification is processed exponentially to optimize the analysis effect of attack behavior. ∂ is the value of attack strategy identification result $Labele\ d-Tact$ processed in exponential form. $Labele\ d-Tech$ is the feature weighting attack behavior identification result.

Based on what is mentioned above, the in-depth analysis of the threat intelligence data is successfully realized, which can output the structured attack behavior labels and the corresponding probability value with high accuracy and readability.

3.1.5. Generation of Attack Behavior Threat Value.

Based on the method above, the identification of the attack behavior in the threat intelligence is realized. However, it is difficult to judge the threat degree of the attack behavior, resulting in difficulty in priority warning to the attack with a higher harm degree. Therefore, it is necessary to consider the threat of attack behavior and quantify such important risk indicators, including the possibility of attack behavior and the harmful degree of attack behavior. Based on the data of Common Attack Pattern Enumeration and Classification (CAPEC) [25] published by MITRE, the possibility and harm degree of the attack behavior were calculated in this study, and the threat value of attack behavior was formed.

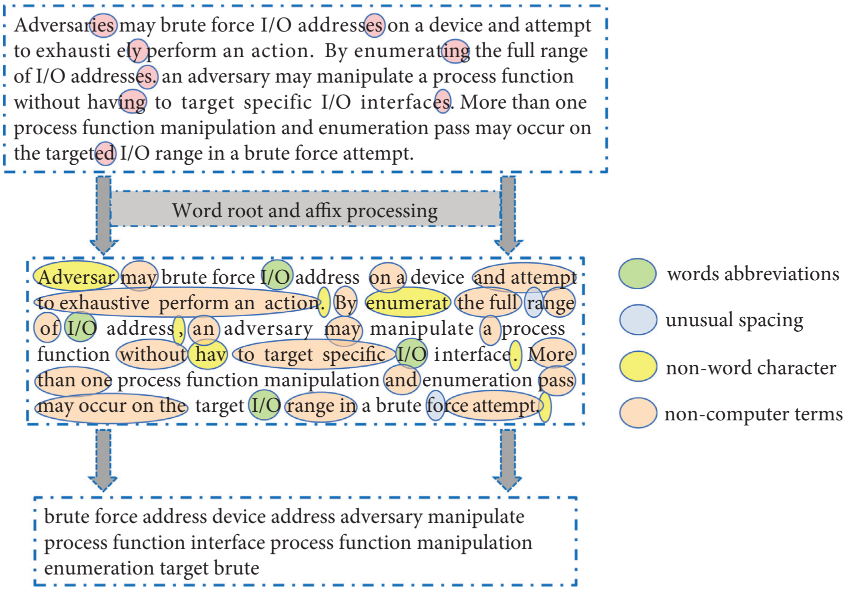

Adversaries may brute force I/O addresses on a device and attempt to exhausti ely perform an action. By enumerating the full range of I/O addresses, an adversary may manipulate a process function without having to target specific I/O interfaces. More than one process function manipulation and enumeration pass may occur on the targeted I/O range in a brute force attempt.
Word root and affix processing
Adversar may brute force I/O address on a device and attempt to exhaustive perform an action. By enumerat the full range of I/O address, an adversary may manipulate a process function without hav to target specific I/O interface. More than one process function manipulation and enumeration pass may occur on the target I/O range in a brute force attempt.
words abbreviations
unusual spacing
non-word character
non-computer terms
brute force address device address adversary manipulate process function interface process function manipulation enumeration target brute

Figure 2: Threat intelligence data preprocessing.

CAPEC is an enumeration and classification data set of attack types established by the United States Department of Homeland Security in 2007, a widely accepted and recognized public standard for attack modes, as shown in Table 1. CAPEC comprises two indicators of "likelihood of attack" and "typical severity." Both were classified into five levels, including "very low," "low," "medium," "high," and "very high." The "likelihood of attack" represents the probability of successful attack behavior. It considers relevant factors, including attack prerequisites, required attacker resources, and possible countermeasures. The "typical severity" aims to reveal the severity degree of the consequences of successful attack behavior.

The attack behavior labels in CAPEC and ATT&CK for ICS are the same. When we calculate the threat value of attack behavior, we firstly map the attack behavior in ATT&CK for ICS to the CAPEC, as shown in formula (5). Secondly, we quantify the unstructured CAPEC level labels to 1–5, as shown in formulas (6) and (7). Since the "typical severity" indicator is more valuable for attack defense, it is given a higher weight, as shown in formula (8). Then, the CAPEC indicator score and the attack behavior label classification result are combined to form the threat score of each attack behavior, as shown in formula (9).

$$\mathrm{Tech}_i, \mathrm{Tact}_i \longleftarrow \mathrm{analysisCAPECtext}, \tag{5}$$

$$\{x \mid x = 1, 2, 3, 4, 5\} \longleftarrow \mathrm{LikelihoodScore}, \tag{6}$$

$$\{y \mid y = 1, 2, 3, 4, 5\} \longleftarrow \mathrm{SeverityScore}, \tag{7}$$

$$Sc\,(\mathrm{CAPEC}) = \mathrm{LikelihoodScore} + \mathrm{EXP}\,(\mathrm{SeverityScore}), \tag{8}$$

$$Sc\,(\mathrm{Tech}_\Delta) = \mathrm{Labeled} - \mathrm{Tech}_\Delta * Sc\,(\mathrm{CAPEC})_\Delta. \tag{9}$$

The threat value of each attack behavior can be obtained based on the above steps. Security personnel can determine the corresponding emergency response sequence according to the threat value of the attack behavior.

4. Experiments and Analysis

4.1. Experiment 1: Demonstration of the Results of Proposed Method. The analysis of the Industroyer attack [26] is taken as an example. It is shown in Figure 4. Industroyer is a malicious program that could destroy the critical assets of the industrial control system. It invaded and attacked the Ukrainian power grid in 2016, causing a significant impact. It has been one of the biggest threats to the industrial control system since the "earthquake network virus" appeared, bringing about a large-scale power outage and property loss. Figure 4 shows an Industroyer attack event text on the IBM security platform, and Figure 5 shows the identification results obtained using the proposed method in this study. To enhance the readability of the identification results, the truth map was constructed using the Neo4j technology, as shown in Figure 6. After matching with the detailed list of Industroyer attacks provided by the MITRE platform and widely identified by security experts [27], the accuracy and recall values of the identification results using the present method are as high as 89.87% and 87.1%, which are much higher than those obtained using other methods.

4.2. Experiment 2: Comparative Experiment with Other Methods. In this section, we conduct three groups of comparative experiments, which are, respectively, as follows: 1. comparison between the present method and the BERT-BiGRU method without feature weighting; 2. comparison between the present method and the KNN and random

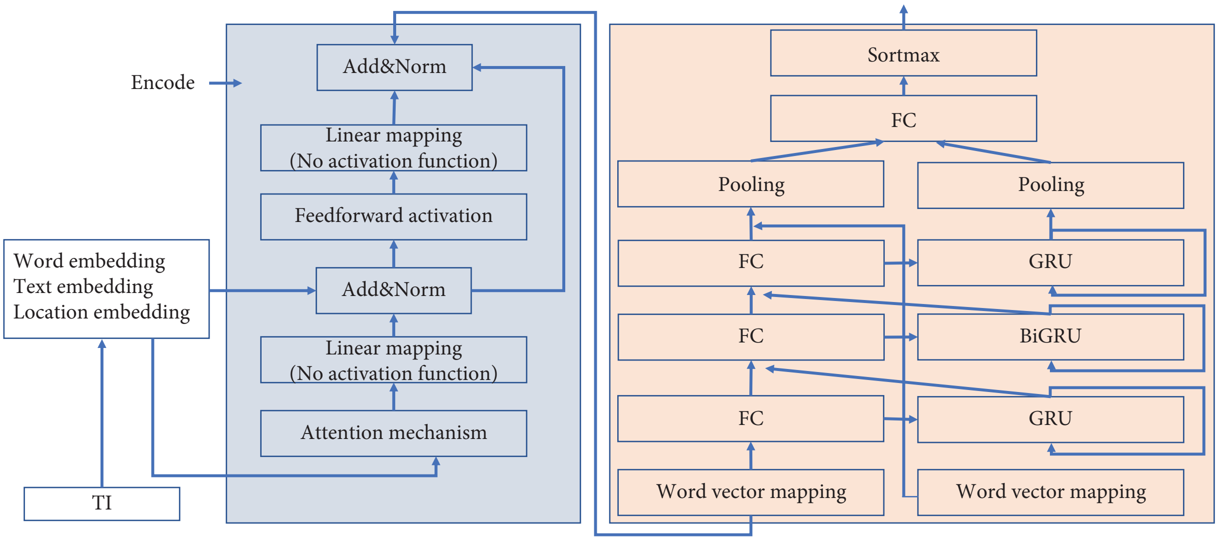
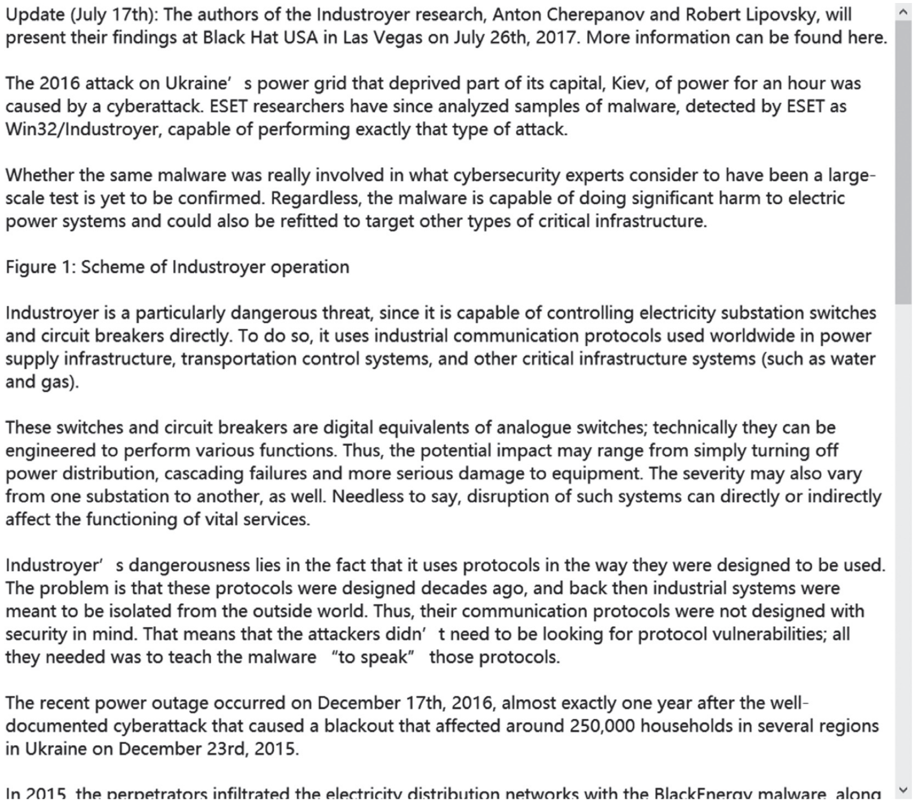


FIGURE 3: BERT-BiGRU model.

TABLE 1: Some attack types of CAPEC.

Attack	Likelihood of attack	Typical severity	Resources required
CAPEC-94: man-in-the-middle attack	High	Very high	Two components are communicating with each other. The communication occurs in clear (not encrypted) or with insufficient and deceivable encryption.
CAPEC-125: flooding	High	Medium	A script or program capable of generating more requests than the target can handle, or a network or cluster of objects can make simultaneous requests.
CAPEC-163: spear phishing	High	High	An adversary must have the ability to communicate their phishing scheme to the victims (via email and instant message), as well as a website or other platforms for victims to enter personal information into.
… …	… …	… …	… …



Update (July 17th): The authors of the Industroyer research, Anton Cherepanov and Robert Lipovsky, will present their findings at Black Hat USA in Las Vegas on July 26th, 2017. More information can be found here.

The 2016 attack on Ukraine’ s power grid that deprived part of its capital, Kiev, of power for an hour was caused by a cyberattack. ESET researchers have since analyzed samples of malware, detected by ESET as Win32/Industroyer, capable of performing exactly that type of attack.

Whether the same malware was really involved in what cybersecurity experts consider to have been a large-scale test is yet to be confirmed. Regardless, the malware is capable of doing significant harm to electric power systems and could also be refitted to target other types of critical infrastructure.

Figure 1: Scheme of Industroyer operation

Industroyer is a particularly dangerous threat, since it is capable of controlling electricity substation switches and circuit breakers directly. To do so, it uses industrial communication protocols used worldwide in power supply infrastructure, transportation control systems, and other critical infrastructure systems (such as water and gas).

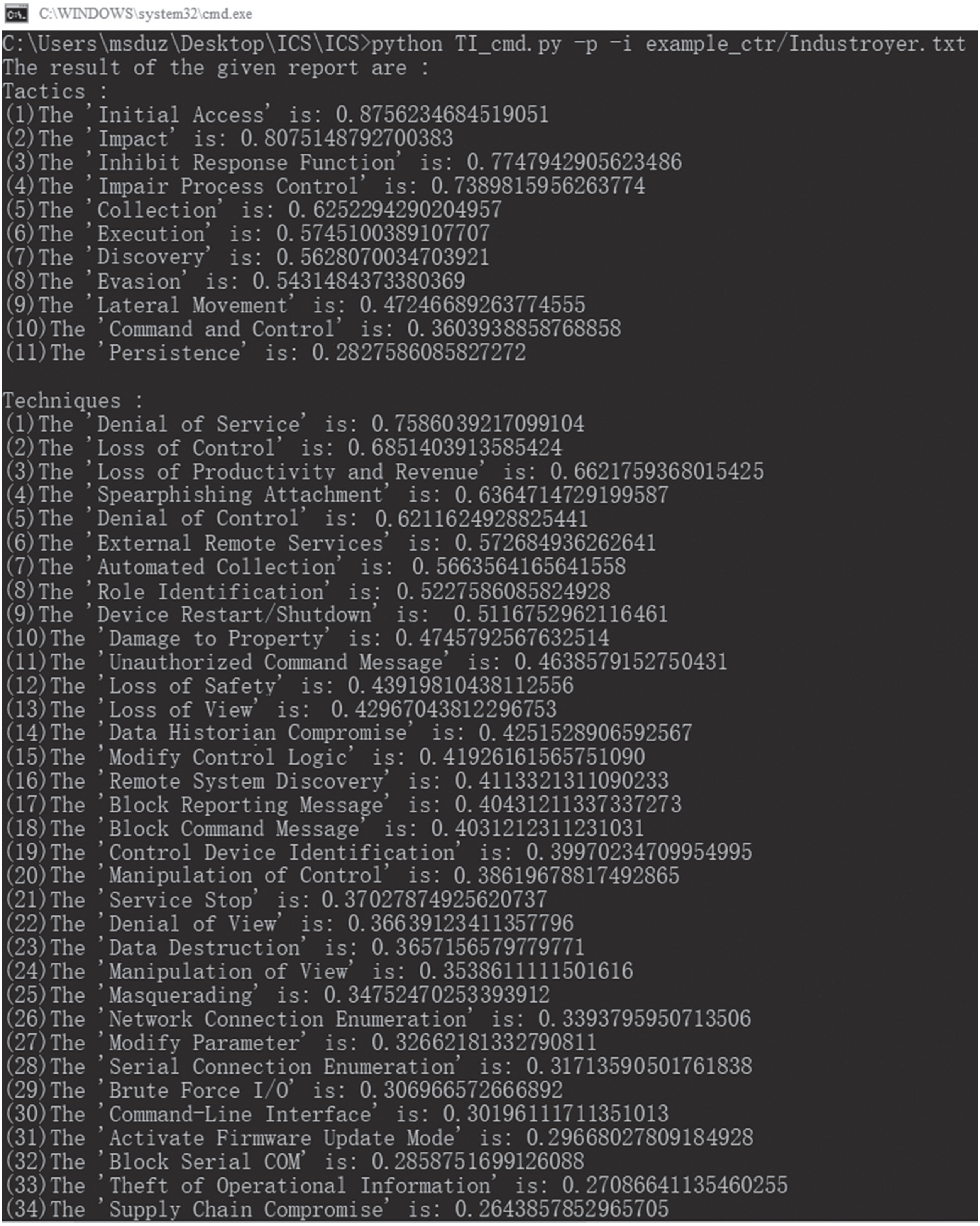
These switches and circuit breakers are digital equivalents of analogue switches; technically they can be engineered to perform various functions. Thus, the potential impact may range from simply turning off power distribution, cascading failures and more serious damage to equipment. The severity may also vary from one substation to another, as well. Needless to say, disruption of such systems can directly or indirectly affect the functioning of vital services.

Industroyer’ s dangerousness lies in the fact that it uses protocols in the way they were designed to be used. The problem is that these protocols were designed decades ago, and back then industrial systems were meant to be isolated from the outside world. Thus, their communication protocols were not designed with security in mind. That means that the attackers didn’ t need to be looking for protocol vulnerabilities; all they needed was to teach the malware “to speak” those protocols.

The recent power outage occurred on December 17th, 2016, almost exactly one year after the well-documented cyberattack that caused a blackout that affected around 250,000 households in several regions in Ukraine on December 23rd, 2015.

In 2015, the perpetrators infiltrated the electricity distribution networks with the BlackEnergy malware, along

FIGURE 4: Threat intelligence of Industroyer attack.



```
C:\Users\msduz\Desktop\ICS\ICS>python TI_cmd.py -p -i example_ctr/Industroyer.txt
The result of the given report are :
Tactics :
(1)The 'Initial Access' is: 0.8756234684519051
(2)The 'Impact' is: 0.8075148792700383
(3)The 'Inhibit Response Function' is: 0.7747942905623486
(4)The 'Impair Process Control' is: 0.7389815956263774
(5)The 'Collection' is: 0.6252294290204957
(6)The 'Execution' is: 0.5745100389107707
(7)The 'Discovery' is: 0.5628070034703921
(8)The 'Evasion' is: 0.5431484373380369
(9)The 'Lateral Movement' is: 0.47246689263774555
(10)The 'Command and Control' is: 0.3603938858768858
(11)The 'Persistence' is: 0.2827586085827272

Techniques :
(1)The 'Denial of Service' is: 0.75860 39217099104
(2)The 'Loss of Control' is: 0.6851403913585424
(3)The 'Loss of Productivity and Revenue' is: 0.6621759368015425
(4)The 'Spearphishing Attachment' is: 0.6364714729199587
(5)The 'Denial of Control' is: 0.6211624928825441
(6)The 'External Remote Services' is: 0.572684936262641
(7)The 'Automated Collection' is:  0.5663564165641558
(8)The 'Role Identification' is: 0.5227586085824928
(9)The 'Device Restart/Shutdown' is:  0.5116752962116461
(10)The 'Damage to Property' is: 0.4745792567632514
(11)The 'Unauthorized Command Message' is: 0.4638579152750431
(12)The 'Loss of Safety' is: 0.43919810438112556
(13)The 'Loss of View' is:  0.42967043812296753
(14)The 'Data Historian Compromise' is: 0.4251528906592567
(15)The 'Modify Control Logic' is: 0.41926161565751090
(16)The 'Remote System Discovery' is: 0.4113321311090233
(17)The 'Block Reporting Message' is: 0.40431211337337273
(18)The 'Block Command Message' is: 0.4031212311231031
(19)The 'Control Device Identification' is: 0.39970234709954995
(20)The 'Manipulation of Control' is: 0.38619678817492865
(21)The 'Service Stop' is: 0.37027874925620737
(22)The 'Denial of View' is: 0.366 39123411357796
(23)The 'Data Destruction' is: 0.3657156579779771
(24)The 'Manipulation of View' is: 0.3538611111501616
(25)The 'Masquerading' is: 0.34752470253393912
(26)The 'Network Connection Enumeration' is: 0.3393795950713506
(27)The 'Modify Parameter' is: 0.32662181332790811
(28)The 'Serial Connection Enumeration' is: 0.31713590501761838
(29)The 'Brute Force I/O' is: 0.306966572666892
(30)The 'Command-Line Interface' is: 0.30196111711351013
(31)The 'Activate Firmware Update Mode' is: 0.29668027809184928
(32)The 'Block Serial COM' is: 0.2858751699126088
(33)The 'Theft of Operational Information' is: 0.27086641135460255
(34)The 'Supply Chain Compromise' is: 0.2643857852965705
```

Figure 5: Identification results by the proposed method.

forest; and 3. comparison between the present method and the SyntaxNet method. The results are analyzed and discussed from two aspects of accuracy and recall.

The Python platform was used to train the model. We collected more than 4000 threat intelligence of industrial control systems from the open-source threat platform,

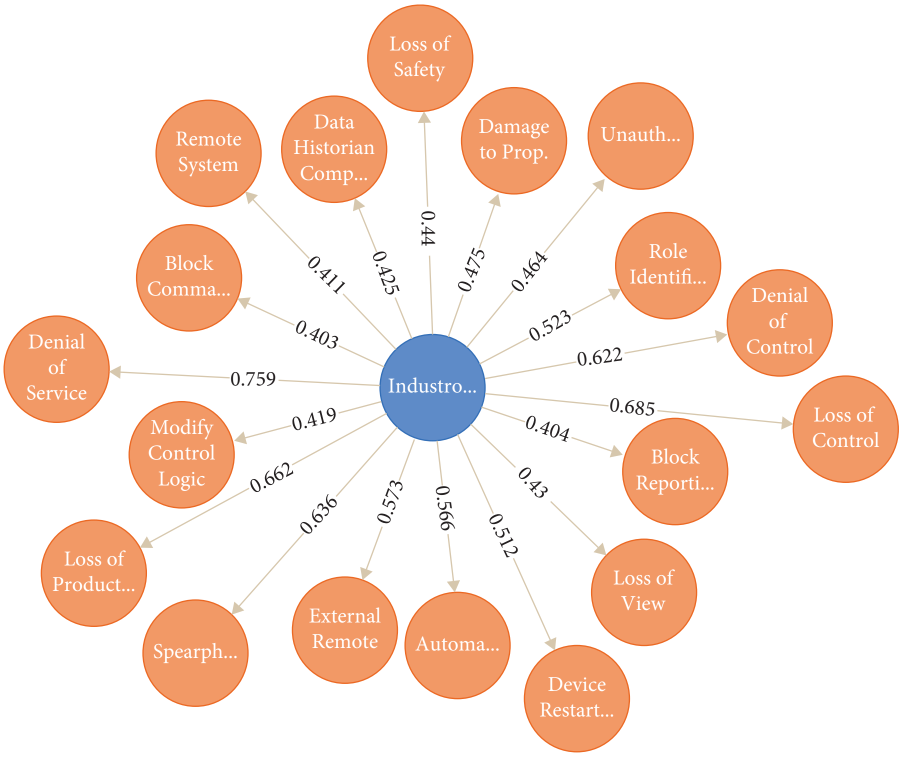


Figure 6: Logic map of attack behavior.

Table 2: Accuracy and recall of the two methods.

	No feature weighting added	Feature weighting added (the present method) (%)
Accuracy rate	86.69%	89.87
Recall rate	83.6%	87.1

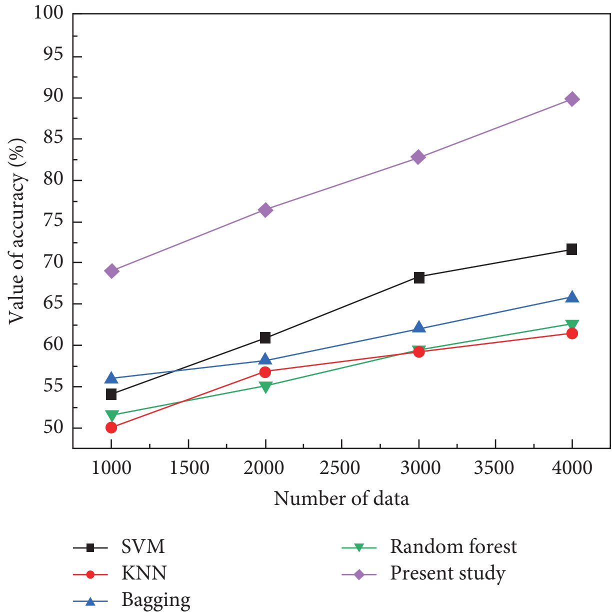


Figure 7: Accuracy of the proposed method, KNN, random forest, and bagging.

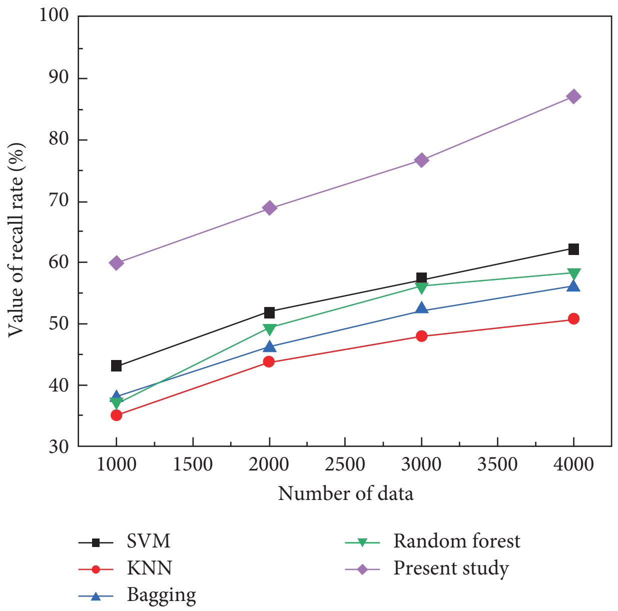


Figure 8: Recall rate of the proposed method, KNN, random forest, and bagging.

Table 3: Accuracy and recall of each method.

	The present method (%)	SVM (%)	KNN (%)	Bagging (%)	Random forest (%)
Accuracy rate	89.87	71.67	61.4	67.81	64.8
Recall rate	87.1	62.14	50.78	56.12	58.4

forming the training set. In addition, we randomly selected 100 threat intelligence of more than ten mainstream attacks faced by the current industrial control system, including Industroyer, WannaCry, and Stuxnet, forming the test data set of the present experiment.

To verify the value of "feature weighting operation" in improving attack analysis capability, we compared the methods with feature weighting and without feature weighting. The results are shown in Table 2. The accuracy and recall of the method without feature weighting are 86.69% and 83.6%, respectively, whereas they are significantly improved with feature weighing, which can be as high as 89.87% and 87.1%. The performance of the method with feature weighing appears much better.

Many researchers have carried out text analysis in recent years and achieved good results using SVM [28], bagging algorithm, KNN classification algorithm, decision tree algorithm, random forest algorithm, neural network model, and other methods. Therefore, we selected four typical methods of SVM, bagging algorithm, K-nearest neighbor classification algorithm, and random forest algorithm to compare and analyze the attack behavior identification results with the method proposed in this study to verify the effectiveness of the current method. It can be seen from Figures 7 and 8 that the accuracy and recall rate can be significantly improved with the iterative training of the amount of a large sample. As shown in Table 3, the accuracy rate of the present method, SVM, K-nearest neighbor classification algorithm (KNN), bagging algorithm, and random forest algorithm is 89.87%, 71.67%, 61.4%, 67.81%, and 64.8%; the recall rate of the present method is 87.1%, 62.14%, 50.78%, 56.12%, and 58.4% [29].

The results indicate that the proposed method in this study is ideal with higher accuracy and a better recall. The result of the SVM model is slightly better than other methods, but the accuracy is much lower than that of this method. KNN classification method needs no training, and it is time-saving, but it possesses the disadvantages of common computing capability. The accuracy of the bagging method is high. Still, all the predicted variables are considered during training, and the more robust predicted variables are placed at the top split point of the method. Hence, the reliability of this method is relatively low. The random forest method uses the decision tree as the primary classifier, improving the overall recall rate. However, due to the high number of iterations, it is time-wasting and accessible to overfitting.

4.3. Experiment 3: Usefulness Verification of Attack Behavior Threat Value.

We hired four safety experts to carry out the experiments with us. By comparing the experimental results of the proposed method in this study with the evaluation results of security experts, the rationality of the attack threat value generation method proposed in this study is verified.

In the experiment, "man-in-the-middle attack," "flooding," "spear phishing," and "code inclusion" are selected to simulate

Table 4: Evaluation scores.

Attack behavior	Expert evaluation Score_1	Expert evaluation Score_2	Expert evaluation Score_3	Expert evaluation Score_4	Our approach score
Man-in-the-middle attack	0.76	0.73	0.79	0.75	0.75
Flooding	0.89	0.86	0.91	0.88	0.92
Spear phishing	0.59	0.57	0.58	0.60	0.55
Code inclusion	0.12	0.10	0.15	0.13	0.14

the attack on the industrial control system. The present method and security experts in this study analyze and evaluate the threat degree of different attacks, respectively. Among them, the evaluation score is in the range of 0–1, and the threat degree of attack behavior is directly proportional to the score. The evaluation results are shown in Table 4.

It can be seen from Table 4 that the evaluation results of the method proposed in this study are consistent with the evaluation opinions given by security experts. The threat of attack behavior is flooding > man-in-the-middle attack > spear phishing > code inclusion. Experiments show that the attack behavior threat value generation method proposed in this study can effectively analyze the threat degree of attack behavior and early warning response. According to the threat score of attack behavior, the information security analysts of IIoT can take corresponding precautions to ensure the safe operation of IIoT.

By comparing the experiments and the current classical text analysis and attack behavior evaluation methods, the IIoT threat intelligence analysis method based on feature weighting and BERT-BiGRU proposed in this study possess advantages in accuracy and recall. In addition, it is more effective in evaluating threat behavior score, which is much closer to the score assessed by experts, resulting in more practical.

5. Conclusions

This study presents a threat intelligence analysis method of IIoT based on feature weighting and BERT-BiGRU. This method can automatically identify and classify the attacks in threat intelligence and calculate the threat value of each attack behavior. The threat value can provide a reference for the judgment of emergency response sequence and improve the accuracy and efficiency of emergency response, resulting in adequate security protection for 5G-oriented IIoT. The experiments show that the proposed method is more accurate than the other common methods and is more suitable for the unstructured threat intelligence analysis of IIoT.

In the future, we will complete an affair map based on threat intelligence to improve our emergency response capabilities to attack further.

Data Availability

The raw/processed data required to reproduce these findings cannot be shared at this time as the data also form part of an ongoing study.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] D. Minoli and B. Occhiogrosso, “Practical aspects for the integration of 5G networks and IoT applications in smart cities environments,” *Wireless Communications and Mobile Computing*, vol. 2019, Article ID 5710834, 2019.

[2] B. Antiy, “Comprehensive analysis of attacks on Ukraine’s power system,” *Information Security Research*, vol. 2, no. 6, p. 243, 2016.

[3] L. Fang, L. Huang, Q. Zhao, and A. Pan, “Power grid security in super large cities from the perspective of blackout in Venezuela,” *Power and Energy*, vol. vol. 4, no. 6, pp. 674–677, 2019.

[4] M. Bromiley, “Threat intelligence: what it is, and how to use it effectively,” *SANS Institute InfoSec Reading Room*, vol. 15, p. 172, 2016.

[5] M. Broda, M. Hervieux, and H. Habib, “Cyber threat intelligence threat and vulnerability assessment of service supplier chain,” *U.S. Patent Application*, 2020.

[6] X. Rong and D. Song, ““Research on cyber attack defense system based on big data and threat intelligence,” *Journal of Information Security Research*, vol. 5, pp. 383–387, 2019.

[7] A. AlienVault, “Threat intel resources,” https://start.me/p/rxRbpo/ti.

[8] IBM, Threat Intel Resources, https://exchange.xforce.ibmcloud.com/.

[9] B. Strom, A. Applebaum, D. Miller, N. Nickels, A. Pennington, and C. Thomas, “Mitre ATT&CKTM: Design and Philosophy,” MITRE, McLean, VA, USA, 2018, https://www.mitre.org/sites/default/files/publications/pr-18-0944-11-mitre-attack-design-and-philosophy.pdf.

[10] J. Gao and L. Fan, ““Kernel-based weighted discriminant analysis with QR decomposition and its application to face recognition,” *WSEAS Transactions on Mathematics archive*, vol. 10, no. 10, pp. 358–367, 2011.

[11] Y. Wu, C. Huang, X. Zhang, and H Zhou, “GroupTracer: automatic attacker TTP profile extraction and group cluster in Internet of things,” security and communication networks,” vol. 2020, Article ID 8842539, 14 pages, 2020.

[12] L. Liu, J. Lin, Q. Wang, and X. Xu, “Research on network malicious Code detection and provenance tracking in future network,” in *Proceedings of the 2018 IEEE International Conference on Software Quality, Reliability and Security Companion (QRS-C)*, Lisbon, Portugal, July 2018.

[13] H. Zhang, G. Shen, C. Guo, Y. Cui, and C. Jiang, “EX-action: automatically extracting threat actions from cyber threat intelligence report based on multimodal learning,” *Security and Communication Networks*, vol. 2021, Article ID 5586335, 12 pages, 2021.

[14] H. Zhang, Y. Yi, J. Wang, N. Cao, and Q. Duan, "Network attack prediction method based on threat intelligence for IoT," *Multimedia Tools and Applications*, vol. 78, no. 21, pp. 30257–30270, 2019.

[15] D. Preuveneers, W. Joosen, J. B. Bernabe, and A. Skarmeta, "Distributed security framework for reliable threat intelligence sharing," *Security and Communication Networks*, vol. 2018, Article ID 8833765, 15 pages, 2018.

[16] H. Hinne, "Rationality constraints in cyber defense: incident handling, attribution and cyber threat intelligence," *Computers & Security*, vol. 109, 2021.

[17] P. Bernhard, H. Geoff, and F. Eibe, "Classifier chains for multi-label classification," *Machine Learning*, vol. 85, no. 3, 2011.

[18] I. Yen, X. Huang, W. Dai, P. Ravikumar, I. Dhillon, and E. Xing, "Ppdsparse: a parallel primal-dual sparse method for extreme classification," in *Proceedings of the 2017 Conference on ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 545–553, ACM, Halifax NS, Canada, Auguest 2017.

[19] J. Yang, Y. Liu, and J. Luo, "Multi-label extreme classification based on subset topic model," *Computer Engineering and Design*, vol. 12, pp. 140–145, 2020.

[20] H. Tan and Z. Liu, "Multi-label K nearest neighbor algorithm by exploiting label correlation," *Journal of Computer Applications*, vol. 35, no. 10, pp. 2761–2765, 2015.

[21] Y. Prabhu and M. Varma, "Fastxml: a fast, accurate and stable tree-classifier for extreme multi-label learning," ", ACM, in *proceedings of the 2014 Conference on ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 263–272, Auguet 2014.

[22] C. Guibin, Y. Deheng, X. Zhenchang, C. Jieshan, and C. Erik, "Ensemble application of convolutional and recurrent neural networks for multi-label text categorization," in *Proceedings of the 2017 International Joint Conference on Neural Network-sAnchorage*, pp. 2377–2383, AK, USA, May 2017.

[23] L. Xiao, X. Huang, B. Chen, and L. Jing, "Label-specific document representation for multi-label text classification," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, pp. 466–475, Hong kong, China, January 2019.

[24] J. Wehrmann, R. Cerri, and R. Barros, "Hierarchical multi-label classification networks," in *Proceedings of the 35th International Conference on Machine earning ICML*, pp. 8321–8330, Stockholmsmässan, Stockholm SWEDEN, February 2018.

[25] S. Barnum, "Common attack pattern enumeration and classification (CAPEC) schema," *Department of Homeland Security*, 2008.

[26] A. Cherepanov and R. Lipovsky, "Industroyer: biggest threat to industrial control systems since stuxnet," *WeLiveSecurity, ESET*, vol. 12, 2017.

[27] A. Cherepanov, *Win32/Industroyer: A New Threat for Industrial Control Systems*, ESET, 2017, https://www.welivesecurity.com/wpcontent/uploads/2017/06/Win32_Industroyer.pdf.

[28] M. Awad and R. Khanna, "Support vector machines for classification," in *Efficient Learning Machines*, pp. 39–66, 2015.

[29] F. Pedregosa, G. Varoquaux, A. Gramfort, and V. Michel: API Reference, https://scikit-learn.org/stable/modules/classes.html.

Hindawi
Security and Communication Networks
Volume 2021, Article ID 8251697, 10 pages
https://doi.org/10.1155/2021/8251697

Research Article

Trajectory as an Identity: Privacy-Preserving and Sybil-Resistant Authentication for Internet of Vehicles

Jiangtao Li,[1,2] Zhaoheng Song,[1] Yufeng Li,[1,2] Chenhong Cao,[1,2] and Yuanhang He[3]

[1] *School of Computer Engineering and Science, Shanghai University, Shanghai, China*
[2] *Purple Mountain Laboratories, Nanjing, China*
[3] *No. 30 Research Institute of China Electronics Technology Group Corporation, Chengdu, China*

Correspondence should be addressed to Yufeng Li; liyufeng_shu@shu.edu.cn

Received 30 July 2021; Revised 15 September 2021; Accepted 3 December 2021; Published 31 December 2021

Academic Editor: Emanuele Maiorana

With the advancement of the 5G network, the Internet of Vehicles (IoV) is becoming more and more attractive for academic researchers and industrial. A main challenge of IoV is to guarantee the authenticity of messages and protect drivers' privacy simultaneously. The majority of privacy-preserving authentication schemes for IoV adopt pseudonyms or group signatures to achieve a balance between security and privacy. However, defending the Sybil attacks in these schemes is challenging. In this work, we propose a novel privacy-preserving authentication scheme for announcement messages, which utilizes the trajectories of vehicles as their identities. When an authenticated message is verified, the verifier is convinced that the message is generated by a vehicle that has a unique masked trajectory. Meanwhile, the real trajectories of vehicles are kept private. In particular, our scheme achieves Sybil attack resistance without the limitation of trajectory length even when the attacker is allowed to use cloud services.

1. Introduction

Internet of Vehicles (IoV) is made possible with the advancement of 5G network, which achieves lower latency and supports higher mobility [1, 2]. As a special type of Internet of Things (IoT) [3], IoV allows vehicles to communicate with surrounding vehicles, Road Side Units (RSUs), and infrastructures. By providing road condition warnings, cooperative perception, and driving assistant services, IoV can improve traffic efficiency and road safety [4]. However, vehicle-to-vehicle communications in IoV do not adopt any security mechanism in the network layer [5, 6]. For privacy concerns, the IP address of each vehicle is also dynamically changed. Without a security mechanism, a malicious vehicle in IoV may send false data or misleading messages to other vehicles and infrastructures. For instance, a vehicle may claim a fake traffic jam by broadcasting an announcement message to achieve a better driving condition for himself. Hence, it is necessary to authenticate the messages broadcast in IoV.

Adopting signature schemes is a commonly used way to guarantee the authenticity of messages in IoV [7]. If a signature is verified, a vehicle is convinced that the message is sent by the claimed sender and not tempered in transmission. However, due to the openness of wireless communication, an attacker can violate vehicle privacy by simply collecting messages broadcast in IoV [8]. Pseudonym mechanism is a classical way to achieve a trade-off between security and privacy in IoV. Under the pseudonym mechanism, each vehicle will generate multiple certified identities (i.e., public/private key pairs). A vehicle needs to use a fresh identity for each message to keep its identity private. This will bring the vehicle extra storage cost. Besides, the pseudonym mechanism is vulnerable to Sybil attacks. A vehicle can use multiple identities simultaneously to pretend to be multiple vehicles.

One approach to overcome the drawback of the pseudonym mechanism is employing the group signature instead of traditional signatures. In group signatures, any group member is allowed to generate signatures on behalf of the group. This way, vehicles' identities will not be leaked during message authentication. To defend Sybil attacks, the researchers also propose a new primitive named message linkable group signature [9]. If an attacker broadcasts the same message twice, this behavior can be efficiently detected. It is obvious that the message linking technique is useful only when the attacker simply sends the same message repeatedly in launching a Sybil attack. If an attacker tries to send different messages with the same semantic, this Sybil attack defense technique will fail.

Several works have been proposed to deal with the Sybil attacks, launched with different messages. One solution is to use the cryptographic puzzle to prevent a vehicle from impersonating multiple vehicles since the computational resource of each vehicle is limited [10]. However, if a vehicle is allowed to outsource the puzzle to the cloud, this solution will not work. Later, Baza et al. [11] proposed to consider both the locations (trajectories) [12] and cryptographic puzzles in detecting Sybil attacks. In [11], only vehicles proved to have traversed some RSUs are regarded as valid vehicles. In contrast, their work does not consider the situation where vehicles may collude with each other. Furthermore, when the vehicle is allowed to use cloud service, the Sybil attack can only be efficiently detected if the trajectory length is long enough.

1.1. Contribution. To cope with the above-mentioned problems. We propose to use trajectory as an identity of a vehicle to achieve privacy-preserving and Sybil-resistant authentication for IoV.

In the proposed scheme, when a vehicle needs to broadcast an announcement message, it will sign the message with a freshly generated identity. Then, the signed message is broadcast together with the vehicle's trajectory. The trajectory can be viewed as a certificate of the freshly generated identity. When an announcement message is verified, it is established that the message is signed by the vehicle, confirmed to have a valid trajectory in IoV. Besides, as the identities of RSUs are masked, the real trajectory of vehicles will not be revealed, and hence, the privacy of vehicles is preserved.

In particular, when the vehicle is allowed to use cloud services, our proposal is also secure against Sybil attacks without the trajectory length limitation. A vehicle in IoV needs to solve a series of computational puzzles before it can send a valid trajectory request to the RSU. Since the computational puzzles are bound with the secret keys of vehicles, rational vehicles will not choose to outsource the secret key and hence the computational puzzles. Besides benefits from the sequential aggregate signatures used in the scheme, the scheme also achieves efficient RSU verification.

1.2. Related Work. Existing privacy protection authentication schemes for IoV can generally be categorized into two types: pseudonym-based ones and group signature-based ones. In the pseudonym-based authentication scheme, each vehicle can hold multiple pseudonymous identities [13]. The vehicle can use a new pseudonymous identity to send messages at regular intervals or after sending several messages according to its own privacy needs. No entity except for a trusted authority can distinguish whether any two pseudonymous identities belong to the same vehicle. However, the anonymity of these schemes is achieved at the cost of frequently changing pseudonymous identities [14]. Under the traditional Public Key Infrastructure (PKI) architecture [15], the trusted authority needs to issue a certificate for each pseudonymous public key, so the vehicle needs to store a large number of pseudonymous certificates and their corresponding keys in advance. This brings extra storage costs to vehicles.

In order to reduce the cost of storage and communication of vehicles caused by pseudonym management, privacy-preserving authentication schemes based on group signature are proposed [16, 17]. In these schemes, when the vehicle needs to change its pseudonym, the vehicle can use its group private key to issue a certificate to the new pseudonym, so there is no need to store a large number of pseudonyms. Later, Hao et al. [18] also proposed a distributed group model [19] for group signature-based authentication schemes in IoV. Under this model, an RSU acts as a group administrator to manage the group and distribute certificates and keys to vehicles entering its area. When the vehicle leaves the communication range of the current RSU and enters the next RSU, the vehicle will be automatically revoked. Recently, Lin et al. [20] and Mollah et al. [21] also proposed to use blockchain and smart contracts in realizing identity management and improve existing privacy protection authentication schemes. However, traditional group signature-based solutions are not secure against Sybil attacks.

In order to defend Sybil attacks, Wu et al. [9] proposed the concept of message linkable group signature. In the message linkable group signature-based schemes, if a vehicle generates more than two signatures for the same message, it can be efficiently detected. Chen et al. [22] also proposed to use direct anonymous authentication schemes and one-time anonymous signatures to realize message linkability, which is similar to the effect of [9]. However, the message linking techniques are only useful for event-driven announcement messages in IoV. For the announcement messages that allow the vehicle to send different messages with the same semantics, message linking techniques will not play the role of defending Sybil attacks.

Another method to realize Sybil detection is anonymous trajectory-based authentication. In [12], each vehicle needs to request a trajectory from the RSU when it passes through. Compared with the message linkable solutions, this method may prevent an attacker from sending Sybil messages with the same semantics. At the same time, in order to achieve anonymity, RSU needs to use a zero-knowledge signature that can be linked in a short time to protect the privacy of the vehicle's location. However, the vehicle may generate multiple false trajectories and send them to the RSU to achieve a denial-of-service (DOS) attack. Hence, Liu et al. [10] proposed to use cryptographic puzzles to resist DOS and Sybil attacks in the IoV. Recently, Baza et al. [11] combined the idea of the cryptographic puzzle with the trajectory proof method in [12] and proposed a solution to detect Sybil attacks. However, this work does not consider the situation where vehicles may collude with each other. When the vehicle is allowed to use cloud service, the Sybil attack can only be efficiently detected if the trajectory length is long enough.

1.3. Organization. The rest of the paper is organized as follows: we describe the system architecture and design goals in Section 2. Section 3 gives some of the cryptographic primitives used in our scheme. We present our scheme in Section 4 and analyze its security and efficiency in Section 5. Finally, we conclude the paper in Section 6.

2. System Architecture and Design Goals

In this section, we describe the system models and design goals of this work.

2.1. System Architecture. As shown in Figure 1, the IoV scenario considered in this work consists of the following entities:

(i) Road side units (RSUs): RSUs in the system are equipped with computation and communication modules. They are responsible for issuing partial authenticated trajectories for vehicles.

(ii) Vehicles: vehicles in the system can communicate with surrounding vehicles and RSUs. When they encounter an event (e.g., traffic accident), they can send an announcement message to the announcement manager. We note that the vehicles might be malicious. They may send fake messages or launch Sybil attacks.

(iii) Trusted authority (TA): the trusted authority can be the Department of Motor Vehicles (DMV). It is responsible for issuing certificates for vehicles and initializing the system.

(iv) Announcement manager: the announcement manager is an authority that collects and verifies the announcement messages sent by the vehicles. It is supposed to be a semihonest authority; that is, it will perform all steps of the scheme honestly. However, it is curious about the privacy of vehicles.

2.2. Design Goals. A privacy-preserving Sybil-resistant announcement scheme for IoV should achieve the following security and efficiency requirements.

(i) Privacy-preserving authentication: it is required that all announcement messages should be authenticated. Besides, the real trajectories of vehicles should be kept private.

(ii) Sybil attack resistance: if a vehicle tries to launch a Sybil attack, that is, pretending to be multiple vehicles, it can be efficiently detected. In particular, the scheme should be secure, even if the vehicle is allowed to use cloud service to launch such an attack.

(iii) Resisting RSU compromise: in the system, the authenticity of messages should be kept even if some RSUs are compromised by the attacker.

(iv) Efficient RSU verification: the RSU is responsible for verifying the trajectory requests. The request verification should be efficiently processed. Besides, the RSU should also be able to handle the trajectory requests efficiently even under Denial-of-Service (DoS) attacks.

3. Preliminary

This section describes several cryptographic primitives used in our proposed scheme.

3.1. Sequential Aggregate Signature. Sequential aggregate signature (SeqAS) allows a signer to add his signature on a different message to the current signature sequentially. The aggregated signature has the same size as the previous signature. A signature will be able to pass the verification if and only if all the aggregated signatures are generated correctly.

Definition 1. (sequential aggregate signature). A sequential aggregate signature (SeqAS) scheme consists of four algorithms, AggSetup, AggKeyGen, AggSign, and AggVerify, defined as follows:

(i) AggSetup: the setup algorithm takes as input a security parameter and outputs public parameters

(ii) AggKeyGen: the key generation algorithm takes as input a security parameter and outputs a public key PK and a private key SK

(iii) $\mathrm{AggSign}_{\mathrm{SK}}(\alpha\prime, M; M)$: the aggregate signing algorithm takes as input an aggregate-so-far $\alpha\prime$ on messages $M = (tr_1, \ldots M_k)$, a message M, and a private key SK; it outputs a new aggregated signature α

(iv) $AggVerify(\alpha, M; \mathrm{PK})$: the aggregate verification algorithm takes as input an aggregate signature α on messages $M = (tr_1, \ldots, M_l)$; it outputs $\top$ or $\perp$ which indicates the signature is valid or not

In this paper, we adopt the SeqAS signature scheme proposed by Lee et al. [23].

3.2. Threshold Signatures without a Trusted Dealer. A threshold signature is a distributed multiparty signature protocol. In a (θ, n)-threshold signature, a valid signature will be reconstructed only when no less than θ signers sign on the same message.

Definition 2. (threshold signature without a trusted dealer). A (θ, n)-threshold signature scheme without trusted dealer consists of the following algorithms:

(i) T Setup : this can be an interactive algorithm run by the players $P_1, P_2, \ldots, P_n$. The algorithm outputs a public key PK. The private output of each player P_i

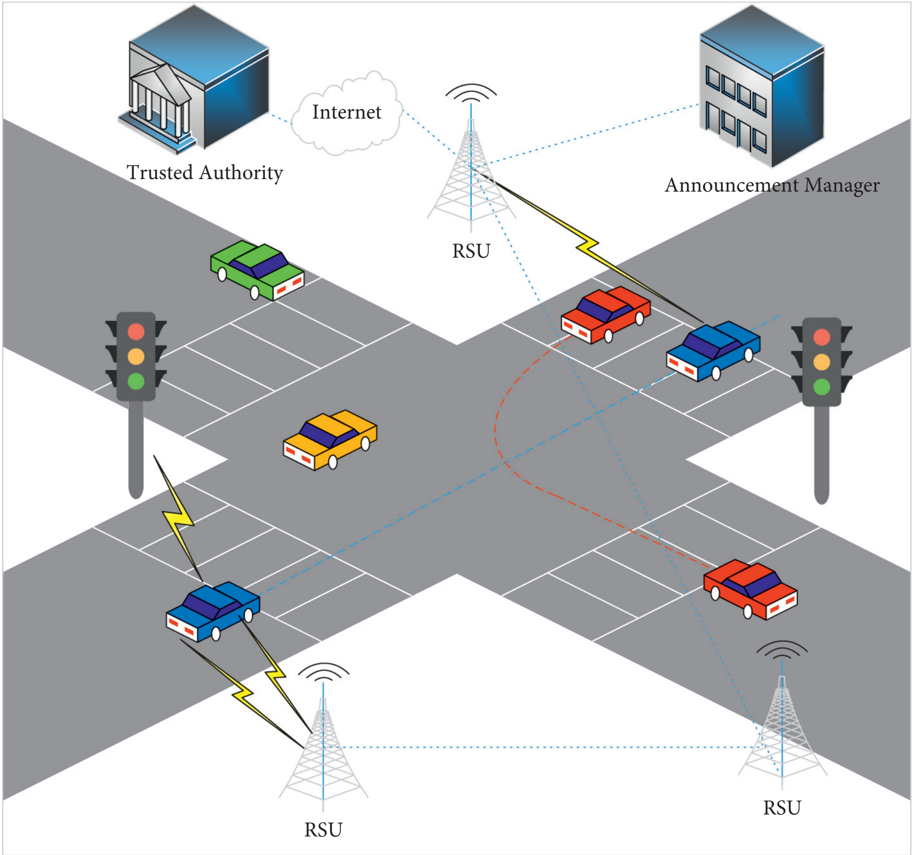


Figure 1: System architecture.

is x_i such that $(x_1, \ldots, x_n) \longrightarrow$ SK, where SK is the secret key corresponding to PK.

(ii) *T Sign*: this algorithm generates the signature share. A player P_i takes a message m and its private key x_i as input and outputs a signature share σ_i.

(iii) Reconstruct : this algorithm takes as a collection of θ signature shares. If all signature shares are valid, it outputs a message signature pair (m, σ); else, it outputs $\perp$.

(iv) $TVerify$: this algorithm takes a message signature pair as input and outputs $\top/\perp$, which indicates the signature is valid or not.

It is easy to construct a secure threshold signature algorithm using BLS short signature [24] and the distributed key generation algorithm for discrete-log-based cryptosystems [25].

3.3. Message Authentication Code.

A message authentication code (MAC) is a cryptographic primitive that can prevent an adversary from modifying the messages sent from one entity to another.

Definition 3. (message authentication code). A message authentication code consists of the following three algorithms:

(i) *Gen*: it outputs a uniformly distributed key k

(ii) MAC: it takes a message m and the secret key k as input and outputs a tag t

(iii) $Vrfy$: it takes a message m and the secret key k and a tag t as input and outputs $\top/\perp$, which indicates the MAC is valid or not

For every secret key k, it holds that $\text{Vrfy}_k(m, \text{MAC}_k(m)) = \top$.

3.4. Computational Puzzle.

A computational puzzle is a computational problem, which is used to verify if an entity has an estimated computational power. The hardness of a computational puzzle can be adjusted by a hardness parameter. A specific computational puzzle is usually represented by a random seed. To solve a computational puzzle, an entity needs to consume some computational resources. However, given a solution, it should be easy to verify the correctness of the solution. The hash-based puzzle is one of the famous computational puzzles used in the proof of work mechanism.

4. Privacy-Preserving and Sybil Attack Resistant Announcement

In this section, we describe the proposed privacy-preserving and Sybil attack resistant announcement scheme. We first give a high-level description of our scheme. Then, we explain the scheme in detail.

4.1. High-Level Description. In the scheme, we use the trajectory as the vehicle's identity to achieve privacy-preserving authentication of announcement messages. The trajectory of a vehicle contains a number of RSUs and timestamps pairs, which indicate that the vehicle encounters an RSU at a specific time. For privacy concerns, the identity of each RSU is masked as a time-varying tag. To get the trajectory authenticated by an RSU, a vehicle needs to solve a series of computational puzzles generated by one of its neighboring RSUs. This is the first line of defending a vehicle from generating multiple trajectories. Besides, considering that a vehicle may use cloud service to bypass this computational puzzle defense line, a vehicle is required to sign on the intermediate computational puzzle results. When a vehicle obtains a verified trajectory, it is able to send an authenticated announcement message. Finally, before accepting an announcement message, we also adopt a Sybil detection method to prevent vehicles from colluding with each other to launch a Sybil attack. The details of our scheme are described in the following subsections.

4.2. System Initialization. In this stage, RSUs run the T Setup algorithm of the (θ, n)-threshold signature scheme without a trusted dealer. At the end of this algorithm, it generates a public key PK. Each RSU R_k holds a secret key x_k. Besides, each vehicle holds the public keys corresponding to the secret key shares of the neighboring RSUs.

Besides, TA runs the *Gen* algorithm of message authentication code to generate the secret key MK and the *AggSetup* algorithm of the sequential aggregate signature (SeqAS) to generate the public parameters. MK is distributed to all the RSUs in the system. Each vehicle v_i generates an initial public/private key pairs $(\mathrm{PK}_1^{v_i}, \mathrm{SK}_1^{v_i})$ using the *AggKeyGen* algorithm of SeqAS and obtains certificates for the public key from the TA so that vehicles can anonymously authenticate to the RSUs. Getting these certificates from the TA can be done during the annual vehicle registration at the DMV.

4.3. Trajectory Requests. In this section, we explain the trajectory requests stage by an example illustrated in Figure 2 where a vehicle travels along 4 RSUs. It is assumed that the threshold θ of the signature scheme is set to be 3. The vehicle v_i requests a trajectory proof as follows:

When v_i encounters the first RSU R_1, v_i interacts with R_1 as follows:

(1) It computes a signature $\alpha_1^{v_i} = Aggsign_{\mathrm{SK}_1^{v_i}}(t_0)$, where t_0 is the timestamp, then sends the request:

$$\mathrm{Ticket}_0 \coloneqq t_0 \| \alpha_1^{v_i} \| C_{TA}(\mathrm{PK}_1^{v_i}), \tag{1}$$

to R_k, where $C_{TA}(\mathrm{PK}_1^{v_i})$ is a certificate of vehicle v_i which was generated by the TA.

On receiving the request from v_i, R_1 verifies the signature $\alpha_1^{v_i}$. If $\alpha_1^{v_i}$ is valid, it computes $\mathrm{Tag}_1 = MAC_{\mathrm{MK}}(ID_{R_1} \| t_1)$, where MK is the shared secret key among all RSUs, ID_{R_1} is the identity of RSU R_1, t_1 is the current timestamp. Let $tr_1 = (\mathrm{PK}_1^{v_1} \| t_1 \| \mathbf{Tag}_1)$; it then computes $\sigma_{R_1}(tr_1) = T\ Sign_{\mathrm{sk}_{R_1}}(tr_1)$ and sends back

$$\mathcal{T}_{R_1} \coloneqq tr_1 \| \sigma_{R_1}(tr_1). \tag{2}$$

(2) When v_i receives $\mathcal{T}_{R_1}$ and travels from R_1 to R_2, it generates a new public/private key pairs $(\mathrm{PK}_2^{v_i}, \mathrm{SK}_2^{v_i})$ as its fresh pseudonym. It needs to solve a series of the computational puzzle before submitting a trajectory request. In detail, let $\mathcal{N}_0 = \mathcal{T}_{R_1}, \alpha_0^{v_i} \coloneqq \mathcal{T}_{R_1}$, and $j = 1$. It performs the following operations repeatedly until it arrives at the communication radius of R_2:

(a) Find a nonce $\mathcal{N}_j$ such that $H(\mathcal{N}_j \| \mathrm{PK}_2^{v_i} \| \mathcal{N}_{j-1}) < \beta$

(b) Compute a signature using the AggSign algorithm: $\alpha_j^{v_\mathrm{i}} = \mathrm{AggSign}_{\mathrm{SK}_2^{v_\mathrm{i}}}(\alpha_{j-1}^{v_\mathrm{i}}, \mathcal{N}_j)$

(c) $j = j + 1$

Suppose $j = j^*$; when v_i arrives at the range of R_1, it sends the ticket to R_2:

$$\mathrm{Ticket}_1 \coloneqq \left(\mathcal{T}_{R_1,} \mathcal{N}_1^{R_1}, \ldots, \mathcal{N}_{j^*}^{R_1}, \alpha_1^{v_i,R_1}, \ldots, \alpha_{j^*}^{v_i,R_1}\right). \tag{3}$$

When R_2 receives the ticket, R_2 verifies the ticket as follows:

(a) Compute $\hat{j} = t^* - t_1/\Delta$, where t^* is the current timestamp and Δ is the estimated time for a vehicle to solve one computational puzzle. If $j^* \geq \hat{j}$, go to the next step; otherwise, return $\perp$. The estimated number of puzzles v_i can solve when traveling from R_1 to R_2,

(b) If $H(\mathcal{N}_j \| PK_2^{v_i} \| \alpha_{j-1}^{v_i}) < T$ holds for all $j \in 1, \ldots, j^*$, go to the next step; otherwise, return $\perp$.

(c) If $AggVerify(\alpha_{j^*}^{v_i}, \mathcal{N}_0, \ldots, \mathcal{N}_{j^*}; \mathrm{PK}_2^{v_i}) = 1$, go to the next step; otherwise, return $\perp$.

(d) If $TVerify(tr_1, \sigma_{R_1}(tr_1)) = \top$, and $\mathbf{Tag}_1 = MAC_{\mathrm{MK}}(ID_{R_1} \| \| t_1)$; otherwise, return $\perp$.

If the ticket is valid, it sends back the partial trajectory:

$$\mathcal{T}_{R_2} \coloneqq tr_2 \| \sigma_{R_2}(tr_2) \| \sigma_{R_2}(tr_1), \tag{4}$$

where

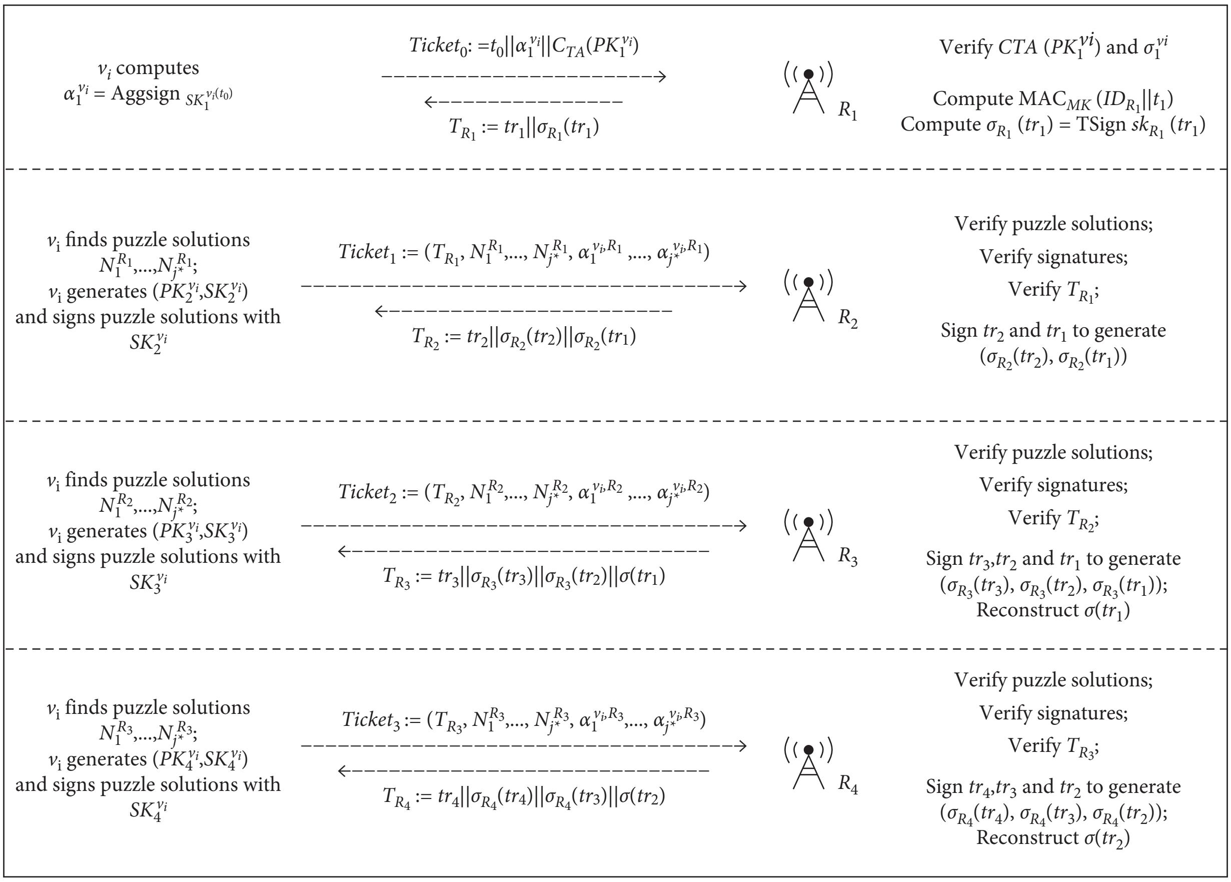

v_i computes
$\alpha_1^{v_i}$ = Aggsign $_{SK_1^{v_i(t_0)}}$
$Ticket_0$: $=t_0||\alpha_1^{v_i}||C_{TA}(PK_1^{v_i})$
$T_{R_1} := tr_1||\sigma_{R_1}(tr_1)$
R_1
Verify CTA (PK_1^{vi}) and σ_1^{vi}
Compute MAC$_{MK}$ $(ID_{R_1}||t_1)$
Compute σ_{R_1} (tr_1) = TSign sk_{R_1} (tr_1)
v_i finds puzzle solutions
$N_1^{R_1}$,...,$N_{j^*}^{R_1}$;
v_i generates $(PK_2^{v_i}$,$SK_2^{v_i})$
and signs puzzle solutions with
$SK_2^{v_i}$
$Ticket_1 := (T_{R_1}, N_1^{R_1},..., N_{j^*}^{R_1}, \alpha_1^{v_i,R_1},..., \alpha_{j^*}^{v_i,R_1})$
$T_{R_2} := tr_2||\sigma_{R_2}(tr_2)||\sigma_{R_2}(tr_1)$
R_2
Verify puzzle solutions;
Verify signatures;
Verify T_{R_1};
Sign tr_2 and tr_1 to generate
$(\sigma_{R_2}(tr_2), \sigma_{R_2}(tr_1))$
v_i finds puzzle solutions
$N_1^{R_2}$,...,$N_{j^*}^{R_2}$;
v_i generates $(PK_3^{v_i}$,$SK_3^{v_i})$
and signs puzzle solutions with
$SK_3^{v_i}$
$Ticket_2 := (T_{R_2}, N_1^{R_2},..., N_{j^*}^{R_2}, \alpha_1^{v_i,R_2},..., \alpha_{j^*}^{v_i,R_2})$
$T_{R_3} := tr_3||\sigma_{R_3}(tr_3)||\sigma_{R_3}(tr_2)||\sigma(tr_1)$
R_3
Verify puzzle solutions;
Verify signatures;
Verify T_{R_2};
Sign tr_3,tr_2 and tr_1 to generate
$(\sigma_{R_3}(tr_3), \sigma_{R_3}(tr_2), \sigma_{R_3}(tr_1))$;
Reconstruct $\sigma(tr_1)$
v_i finds puzzle solutions
$N_1^{R_3}$,...,$N_{j^*}^{R_3}$;
v_i generates $(PK_4^{v_i}$,$SK_4^{v_i})$
and signs puzzle solutions with
$SK_4^{v_i}$
$Ticket_3 := (T_{R_3}, N_1^{R_3},..., N_{j^*}^{R_3}, \alpha_1^{v_i,R_3},..., \alpha_{j^*}^{v_i,R_3})$
$T_{R_4} := tr_4||\sigma_{R_4}(tr_4)||\sigma_{R_4}(tr_3)||\sigma(tr_2)$
R_4
Verify puzzle solutions;
Verify signatures;
Verify T_{R_3};
Sign tr_4,tr_3 and tr_2 to generate
$(\sigma_{R_4}(tr_4), \sigma_{R_4}(tr_3), \sigma_{R_4}(tr_2))$;
Reconstruct $\sigma(tr_2)$

Figure 2: Example of trajectory requests with 4 RSUs.

$$tr_2 = \left(\mathrm{PK}_2^{v_i}\|t_2\|\mathrm{Tag}_2\|t_1\|\mathrm{Tag}_1\right), \qquad (5)$$

and $\mathrm{Tag}_2 = \mathrm{MAC}_{\mathrm{MK}}(ID_{R_2}\|t_2)$.

(3) When the vehicle v_i moves on and reaches the communication radius of R_3, it computes the new ticket:

$$\mathrm{Ticket}_2 := \left(\mathcal{T}_{R_2}, \mathcal{N}_1^{R_2}, \ldots, \mathcal{N}_{j^*}^{R_2}, \alpha_1^{v_i,R_2}, \ldots, \alpha_{j^*}^{v_i,R_2}\right), \qquad (6)$$

similar to the procedure when traveling from R_1 to R_2. When R_3 receives the ticket, it verifies the ticket. If the ticket is valid, it sends back the partial trajectory:

$$\mathcal{T}_{R_3} := tr_3\left\|\sigma_{R_3}(tr_3)\right\|\sigma_{R_3}(tr_2)\|\sigma(tr_1), \qquad (7)$$

where

$$tr_3 = \left(\mathrm{PK}_3^{v_i}\|t_3\|\mathrm{Tag}_3\|t_2\|\mathrm{Tag}_2\|t_1\|\mathrm{Tag}_1\right), \qquad (8)$$

and $\mathrm{Tag}_3 = \mathrm{MAC}_{\mathrm{MK}}(ID_{R_3}\|t_3)$.

We note that, to compute $\sigma(tr_1)$, R_3 needs to compute $\sigma_{R_3}(tr_1)$ and then perform the *Reconstruct* algorithm of the threshold signature scheme described in Section 3.2.

4.4. Message Announcement and Trajectory Update. When a vehicle encounters at least θ RSUs, a vehicle can generate/update a trajectory and send an announcement message with its secret key.

When considering the example that $\theta = 3$, after v_i receives $\mathcal{T}_{R_3}$, it is able to retrieve the signature $\sigma(tr_1)$ on message $tr_1 = (\mathrm{PK}_1^{v_i}\|t_1\|\mathrm{Tag}_1)$. This message signature pair indicates that the vehicle with public key $\mathrm{PK}_1^{v_i}$ reaches a RSU (masked as Tag_1) at time t_1. Hence, $tr_1, \sigma(tr_1)$ is an authenticated trajectory, v_i can compute the signature $\alpha^{v_i}(M) = \mathrm{AggSign}(M)$ and broadcast the announcement message:

$$\mathrm{Annou} := \left(M, \alpha^{v_i}(M); tr_1, \sigma(tr_1); \hat{t}\right), \qquad (9)$$

where $\hat{t}$ is the timestamp.

When v_i reaches R_4, it can obtain

$$\mathcal{T}_{R_4} := tr_4\left\|\sigma_{R_4}(tr_4)\right\|\sigma_{R_4}(tr_3)\|\sigma\|(tr_2) \qquad (10)$$

and hence retrieve $tr_2, \sigma(tr_2)$ as its new authenticated trajectory, which indicates that the vehicle with public key $\mathrm{PK}_2^{v_i}$ reaches a RSU (masked as Tag_1) at time t_1 and another RSU (masked as Tag_2 at time t_2).

We note that if a vehicle keeps extending one trajectory, it is possible for the announcement manager to distinguish whether two announcement messages are generated by the same vehicle or not. If a vehicle does not wish to link two messages, it just needs to drop the old trajectory and start the trajectory request from scratch.

4.5. Announcement Verification and Sybil Attacks Detection. On receiving an announcement message $\mathrm{Annou} := (M, \alpha^{v_i}(M); tr_k, \sigma(tr_k); \hat{t})$, the announcement manager verifies the message as follows:

(a) If $T\,\mathrm{Verify}(tr_k, \sigma(tr_k)) = 1$, go to the next step; otherwise, return $\perp$

(b) If $\mathrm{AggVerify}(M, \alpha^{v_i}(M))$, return $\top$; otherwise, return $\perp$

If an announcement message passes the above verification, the announcement manager will search all existing trajectories to finish the Sybil attack detection. Similar to the method in [12], the following two types of trajectories will be regarded as Sybil trajectories:

(a) Small windows size trajectories: since it is impossible for a vehicle to travel from one RSU to another in a time much shorter than an estimated time, any trajectory containing a window size shorter than a certain time will be regarded as a Sybil trajectory

(b) Long trajectories within a time period: since a vehicle cannot encounter a large number of RSUs within a fixed time, any trajectory that is longer than an estimated length will be regarded as a Sybil trajectory

If a trajectory survives the Sybil attack detection and the signature verification, this announcement message will be accepted by the announcement manager.

5. Analysis and Evaluation

5.1. Security Analysis. We show that our scheme achieves privacy-preserving authentication and Sybil attack resistance via the following two theorems.

Theorem 1. *Our scheme achieves privacy-preserving authentication property defined in Section 2.2.*

Proof. We show our scheme achieves privacy-preserving authentication in the following two aspects.

Firstly, a message will be accepted by the announcement manager if and only if both $(m_k, \sigma(m_k))$ and $M, \alpha^{v_i}(M)$ are valid message signature pairs. The first signature guarantees that the trajectory m_k is signed by at least θ RSUs. The second signature guarantees that the announcement message is signed by the vehicle with public key $\mathrm{PK}_k^{v_i}$ that is the owner of the trajectory m_k. Combining the unforgeability of SeqAS signatures and threshold signatures, we have that the announcement messages are authenticated.

Secondly, the real trajectories of vehicles are hidden to both eavesdroppers and curious announcement managers. Since vehicles will generate a fresh pseudonym when they travel from one RSU to another, it is not possible to link different pseudonyms for both eavesdroppers and announcement manager. Since all the identities of RSUs are masked as a time-varying tag, it is not possible to link two vehicles that passed by the same location at different times. Hence, it is not possible for vehicles to reveal the real trajectories of vehicles.

Overall, we have that our scheme achieves privacy-preserving authentication. □ □

Theorem 2. *Our scheme achieves the Sybil attack resistant property defined in Section 2.2.*

Proof. Firstly, it is difficult for a vehicle to generate two valid trajectory requests to a specific RSU. When a vehicle travels from one RSU to another, it has to solve a series of computational puzzles. Due to the security of the hash functions, the computational hash puzzle cannot be solved with less than an estimated time. Besides, since the initial random seed of the puzzle, that is, $\mathcal{N}_0 := \mathcal{T}_{R_1}$ is generated by the previous RSU, the hash puzzle cannot be precomputed. Furthermore, all the subsequent random seeds depend on solutions to previous hash puzzles. Hence, the hash puzzle should be computed sequentially. This guarantees that it is difficult for a vehicle to generate two valid trajectory requests to a specific RSU.

Secondly, a rational attacker will not choose to outsource the computational puzzle. Every solution to a computational puzzle should be signed before it proceeds to the next computational puzzle. Hence, if a vehicle chooses to outsource the computational puzzle to the cloud service, it has to provide its secret key to the cloud. Otherwise, the vehicle has to communicate frequently with the cloud service, and the communication delay might be even longer than the time to solve one computational puzzle.

Overall, the proposed scheme is secure against Sybil attacks. □

In the following, we briefly argue that our scheme also satisfies the property of resisting RSU compromise and efficient RSU verification. □

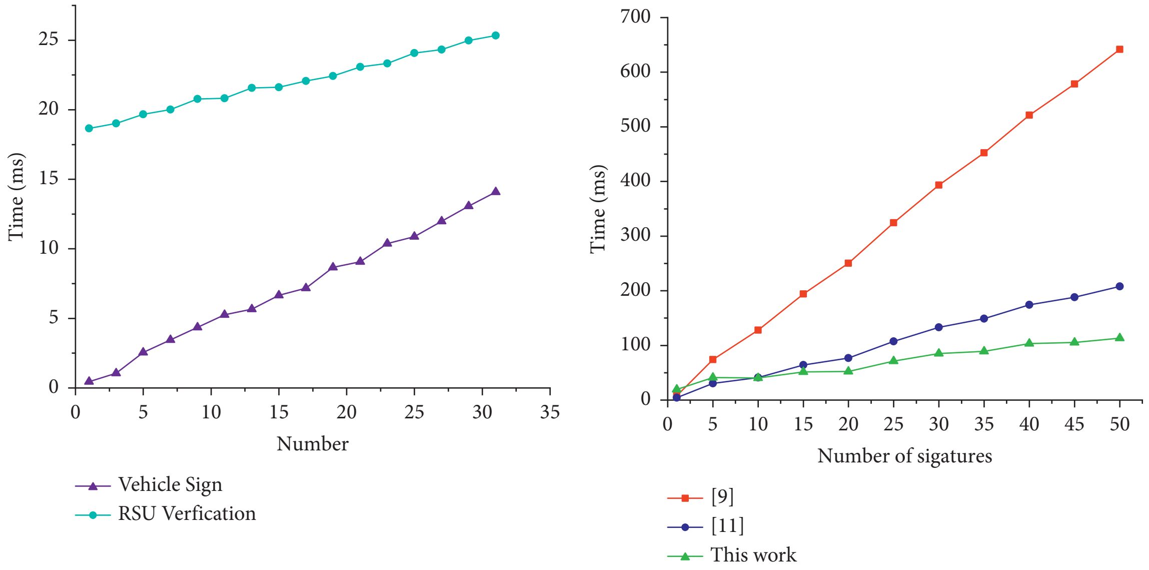
5.1.1. Resisting RSU Compromise. The security of the threshold signature scheme guarantees that the signature is unforgeable even if the adversary has $\theta - 1$ secret keys. Hence, even if $\theta - 1$ RSUs are compromised by an attacker, it is not possible to generate a forged trajectory.

5.1.2. Efficient RSU Verification. When verifying the validity of a trajectory request, the RSU needs to verify the correctness of the solution to the hash puzzle and the intermediate signatures. It is obvious that the puzzle verification is fast since only hash computation is required. Hence, the computational puzzle also guarantees the efficiency of RSU verification even under the DoS attacks. Besides, since the aggregate signature is adopted in our scheme, multiple signatures generated by the same vehicle can be verified efficiently.

Table 1: Theoretical comparison.

	Signature	Sign	Verify n signatures	PPA[†]	Sybil	RSU compromise
[9]	160 bytes	$6T_{\text{Exp}}$	$3nT_{\text{Pairing}}$	✓	✗	✗
[11]	35 bytes	T_{Exp}	nT_{Pairing}	✓	✓[‡]	✗
This work	96 bytes	$4T_{\text{Exp}}$	$2(n-1)T_{\text{Exp}} + 5T_{\text{Pairing}}$	✓	✓	✓

[†] PPA: privacy-preserving authentication; Sybil: Sybil attack resistance; RSU Compromise: security against RSU compromise attacks. [‡] It achieves Sybil attack resistance with the restriction of long enough trajectories.



(a)

(b)

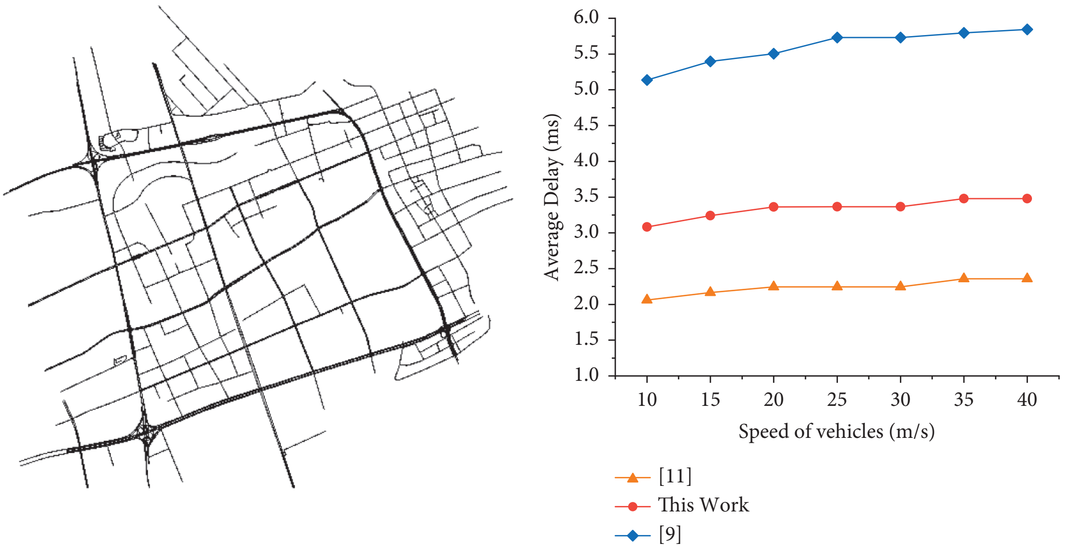
Figure 3: The computational costs. (a) The cost of sign and verify. (b) Comparison.

5.2. Theoretical Comparison. In Table 1, we compare the efficiency and security of the proposed scheme with the result of Wu et al. [9] and Baza et al. [11]. For efficiency, we compare the signature size and computational cost of message announcement and verification. T_{Exp} and T_{Pairing} refer to the running time of exponential computation in $\mathbb{G}_1$ and pairing computation. From the benchmark result, the running time of T_{Pairing} is around 10 times that of T_{Exp}. The signature size and signature generation cost of our scheme are comparable with related works. When verifying multiple signatures, the computational cost of our proposal is lower than that of [9, 11]. Besides, compared with the existing result, our scheme achieves Sybil attack resistance without requiring long enough trajectories.

5.3. Efficiency Evaluation. To evaluate the efficiency of our scheme, we first perform a simulation to evaluate the computational cost of signature generation and verification. Besides, we compare the signature verification cost of our proposal with that of Wu et al. [9] and Baza et al. [11]. The simulations were run on a Linux machine with an Intel Core i7-4790 processor @ 3.6 GHz. The bilinear group is implemented by a BN Curve with a 128-bit security level. We use the MIRACL library to implement the cryptographic algorithms of our scheme. We simulate the stages where a vehicle requests a trajectory, and the RSU verifies the signature of vehicles. In the trajectory request generation phase, we only evaluate the signature generation time since the time for solving a puzzle is almost fixed.

Figure 3(a) shows the computational cost of generating and verifying signatures of our scheme. When the number of puzzles a vehicle that needs to solve varies from 1 to 31, the signature generation time grows from 0.45 milliseconds to 14 milliseconds. Meanwhile, the cost of signature verification grows much slower than the signature generation time. When the number of puzzles grows from 1 to 31, the signature verification time for RSU grows from 18.6 ms to 26.3 ms. Figure 3(b) compares the computation cost for verifying n signatures of our proposal with that of Wu et al. [9] and Baza et al. [11]. It is shown that when the number of signatures is larger than 10, our proposal has a lower computational cost than the result of [9, 11].

We then use MOSAIC to simulate the communication delay of the proposed scheme and compare our result with that of [9, 11]. The road scenario is shown in Figure 4(a), which is an area of $5 \times 10\,\text{km}^2$ in Shanghai, China. The average speed of vehicles grows from 10 m/s to 40 m/s. The number of vehicles is set to be 100. The time for each simulation is 400 s. Similar to [9], the communication delay is computed as the average delay of each message for each vehicle. Figure 4(b) shows the simulation result. At the cost



(a)

(b)

Figure 4: The communication delay. (a) The road scenario. (b) The simulation result.

of realizing a stronger security guarantee, the communication delay of our proposal is slightly higher than [11]. However, the highest delay of our proposal in the simulation is less than 4 ms.

6. Conclusion

Privacy-preserving authentication and Sybil attack defense are major challenges in IoV. In this work, we propose to use verified trajectories as the identities of vehicles to achieve privacy-preserving authentication and Sybil attack resistance. Since all the trajectories are masked, the privacy of vehicles is preserved. Furthermore, with the help of trajectories, Sybil attacks can be efficiently detected. Benefitting from our proposed puzzle chains, our scheme is secure against Sybil attacks in the presence of cloud service assistant attackers. Finally, the efficient evaluation shows that the computational overhead of our scheme is acceptable.

Data Availability

Data are available from the corresponding author on request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the Shanghai Sailing Program (21YF1413800 and 20YF1413700), the National Science Foundation of China (no. 62002213), and the Program of Industrial Internet Visualized Asset Management and Operation Technology and Products.

References

[1] S. Chen, J. Hu, Y. Shi et al., “Vehicle-to-everything (v2x) services supported by lte-based systems and 5g,” *IEEE Communications Standards Magazine*, vol. 1, no. 2, pp. 70–76, 2017.

[2] F. Li, K. Y. Lam, Z. Sheng, W. Lu, X. Liu, and L. Wang, “Agent-based spectrum management scheme in satellite communication systems,” *IEEE Transactions on Vehicular Technology*, vol. 70, no. 3, pp. 2877–2881, 2021.

[3] K.-Y. Lam, S. Mitra, F. Gondesen, and X. Yi, “Ant-centric iot security reference architecture–security-by-design for satellite-enabled smart cities,” *IEEE Internet of Things Journal*, 2021.

[4] A. Chattopadhyay, K. Y. Lam, and Y. Tavva, “Autonomous Vehicle: Security by Design,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, 2020.

[5] L. Miao, J. J. Virtusio, and K. L. Hua, “Pc5-based cellular-v2x evolution and deployment,” *Sensors*, vol. 21, no. 3, 2021.

[6] S. Chen, J. Hu, Y. Shi, and L. Zhao, “Lte-v: a td-lte-based v2x solution for future vehicular network,” *IEEE Internet of Things Journal*, vol. 3, no. 6, pp. 997–1005, 2016.

[7] S. Chen, J. Hu, Y. Shi, L. Zhao, and W. Li, “A vision of c-v2x: technologies, field testing, and challenges with Chinese development,” *IEEE Internet of Things Journal*, vol. 7, no. 5, pp. 3872–3881, 2020.

[8] X. Li, L. Cheng, C. Sun, K.-Y. Lam, X. Wang, and F. Li, “Federated-learning-empowered collaborative data sharing for vehicular edge networks,” *IEEE Network*, vol. 35, no. 3, pp. 116–124, 2021.

[9] Q. Wu, J. Domingo-Ferrer, and U. González-Nicolás, “Balanced trustworthiness, safety, and privacy in vehicle-to-vehicle communications,” *IEEE Transactions on Vehicular Technology*, vol. 59, no. 2, pp. 559–573, 2009.

[10] J. Liu, J. Li, L. Zhang et al., “Secure intelligent traffic light control using fog computing,” *Future Generation Computer Systems*, vol. 78, pp. 817–824, 2018.

[11] M. Baza, M. Nabil, and M. E. Abdelsalam Mahmoud, “Detecting sybil attacks using proofs of work and location in

vanets," *IEEE Transactions on Dependable and Secure Computing*, 2020.

[12] S. Chang, Y. Qi, H. Zhu, J. Zhao, and X. Shen, "Footprint: detecting sybil attacks in urban vehicular networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 23, no. 6, pp. 1103–1114, 2012.

[13] J. Cui, J. Zhang, H. Zhong, and Y. Xu, "Spacf: a secure privacy-preserving authentication scheme for vanet with cuckoo filter," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 11, Article ID 10283, 2017.

[14] S. A. Chaudhry, "Designing an efficient and secure message exchange protocol for internet of vehicles," *Security and Communication Networks*, vol. 2021, Article ID 5554318, 9 pages, 2021.

[15] I. Ali and F. Li, "An efficient conditional privacy-preserving authentication scheme for vehicle-to-infrastructure communication in vanets," *Vehicular Communications*, vol. 22, Article ID 100228, 2020.

[16] R. Lu, X. Lin, H. Zhu, P. H. Ho, and X. Shen, "Ecpp: efficient conditional privacy preservation protocol for secure vehicular communications," in *Proceedings of the IEEE INFOCOM 2008-The 27th Conference on Computer Communications*, pp. 1229–1237, IEEE, Phoenix, AZ, USA, April 2008.

[17] X. Xiaodong Lin, X. Xiaoting Sun, P. H. Xuemin Shen, and X. Shen, "GSIS: a secure and privacy-preserving protocol for vehicular communications," *IEEE Transactions on Vehicular Technology*, vol. 56, no. 6, pp. 3442–3456, 2007.

[18] Y. Hao, Y. Cheng, and K. Ren, "Distributed key management with protection against rsu compromise in group signature based vanets," in *Proceedings of the IEEE GLOBECOM 2008-2008 IEEE global telecommunications conference*, pp. 1–5, IEEE, New Orleans, LA, USA, December 2008.

[19] Y. Sun, Z. Feng, Q. Hu, and J. Su, "An efficient distributed key management scheme for group-signature based anonymous authentication in VANET," *Security and Communication Networks*, vol. 5, no. 1, pp. 79–86, 2012.

[20] C. Lin, D. He, X. Huang, N. Kumar, and K. K. R. Choo, "Bcppa: a blockchain-based conditional privacy-preserving authentication protocol for vehicular ad hoc networks," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, 2020.

[21] M. B. Mollah, J. Zhao, D. Niyato et al., "Blockchain for the internet of vehicles towards intelligent transportation systems: a survey," *IEEE Internet of Things Journal*, vol. 8, no. 6, pp. 4157–4185, 2020.

[22] L. Chen, S. L. Ng, and G. Wang, "Threshold anonymous announcement in vanets," *IEEE Journal on Selected Areas in Communications*, vol. 29, no. 3, pp. 605–615, 2011.

[23] K. Lee, D. H. Lee, and M. Yung, "Aggregating cl-signatures revisited: extended functionality and better efficiency," in *Proceedings of the International Conference on Financial Cryptography and Data Security*, pp. 171–188, Springer, Okinawa, Japan, April 2013.

[24] D. Boneh, B. Lynn, and H. Shacham, "Short signatures from the weil pairing," in *Proceedings of the International Conference on the Theory and Application of Cryptology and Information Security*, pp. 514–532, Springer, Singapore, December 2001.

[25] R. Gennaro, S. Jarecki, H. Krawczyk, and T. Rabin, "Secure distributed key generation for discrete-log based cryptosystems," *Journal of Cryptology*, vol. 20, no. 1, pp. 51–83, 2007.

Hindawi
Security and Communication Networks
Volume 2021, Article ID 1468007, 9 pages
https://doi.org/10.1155/2021/1468007

Research Article

Permutation-Based Lightweight Authenticated Cipher with Beyond Conventional Security

Ping Zhang

School of Computer Science, Nanjing University of Posts and Telecommunications, Nanjing 210023, China

Correspondence should be addressed to Ping Zhang; zhgp@njupt.edu.cn

Received 8 July 2021; Revised 27 September 2021; Accepted 6 October 2021; Published 27 October 2021

Academic Editor: Huaxiong Wang

Lightweight authenticated ciphers are specially designed as authenticated encryption (AE) schemes for resource-constrained devices. Permutation-based lightweight authenticated ciphers have gained more attention in recent years. However, almost all of permutation-based lightweight AE schemes only ensure conventional security, i.e., about $c/2$-bit security, where c is the capacity of the permutation. This may be vulnerable for an insufficiently large capacity. This paper focuses on the stronger security guarantee and the better efficiency optimization of permutation-based lightweight AE schemes. On the basis of APE series (APE, APE^{RI}, APE^{OW}, and APE^{CA}), we propose a new improved permutation-based lightweight online AE mode APE^{+} which supports beyond conventional security and concurrent absorption. Then, we derive a simple security proof and prove that APE^{+} enjoys at most about $\min\{r, c\}$-bit security, where r is the rate of the permutation. Finally, we discuss the properties of APE^{+} on the hardware implementation.

1. Introduction

With the widespread rise of the big data, Internet of Things (IoT), and fifth generation (5G) and beyond 5G (B5G) networks, leaks of sensitive data from wireless sensor devices and network platforms have become more serious and more common. The collection of sensitive data has become one of the important targets of cyberattacks by hackers. How can we protect the security of our sensitive data? Cryptography is an important method to protect the security of sensitive data.

Lightweight cryptography focuses on the symmetric-key cryptography, whose goal is to settle the data security of resource-constrained devices in the embedded systems, sensor networks, RFID, and low-cost environments. The feature of the lightweight cryptography is that the implementation costs of hardware devices (such as areas, footprints, latency, and throughput) are as low as possible and the implementation efficiency (rate) is as high as possible, without sacrificing security guarantee.

The research of the lightweight cryptography began in 2004 and has been going on for more than a decade. The lightweight cryptography mainly includes the lightweight cipher and its modes of operation. Lightweight ciphers are designed to protect the privacy (confidentiality) of sensitive data on lightweight devices. Up to now, a large number of lightweight ciphers have been proposed, analyzed, and implemented [1–9]. Lightweight authenticated encryption (AE) modes of operation, also called lightweight authenticated ciphers, achieve both the privacy protection of sensitive data and the integrity verification of all data on lightweight devices. Competition for Authenticated Encryption: Security, Applicability, and Robustness (CAESAR) held in 2013 greatly contributed to the vigorous development of lightweight AE modes and produced many excellent schemes, such as Ascon [10] and ACORN [10]. From the perspective of the design method, lightweight AE modes include block-cipher-based lightweight AE modes [11–14], stream-cipher-based lightweight AE modes [15, 16], permutation-based lightweight AE modes [17–20], and hash-based lightweight AE modes [19, 20]. Moreover, permutation-based lightweight AE modes have more advantages and attractions than others due to its simple structure, convenient lookup table, and fast running speed.

Authenticated permutation-based encryption (APE) is the first permutation-based lightweight AE mode with nonce-misuse resistance designed by Andreeva et al. [17]. The idea is inspired from Sponge. The encryption algorithm of APE is online (i.e., the i-th block of ciphertext only depends on the first i blocks of plaintext), while the decryption algorithm is inverse-online (i.e., the online decryption of the ciphertext blocks is in reverse order). APE is proven up to the conventional security under the random permutation model (RPM), i.e., APE guarantees at most about $c/2$-bit security, where c is the capacity of the permutation.

However, there exist several drawbacks for APE, such as relatively big bandwidth, large hardware footprint, and high computational complexity. To overcome these drawbacks of APE, Sasaki and Yasuda focused on the implementation costs and the proper using of a nonce on resource-constrained devices [18]. On the basis of APE, they described three new online permutation-based lightweight AE modes, called APE^{RI}, APE^{OW}, and APE^{CA}, to meet the requirements of less bandwidths, smaller hardware footprints, and lower computational complexity. They proved that these three lightweight AE schemes also enjoy the conventional security.

Almost all of the previous permutation-based lightweight AE schemes, including APE, APE^{RI}, APE^{OW}, and APE^{CA}, only ensure at most about $c/2$-bit security. To ensure enough security, one tends to choose a permutation with a big capacity c. Table 1 shows security levels of some permutation-based AE modes using recommended parameters.

Table 1: Security levels of permutation-based AE modes using recommended parameters (b, r, c), where b is the permutation size, r is the rate of the permutation, c is the capacity of the permutation, and $b = r + c$.

Scheme	b	r	c	Security
Ascon [10]	320	128	192	96
	320	64	256	128
APE [17]	256	96	160	80
APE^{RI} [18]	256	96	160	80
APE^{OW} [18]	256	96	160	80
APE^{CA} [18]	256	96	160	80
Bettle [20]	144	64	80	64
	256	128	128	121
APE^{+}	256	96	160	96
256	256	128	128	128

Ascon includes two versions with four configurations (three with 128-bit security and one with 96-bit security). In this table, we just list two of them.

However, in some special environments, such as an insufficiently large capacity of the permutation or the partial information leakage of permutation by side channel attacks, this security bound is not enough. Moreover, the associated data and the message were handled separately in APE, APE^{RI}, APE^{OW}, and APE^{CA}, which is not highly efficient. Whether can we construct an efficient lightweight AE mode with beyond $c/2$-bit security?

This paper is devoted to solving the above problem and gives a positive response. On the basis of the current APE, APE^{RI}, APE^{OW}, and APE^{CA}, we propose a novel improved permutation-based lightweight online AE mode APE^{+}. APE^{+} supports strong security guarantee and high efficiency implementation. The concrete contributions include the following:

(1) In order to achieve higher efficiency, we consider to put some good factors into APE^{+}, such as inverse-free, stream-cipher encryption, concurrent absorption, and pure permutation. APE^{+} is inverse-free, i.e., the decryption algorithm of APE^{+} does not invoke the inverse of permutation. Besides, it is a stream-cipher encryption mode. For the associated data and the message, APE^{+} utilizes the method of concurrent absorption to process them, which makes the number of invoking the underlying permutation as few as possible. In particular, in view of the performance of APE^{+} on the hardware implementation, APE^{+} is built by the cascade method and has no backward feedback. Therefore, it can be fully pipeline implemented on the hardware. Moreover, APE^{+} just requires the forward permutation circuit for the encryption and decryption circuits. Therefore, the area of the hardware device and the number of the hardware footprints are minimized. APE^{+} utilizes the concurrent absorption method, which greatly reduces the computational complexity on the hardware devices.

(2) In order to achieve stronger security, the encryption and authentication parts are considered separately. For the encryption part, we utilize the iterated Even–Mansour cipher with a short key [21] to generate the ciphertext while avoiding the defeat that the current plaintext is XOR-ed with the previous ciphertext. For the authentication part, the authentication tag is generated by the XOR of the rate and the capacity of the last permutation to resist forgery attacks. In this paper, we derive a simple security proof by using a modular proof approach and prove that APE^{+} enjoys at most about $\min\{r, c\}$-bit AE security under the RPM assumption, where r and c are, respectively, the rate and the capacity of the permutation. Specifically, given a permutation with parameters $b = 256$, $r = 96$, and $c = 160$ (or $b = 256$, $r = 128$, and $c = 128$), APE^{+} enjoys at most about 96-bit (or 128-bit) AE security, which is shown in Table 1.

The rest of this paper is organized as follows. Notations and some preliminaries are presented in Section 2. Section 3 describes the security model of lightweight AE schemes. Section 4 provides a new permutation-based lightweight AE mode with beyond conventional security and derives a security proof. Section 5 shows some discussions for APE^{+}. Finally, Section 6 ends up with a conclusion.

2. Preliminaries

Notations. Let $\{0,1\}^*$ denote the set containing all finite bit strings (including the empty string). Let b be an integer and $\{0,1\}^b$ be the set of all strings whose lengths are b bits. For a finite string x, $|x|$ stands for its bit-length. For two finite strings x and y, let $x\|y$ or xy denote their concatenation and let $x \oplus y$ denote their bitwise XOR operation from the least bit to the most bit. If X is a set, let $x \overset{\$}{\leftarrow} X$ stand for that x is uniformly sampled from the finite set X. If a is a decimal, let $\lceil a \rceil$ be the smallest integer greater than or equal to a. Let $\Pr[\mathbf{A}|\mathbf{B}]$ be the conditional probability that event **A** occurs, giving event **B**.

Strong Pseudorandom Permutation (SPRP). One of the most important security concepts in symmetric ciphers is SPRP. What is SPRP? In a nutshell, if a symmetric cipher is indistinguishable from an ideal random permutation under chosen ciphertext attacks, then this symmetric cipher is an SPRP. The detailed definition is shown as follows.

Let E: $\mathscr{K} \times \{0,1\}^b \longrightarrow \{0,1\}^b$ be a symmetric cipher, where $\mathscr{K}$ is a nonempty key set. Then, for any $K \in \mathscr{K}$, $E_K(\cdot)$ is a permutation on b bits and $E_K^{-1}(\cdot)$ is the inverse of $E_K(\cdot)$. Let $\mathrm{Perm}(b)$ be the set of all permutations on b bits. Let P be a primitive utilized in E. Let $\mathscr{A}$ be an adversary with access to encryption, decryption, and the primitive and its inverse oracles, i.e., $(E_K^{\pm}, P^{\pm})$. Let $\mathscr{A}^{O} \Rightarrow 1$ be the event that an adversary $\mathscr{A}$ outputs 1 after interacting with the oracle O.

Let $K \overset{\$}{\leftarrow} \mathscr{K}$, $\pi \overset{\$}{\leftarrow} \mathrm{Perm}(b)$, then the SPRP advantage of $\mathscr{A}$ against E is defined as

$$\begin{aligned}\mathrm{Adv}_E^{\mathrm{sprp}}(\mathscr{A}) &= \left|\Pr\left[\mathscr{A}^{E_K^{\pm},P^{\pm}} \Rightarrow 1\right] - \Pr\left[\mathscr{A}^{\pi^{\pm},P^{\pm}} \Rightarrow 1\right]\right| \\ &= \Delta\left(E_K^{\pm}, P^{\pm}; \pi^{\pm}, P^{\pm}\right).\end{aligned} \tag{1}$$

If the advantage $\mathrm{Adv}_E^{\mathrm{sprp}}(\mathscr{A})$ is negligible, the cipher E_K is a secure strong pseudorandom permutation (SPRP).

If the resources (such as the overall running time t, the number of querying the encryption and decryption oracles q, the total query complexity of the construction σ, and the number of querying the primitive and its inverse oracles p) used by adversaries are limited, we define the maximum advantage as

$$\mathrm{Adv}_E^{\mathrm{sprp}}(t, q, \sigma, p) = \max_{\mathscr{A}} \mathrm{Adv}_E^{\mathrm{sprp}}(\mathscr{A}). \tag{2}$$

Even–Mansour Cipher with a Short Key [21]. Let P be a public random b-bit permutation, c be the capacity of P, r be the rate of P, and $b = r + c$. Let $\mathscr{K} = \{0,1\}^k$ be a k-bit key set. To minimize the key material of the Even–Mansour cipher and achieve beyond conventional security bound, the Even–Mansour cipher with a short key is presented. The Even–Mansour cipher with a short key is a function E: $\mathscr{K} \times \{0,1\}^b \longrightarrow \{0,1\}^b$ that inputs a key $K \in \mathscr{K}$ and a plaintext $x \in \{0,1\}^b$ and produces a ciphertext $y = E_K(x) = E(K,x) = P(x \oplus 0^r \| K) \oplus 0^r \| K$, where $k \le c$.

3. Security Model

Syntax of Authenticated Encryption (AE). Let $\mathscr{K}$, $\mathscr{N}$, $\mathscr{H}$, $\mathscr{M}$, $\mathscr{C}$, and $\mathscr{T}$ be, respectively, the sets of the keys, nonce, associated data, plaintexts, ciphertexts, and authentication tags. A nonce-based AE with associated data scheme $\Pi = (\mathscr{E}, \mathscr{D})$ consists of an encryption algorithm $\mathscr{E}$: $\mathscr{K} \times \mathscr{N} \times \mathscr{H} \times \mathscr{M} \longrightarrow \mathscr{C} \times \mathscr{T}$ and a decryption algorithm $\mathscr{D}$: $\mathscr{K} \times \mathscr{N} \times \mathscr{H} \times \mathscr{C} \times \mathscr{T} \longrightarrow \mathscr{M} \cup \{\perp\}$, where the symbol $\perp$ indicates the failure of the decryption oracle. Let $K \in \mathscr{K}$ be a key, $N \in \mathscr{N}$ be a nonce, $A \in \mathscr{H}$ be an associated data, $M \in \mathscr{M}$ be a plaintext, $C \in \mathscr{C}$ be a ciphertext, and $T \in \mathscr{T}$ be an authentication tag, then the syntax is formalized as follows:

$$\begin{aligned}(C,T) &\leftarrow \mathscr{E}_K(N,A,M), \\ \frac{M}{\perp} &\leftarrow \mathscr{D}_K(N,A,C,T),\end{aligned} \tag{3}$$

where $\mathscr{E}_K(N,A,M) = (C,T)$ if and only if $\mathscr{D}_K(N,A,C,T) = M$. A secure AE scheme returns $\perp$ if it receives an error (N,A,C,T) pair.

The nonce-based AE with associated data scheme $\Pi = (\mathscr{E}, \mathscr{D})$ is called as an online AE scheme (or authenticated online cipher) if and only if the j-th ciphertext block C_j only depends on the first j plaintext blocks $M_1, \ldots, M_j$, where $j = 1, \ldots, m = \lceil |M/r| \rceil$. That is to say, for any fixed key K, nonce N, and associated data A, if two plaintexts M and M' share an l-block common prefix, where $0 \le l \le m-1$, then their encrypted ciphertexts C and $C\prime$ also share an l-block common prefix. Therefore, a secure authenticated online cipher requires that ciphertexts do not reveal any further information about plaintexts than its length and the longest common prefix with previous plaintexts.

Ideal Online Function and Ideal Authenticated Online Cipher. Let f^j be a function randomly chosen from $\mathscr{N} \times \mathscr{H} \times \{0,1\}^{(j-1)r} \times \{0,1\}^s \longrightarrow \{0,1\}^s$, where $1 \le j \le m = \lceil |M|/r \rceil$ and $1 \le s \le r$. We define an ideal online function g: $\mathscr{N} \times \mathscr{H} \times \mathscr{M} \longrightarrow \mathscr{C}$ as follows:

$$\begin{aligned}C &= g(N,A,M) = \|_{j=1}^{m} f^j\left(N, A, M_1 \| \cdots \| M_{j-1}, M_j\right), \\ C_j &= f^j\left(N, A, M_1 \| \cdots \| M_{j-1}, M_j\right), \\ C &= C_1 \| \cdots \| C_m.\end{aligned} \tag{4}$$

Let t be a tag-generation function randomly chosen from $\mathscr{N} \times \mathscr{H} \times \mathscr{M} \longrightarrow \mathscr{T}$, and we define an ideal authenticated online cipher \$: $\mathscr{N} \times \mathscr{H} \times \mathscr{M} \longrightarrow \mathscr{C} \times \mathscr{T}$ as follows:

$$(C,T) = \$(N,A,M), \tag{5}$$

where $C = g(N,A,M)$ and $T = t(N,A,M)$.

AE Security Model. The security model of AE schemes includes the conventional security model (privacy and authenticity) [11, 17] and all-in-one AE security model [18, 22–24]. In fact, all-in-one AE security model covers the conventional privacy and authenticity security models. Therefore, we consider all-in-one AE security model. Let $\Pi = (\mathscr{E}, \mathscr{D})$ be an AE scheme. The all-in-one AE security model is defined as follows.

Definition 1 (AE security [24]). Let P be a public random permutation, K be a key, and $\Pi[P]$ be a P-based AE scheme. Let $q, \sigma, p > 0$. Then, the AE security advantage of the adversary is

$$\mathrm{Adv}^{\mathrm{ae}}_{\Pi[P]}(q,\sigma,p) = \left|\Pr\left[\mathscr{A}^{\mathscr{E}_K,\mathscr{D}_K,P^{\pm}} = 1\right] - \Pr\left[\mathscr{A}^{\$,\perp,\mathrm{P}^{\pm}} = 1\right]\right| = \Delta\left(\mathscr{E}_K,\mathscr{D}_K,P^{\pm};\$,\perp,P^{\pm}\right), \tag{6}$$

where q is the number of querying the encryption oracle $\mathscr{E}$ or the decryption oracle $\mathscr{D}$, generating at most σ blocks, p is the number of querying the permutation P or its inverse P^{-1}, \$ is an ideal authenticated online cipher, and $\perp$ stands for the failure of the decryption oracles.

4. APE^{+}: Authenticated Permutation-Based Encryption Scheme with Beyond Conventional Security for Lightweight Applications

In this section, we provide a new pure permutation-based lightweight online AE mode APE^{+} which enjoys beyond conventional security. Section 4.1 describes the specification of APE^{+}. Section 4.2 derives the security proofs of APE^{+}.

4.1. APE^{+}: Pure Permutation-Based Lightweight Authenticated Online Cipher. Let P be a public b-bit random permutation and $b = r + c$. Let $K \in \mathscr{K}$ be a key with k-bit, $N \in \mathscr{N}$ be a nonce, and $A \in \mathscr{H}$ be an associated data. Let $M = M_1\|M_2\|\cdots\|M_m \in \mathscr{M}$ be a plaintext, $C = C_1\|C_2\|\cdots\|C_m \in \mathscr{C}$ be the corresponding ciphertext, and $T \in \mathscr{T}$ be the corresponding authentication tag, where $m = \lceil |M|/r\rceil$ is the block length of the plaintext. Let τ be the bit-length of the tag and $\tau = k = c$.

To design a lightweight online AE mode with beyond conventional security, we utilize the iterated Even–Mansour cipher with a short key [21] to generate the ciphertext for the encryption part and invoke the Even–Mansour cipher with a short key [21] to generate the authentication tag for the authentication part. Moreover, to prevent forgery attacks, the rate of the last permutation is XOR-ed to the capacity of the last permutation with the short key to realize the authentication tag with a random mask. To make the number of invoking the underlying permutation as few as possible, we utilize the concurrent absorption method [25] to process the associated data and the message. The overview of APE^{+} is shown in Figure 1.

APE^{+} consists of an encryption algorithm $\mathscr{E}$ and a decryption algorithm $\mathscr{D}$. The encryption algorithm $\mathscr{E}$ takes as input a key K, a nonce N, an associated data A, and a plaintext M and returns a ciphertext C and a tag T. The decryption algorithm $\mathscr{D}$ takes K, N, A, C, and T as inputs and returns either M or $\perp$. The encryption and decryption algorithms are depicted in Algorithms 1 and 2.

4.2. Beyond Conventional Security of APE^{+}. APE, APE^{RI}, APE^{OW}, and APE^{CA} only ensure at most about $2^{c/2}$ adversarial queries (i.e., $c/2$-bit security). APE^{+} is a pure permutation-based lightweight AE scheme with beyond conventional security. Besides, APE^{+} is also an authenticated online cipher. In this section, we prove that APE^{+} enjoys at most about $\min\{r,c\}$-bit AE security. Let $\Pi[P] = (\mathscr{E},\mathscr{D})$ stand for our APE^{+} scheme with a permutation P.

Theorem 1. *Let $P \overset{\$}{\leftarrow} Perm(b)$ be a public b-bit random permutation and $b = r + c$. Then,*

$$\mathrm{Adv}^{\mathrm{ae}}_{\Pi[P]}(q,\sigma,p) \le \sqrt{\frac{ep\sigma}{2^b}} + \frac{1.5(\sigma+q)^2}{2^b} + \frac{2\sigma}{2^r} + \frac{q}{2^c}, \tag{7}$$

where $e = 2.71828182845\ldots$ is the base of the natural logarithm.

Proof. We utilize the modular proof approach. First, our scheme can be described as a scheme based on an Even–Mansour cipher with a short key E_K, i.e., $\Pi[P]$ can be represented as $\Pi[E_K]$, where K is the secret key. Then, we replace the Even–Mansour modular structure of our scheme by the random permutation Q and rename the new scheme as $\Pi[Q]$. There exists a nontrivial gap for this replacement. According to the definition of the AE security, we have

$$\begin{aligned}\mathrm{Adv}^{\mathrm{ae}}_{\Pi[P]}(q,\sigma,p) &= \Delta\left(\mathscr{E}_K,\mathscr{D}_K,P;\$,\perp,\mathrm{P}\right)\\ &= \Delta\left(\mathscr{E}[E_K],\mathscr{D}[E_K],P;\$,\perp,\mathrm{P}\right)\\ &\le \Delta\left(\mathscr{E}[E_K],\mathscr{D}[E_K],P;\mathscr{E}[Q],\mathscr{D}[Q],P\right)\\ &\quad + \Delta\left(\mathscr{E}[Q],\mathscr{D}[Q],P;\$,\perp,\mathrm{P}\right)\\ &\le \mathrm{Adv}^{sprp}_{E}(q,\sigma,p) + \mathrm{Adv}^{ae}_{\Pi[Q]}(q,\sigma,p).\end{aligned} \tag{8}$$

It follows that we need to calculate the upper bounds of $\mathrm{Adv}^{\mathrm{sprp}}_{E}(q,\sigma,p)$ and $\mathrm{Adv}^{\mathrm{ae}}_{\Pi[Q]}(q,\sigma,p)$. First, according to the advantage of the Even–Mansour cipher with a short key [21], we have

$$\mathrm{Adv}^{\mathrm{sprp}}_{E}(q,\sigma,p) = \Delta\left(\mathscr{E}[E_K],\mathscr{D}[E_K],P;\mathscr{E}[Q],\mathscr{D}[Q],P\right) \le \frac{\mu p}{2^c}, \tag{9}$$

where μ is the maximal multiplicity. Now, we consider the rationality of μ. The probability that the multiplicity exceeds μ is upper bounded by $\binom{\sigma}{\mu}(1/2^r)^{\mu-1}$, which is very close to zero. By Stirling's approximation, this probability is also bounded by $2^r(e\sigma/\mu 2^r)^{\mu}$, where $e = 2.71828182845\ldots$. Assume that $e\sigma/\mu 2^r = (ep\sigma/2^{r+c})^{1/2}$ and $16ep\sigma/2^{r+c} \ll 1$, and we have $\mu = (e\sigma\cdot 2^c/p\cdot 2^r)^{1/2}$. It follows that

$$\mathrm{Adv}^{\mathrm{sprp}}_{E}(q,\sigma,p) \le \left(\frac{ep\sigma}{2^b}\right)^{1/2}. \tag{10}$$

Then, we need to compute the following advantage:

$$A\,dv^{\mathrm{ae}}_{\Pi[Q]}(q,\sigma,p) = \Delta\left(\mathscr{E}[Q],\mathscr{D}[Q],P;\$,\perp,\mathrm{P}\right). \tag{11}$$

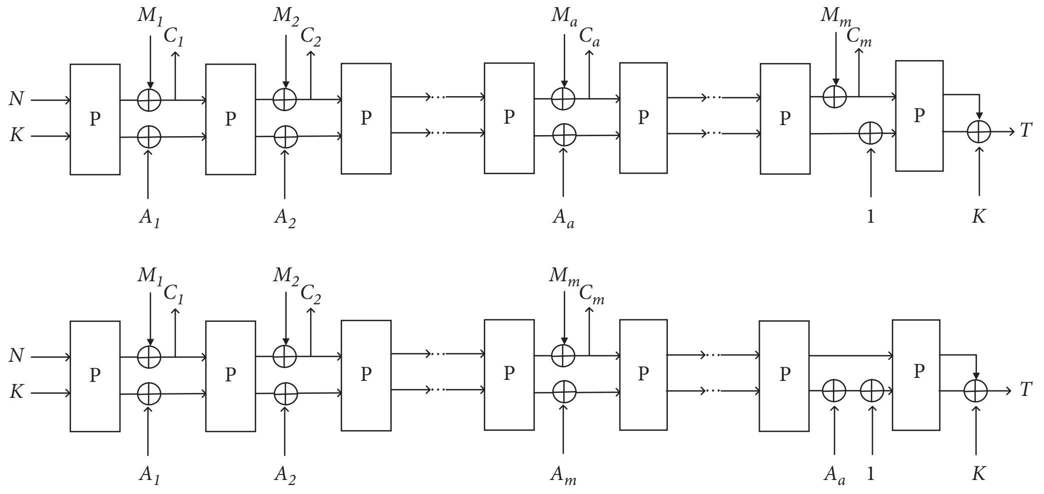


FIGURE 1: APE$^+$: permutation-based lightweight AE mode with beyond conventional security and concurrent absorption for a-block associated data and m-block plaintext (upper: $a \leq m$; lower: $a > m$).

```
**Input:** a key $K$, a nonce $N$, an associated data $A$,
  and a plaintext $M$
**Output:** a ciphertext $C$ and a tag $T$
Partition $M$ into $M_1 \| \cdots \| M_m$, $|M| = r, 1 \leq i \leq m$
Partition $A$ into$A_1 \| \cdots \| A_a$, $|A_j| = c, 1 \leq j \leq a$
$C_0 = N, V_0 = K$
**if** $a \leq m$ **then**
    **for** $0 \leq i \leq a - 1$ **do**
        $(K_{i+1}, W_{i+1}) = P(C_i, V_i)$
        $C_{i+1} = K_{i+1} \oplus M_{i+1}$
        $V_{i+1} = W_{i+1} \oplus A_{i+1}$
    **end for**
    **for** $a \leq i \leq m - 1$ **do**
        $(K_{i+1}, W_{i+1}) = P(C_i, V_i)$
        $C_{i+1} = K_{i+1} \oplus M_{i+1}$
        $V_{i+1} = W_{i+1}$
    **end for**
    $(K_{m+1}, W_{m+1}) = P(C_m, V_m \oplus 1)$
    $T = W_{m+1} \oplus K \oplus K_{m+1}$
**else**
    **for** $0 \leq i \leq m - 1$ **do**
        $(K_{i+1}, W_{i+1}) = P(C_i, V_i)$
        $C_{i+1} = K_{i+1} \oplus M_{i+1}$
        $V_{i+1} = W_{i+1} \oplus A_{i+1}$
    **end for**
    **for** $m \leq i \leq a - 1$ **do**
        $(K_{i+1}, W_{i+1}) P(C_i, V_i)$
        $C_{i+1} = K_{i+1}$
        $V_{i+1} = W_{i+1} \oplus A_{i+1}$
    **end for**
    $(K_{a+1}, W_{a+1}) = P(C_a, V_a \oplus 1)$
    $T = W_{a+1} \oplus K \oplus K_{a+1}$
**end if**
return $(C = C_1 \| C_2 \| \cdots \| C_{m-1} \| C_m, T)$
```

ALGORITHM 1: Encryption algorithm: $\mathscr{E}_K(N, A, M)$.

Now, we replace the random permutation Q by the random function f and rename the new scheme as $\Pi[f]$. According the hybrid argument and the RP/RF switch lemma, we have

```
**Input:** a key $K$, a nonce $N$, an associated data $A$,
  a ciphertext $C$, and a tag $T$
**Output:** a plaintext $M$ or $\perp$
Partition $C$ into $C_1 \| \cdots \| C_m$, $|C_i| = r, 1 \leq i \leq m$
Partition $A$ into $A_1 \| \cdots \| A_a$, $|A_j| = c, 1 \leq j \leq a$
$C_0 = N, V_0 = K$
**if** $a \leq m$ **then**
    **for** $0 \leq i \leq a - 1$ **do**
        $(K_{i+1}, W_{i+1}) = P(C_i, V_i)$
        $M_{i+1} = K_{i+1} \oplus C_{i+1}$
        $V_{i+1} = W_{i+1} \oplus A_{i+1}$
    **end for**
    **for** $a \leq i \leq m - 1$ **do**
        $(K_{i+1}, W_{i+1}) = P(C_i, V_i)$
        $M_{i+1} = K_{i+1} \oplus C_{i+1}$
        $V_{i+1} = W_{i+1}$
    **end for**
    $(K_{m+1}, W_{m+1}) = P(C_m, V_m \oplus 1)$
    $T' = W_{m+1} \oplus K \oplus K_{m+1}$
**else**
    **for** $0 \leq i \leq m - 1$ **do**
        $(K_{i+1}, W_{i+1}) = P(C_i, V_i)$
        $M_{i+1} = K_{i+1} \oplus C_{i+1}$
        $V_{i+1} = W_{i+1} \oplus A_{i+1}$
    **end for**
    **for** $m \leq i \leq a - 1$ **do**
        $(K_{i+1}, W_{i+1}) = P(C_i, V_i)$
        $C_{i+1} = K_{i+1}$
        $V_{i+1} = W_{i+1} \oplus A_{i+1}$
    **end for**
    $(K_{a+1}, W_{a+1}) = P(C_a, V_a \oplus 1)$
    $T' = W_{a+1} \oplus K \oplus K_{a+1}$
**end if**
**if** $T' = T$ **then**
    return $M = M_1 \| M_2 \| \cdots \| M_{m-1} \| M_m$
**else**
    return $\perp$ (INVALID)
**end if**
```

ALGORITHM 2: Decryption algorithm: $\mathscr{D}_K(N, A, C, T)$.

$$\begin{aligned}\mathrm{Adv}^{\mathrm{ae}}_{\Pi[Q]}(q,\sigma,p) &= \Delta(\mathscr{E}[Q],\mathscr{D}[Q],P;\$,\perp,\mathrm{P})\\ &\leq \Delta(\mathscr{E}[Q],\mathscr{D}[Q],P;\mathscr{E}[f],\mathscr{D}[f],P)\\ &\quad+\Delta(\mathscr{E}[f],\mathscr{D}[f],P;\$,\perp,\mathrm{P})\\ &\leq \frac{(\sigma+q)^2}{2^{b+1}}+\Delta(\mathscr{E}[f],\mathscr{D}[f],P;\$,\perp,\mathrm{P}).\end{aligned} \tag{12}$$

Next, we need to evaluate $\Delta(\mathscr{E}[f],\mathscr{D}[f],P;\$,\perp,\mathrm{P})$. According to the definitions of privacy and authenticity [17], we have

$$\begin{aligned}&\Delta(\mathrm{E}[f],\mathscr{D}[f],P;\$,\perp,P)\\ &\quad\leq \Delta(\mathrm{E}[f],\mathscr{D}[f],P;\mathrm{E}[f],\perp,P)+\Delta(\mathrm{E}[f],\perp,P;\$,\perp,P)\\ &\quad= \Delta(\mathrm{E}[f],\mathscr{D}[f],P;\mathrm{E}[f],\perp,P)+\Delta(\mathrm{E}[f],P;\$,P)\\ &\quad= \mathrm{Adv}^{\mathrm{auth}}_{\Pi[f]}(q,\sigma,p)+\mathrm{Adv}^{\mathrm{priv}}_{\Pi[f]}(q,\sigma,p),\end{aligned} \tag{13}$$

where

$$\begin{aligned}\mathrm{Adv}^{\mathrm{auth}}_{\Pi[f]}(q,\sigma,p) &= \Delta(\mathscr{E}[f],\mathscr{D}[f],P;\mathscr{E}[f],\perp,P),\\ \mathrm{Adv}^{\mathrm{priv}}_{\Pi[f]}(q,\sigma,p) &= \Delta(\mathscr{E}[f],P;\$,\mathrm{P}).\end{aligned} \tag{14}$$

In the first step, we calculate the PRIV advantage $\mathrm{Adv}^{\mathrm{priv}}_{\Pi[f]}(q,\sigma,p)$. Assume that the adversary queries $(N^1,A^1,M^1),\ldots,((N^q,A^q,M^q)$ to the encryption oracle $\mathscr{E}[f]$ and gains the corresponding responses $(C^1,T^1),\ldots,(C^q,T^q)$. Here, the adversary is deterministic and adaptive, i.e., each query of the adversary $(N^{w+1},A^{w+1},M^{w+1})$ is completely determined by the previous query-response pairs $\{(N^1,A^1,M^1,C^1,T^1),\ldots,(N^w,A^w,M^w,C^w,T^w)\}$, where $1\leq w\leq q-1$ and $(N^1,A^1,M^1),\ldots,(N^q,A^q,M^q)$ are distinct.

Let us define some symbols for the i-th encryption query-response pair (N^i,A^i,M^i,C^i,T^i), where $1\leq i\leq q$. Let $a^i=\lceil|A^i|/c\rceil$ and $m^i=\lceil|M^i|/r\rceil$ be, respectively, the block lengths of the associated data A^i and the plaintext M^i. Then, $A^i=A^i_1\|A^i_2\|\cdots\|A^i_{a^i}$ and $M^i=M^i_1\|M^i_2\|\cdots\|M^i_{m^i}$. Here, we assume that the block length of the associated data is always less than or equal to the block length of the plaintext. Let $I^i_0=(N^i,0),I^i_1=(C^i_1,V^i_1),\ldots,I^i_{a^i}=(C^i_{a^i},V^i_{a^i}),\ldots,I^i_{m^i}=(C^i_{m^i},V^i_{m^i}\oplus 1)$ and $O^i_1=(K^i_1,W^i_1),\ldots,O^i_{a^i}=(K^i_{a^i},W^i_{a^i})$, $O^i_{a^i+1}=(K^i_{a^i+1},V^i_{a^i+1}),\ldots,O^i_{m^i}=(K^i_{m^i},V^i_{m^i}),O^i_{m^i+1}=(K^i_{m^i+1},T^i\oplus K^i_{m^i+1})$ be the inputs and outputs of the random function f, where $C^i_s{=}K^i_s\oplus M^i_s$ for $1{\leq}s{\leq}m^i$ and $V^i_t{=}W^i_t\oplus A^i_t$ for $1{\leq}t{\leq}a^i$.

We define an event **Coll** that stands for a collision between the inputs of the random function f. For an authenticated online cipher, we consider that any two distinct queries $(N^i,A^i,M^i)\neq(N^j,A^j,M^j)$ share a common prefix, where $1\leq i\neq j\leq q$. The adversary is nonce-misuse; therefore, $N^i=N^j=N$ is a common prefix. We consider the following cases:

Case 1: if $A^i=A^j=A$ is fully common, then $M^i\neq M^j$. Assume that M^i and M^j have an α-longest common prefix, i.e., $M^i_1\|\cdots\|M^i_\alpha=M^j_1\|\cdots\|M^j_\alpha$ and $M^i_{\alpha+1}\neq M^j_{\alpha+1}$, where $\alpha\geq 0$ ($\alpha=0$ means $M^i_1\neq M^j_1$). Therefore, $I^i_0\|\cdots\|I^i_\alpha=I^j_0\|\cdots\|I^j_\alpha$ and $I^i_{\alpha+1}\neq I^j_{\alpha+1}$. The event **Coll** occurs if one of the following collisions happens:

(1) $I^i_{\alpha+1}=I^j_t$ for $t{\neq}\alpha+1$, where $1\leq i{\neq}j\leq q$.

(2) $I^i_s=I^j_t$ for $\alpha+2\leq s\leq m^i, 1\leq t\leq m^j$, where $1\leq i{\neq}j\leq q$.

(3) $I^i_s=I^i_t$ for $1\leq s{\neq}t\leq m^i$, where $1\leq i\leq q$.

(4) $I^i_s=I^i_0=I^j_0=(N^i,0)$ for $1\leq s\leq m^i$, where $1\leq i{\neq}j\leq q$.

Let l be the maximum block length of the plaintext, i.e., $m^i\leq l$ and $m^j\leq l$, and let $\sigma=ql$. Therefore, after removing the duplicate conditions, the probability that the event **Coll** occurs is

$$\begin{aligned}\Pr[\mathbf{Coll}] &= \sum_{1\leq i\neq j\leq q}\sum_{t\neq\alpha+1}\frac{1}{2^b}+\sum_{1\leq i\neq j\leq q}\sum_{\alpha+2\leq s\leq m^i}\sum_{1\leq t\leq m^j}\frac{1}{2^b}+\sum_{i=1}^{q}\sum_{1\leq s\neq t\leq m^i}\frac{1}{2^b}+\sum_{i=1}^{q}\sum_{s=1}^{m^i}\frac{1}{2^r}\\ &\leq \sum_{1\leq i\neq j\leq q}\frac{(l-1)+l(l-2)}{2^b}+\sum_{i=1}^{q}\frac{l(l-1)/2}{2^b}+\sum_{i=1}^{q}\sum_{s=1}^{l}\frac{1}{2^r}\left(\mathrm{As}\ I^i_{s-1}=\left(*,T^i\right)\right)\leq\frac{(q+\sigma)^2}{2^{b+1}}+\frac{\sigma}{2^r}.\end{aligned} \tag{15}$$

Case 2: if $A^i\neq A^j$ but A^i and A^j have an α-longest common prefix, then $A^i_1\|\cdots\|A^i_\alpha=A^j_1\|\cdots\|A^j_\alpha$ and $A^i_{\alpha+1}\neq A^j_{\alpha+1}$, where $\alpha\geq 0$. We assume that M^i and M^j have a β-longest common prefix, where $\beta\geq 0$. Then, $M^i_1\|\cdots\|M^i_\beta=M^j_1\|\cdots\|M^j_\beta$ and $M^i_{\beta+1}\neq M^j_{\beta+1}$.

Case 2.1: if $\beta\geq\alpha$, then $(A^i_{\alpha+1},M^i_{\alpha+1})\neq(A^j_{\alpha+1},M^j_{\alpha+1})$. Therefore, $I^i_0\|\cdots\|I^i_\alpha=I^j_0\|\cdots\|I^j_\alpha$ and $I^i_{\alpha+1}\neq I^j_{\alpha+1}$. The probability that the event **Coll** occurs is the same with Case 1.

Case 2.2: if $\beta<\alpha$, then $(A^i_{\beta+1},M^i_{\beta+1})\neq(A^j_{\beta+1},M^j_{\beta+1})$. Therefore, $I^i_0\|\cdots\|I^i_\beta=I^j_0\|\cdots\|I^j_\beta$ and $I^i_{\beta+1}\neq I^j_{\beta+1}$. The event **Coll** occurs if one of the following collisions happens:

(1) $I^i_{\beta+1}=I^j_t$ for $t{\neq}\beta+1$, where $1\leq i{\neq}j\leq q$.

(2) $I^i_s=I^j_t$ for $\beta+2\leq s\leq m^i$ and $1\leq t\leq m^j$, where $1\leq i{\neq}j\leq q$.

(3) $I^i_s=I^i_t$ for $1\leq s{\neq}t\leq m^i$, where $1\leq i\leq q$.

(4) $I_s^i = I_0^i = I_0^j = (N^i, 0)$ for $1 \le s \le m^i$, where $1 \le i \ne j \le q$.

It follows that, in Case 2.2, the probability that the event **Coll** occurs is

$$\begin{aligned}\Pr[\mathbf{Coll}] &= \sum_{1\le i\ne j\le q}\sum_{t\ne\beta+1}\frac{1}{2^b} + \sum_{1\le i\ne j\le q}\sum_{\beta+2\le s\le m^i}\sum_{1\le t\le m^j}\frac{1}{2^b} + \sum_{i=1}^{q}\sum_{1\le s\ne t\le m^i}\frac{1}{2^b} + \sum_{i=1}^{q}\sum_{s=1}^{m^i}\frac{1}{2^r} \\ &\le \sum_{1\le i\ne j\le q}\frac{(l-1)+l(l-2)}{2^b} + \sum_{i=1}^{q}\frac{l(l-1)/2}{2^b} + \sum_{i=1}^{q}\sum_{s=1}^{l}\frac{1}{2^r}\left(\text{As } I_{s-1}^i = \left(*, T^i\right)\right) \le \frac{(q+\sigma)^2}{2^{b+1}} + \frac{\sigma}{2^r}.\end{aligned} \tag{16}$$

Summarizing the above mutually exclusive cases, the probability that the event **Coll** occurs is

$$\Pr[\mathbf{Coll}] \le \frac{(q+\sigma)^2}{2^{b+1}} + \frac{\sigma}{2^r}. \tag{17}$$

If the event **Coll** does not occur, all inputs of f are fresh, except that the inputs from the common prefix are equal. Therefore, $\mathscr{E}[f]$ is indistinguishable from \$. In the nonce-misuse setting, we have

$$\mathrm{Adv}_{\Pi[f]}^{\mathrm{priv}}(q,\sigma,p) \le \Pr[\mathbf{Coll}] \le \frac{(q+\sigma)^2}{2^{b+1}} + \frac{\sigma}{2^r}. \tag{18}$$

In the second step, we evaluate the AUTH advantage $\mathrm{Adv}_{\Pi[f]}^{\mathrm{auth}}(q,\sigma,p)$. Assume that the adversary makes q_d nontrivial forgery attempts $(N'^1, A'^1, C'^1, T'^1), \ldots, (N'^{q_d}, A'^{q_d}, C'^{q_d}, T'^{q_d})$ to the decryption oracle $\mathscr{D}[f]$ after querying q_e encryption oracles, where $(N'^1, A''^1, C'^1, T'^1), \ldots, (N'^{q_d}, A'^{q_d}, C'^{q_d}, T'^{q_d}) \notin \{(N^1, A^1, C^1, T^1), \ldots, (N^{q_e}, A^{q_e}, C^{q_e}, T^{q_e})\}$ and $q = q_e + q_d$. Here, we define an event **Forge** that some decryption queries among q_d forgery attempts do not return $\perp$. If the event **Forge** does not occur, the responses of querying $(\mathscr{E}[f], \mathscr{D}[f])$ and $(\$, \perp)$ are identical. Therefore, by the total probability formula, we have

$$\begin{aligned}\mathrm{Adv}_{\Pi[f]}^{\mathrm{auth}}(q,\sigma,p) &\le \Pr[\mathbf{Forge}] \\ &= \Pr[\mathbf{Forge}|\mathbf{Coll}]\Pr[\mathbf{Coll}] \\ &\quad + \Pr[\mathbf{Forge}|\neg\mathbf{Coll}]\Pr[\neg\mathbf{Coll}] \\ &\le \Pr[\mathbf{Coll}] + \Pr[\mathbf{Forge}|\neg\mathbf{Coll}].\end{aligned} \tag{19}$$

The probability that the event **Coll** happens is similar to the PRIV advantage except that we need to consider an extra query complexity—the decryption query complexity under the forgery attempts, i.e., $\Pr[\mathbf{Coll}] \le (q+\sigma)^2/2^{b+1} + \sigma/2^r$, where σ is the total query complexity of the encryption and decryption queries.

To compute the probability $\Pr[\mathbf{Forge}|\neg\mathbf{Coll}]$, we consider the following cases:

Case 1: T'^i is new, i.e., $T'^i \notin \{T^1, \ldots, T^{q_e}\}$, where $1 \le i \le q_d$. For each forgery attempt, the probability of correctly guessing the image of a new point for the adversary is at most $1/(2^c - q_e)$.

Case 2: T'^i is old, but (N'^i, A'^i, C'^i) is new. We further analyze this case as follows.

Case 2.1: N'^i is new, i.e., $N'^i \notin \{N^1, \ldots, N^{q_e}\}$. The image of this new point under a new random function is uniform, random, and independent. Therefore, the probability for correctly guessing the tag T'^i is at most $1/2^c$.

Case 2.2: N'^i is old, but (A'^i, C'^i) is new. Under the condition of the event $\neg$**Coll**, the input of the last random function f is new. The outputs of f with distinct inputs are random and independent. Therefore, the probability for correctly guessing the same tag is at most $1/2^c$.

Summarizing the above two cases, the successful probability of q_d forgery attempts is upper bounded by $q_d/(2^c - q_e)$.

Therefore, according the sugar water inequality $a/b \le a + m/b + m$, where $b > a > 0$ and $m \ge 0$, and $q = q_e + q_d$, we have

$$\begin{aligned}\mathrm{Adv}_{\Pi[f]}^{\mathrm{auth}}(q,\sigma,p) &\le \Pr[\mathbf{Forge}] \\ &\le \Pr[\mathbf{Coll}] + \Pr[\mathbf{Forge}|\neg\mathbf{Coll}] \\ &\le \frac{(q+\sigma)^2}{2^{b+1}} + \frac{\sigma}{2^r} + \frac{q}{2^c}.\end{aligned} \tag{20}$$

Therefore, combining (1)–(6), we can obtain the result of Theorem 1.

According to Theorem 1, the AE security of APE$^+$ is up to $2^{\min\{b/2, r, c\}} = 2^{\min\{r, c\}}$ adversarial queries against nonce-misusing adversaries. In other words, APE$^+$ ensures at most about $\min\{r, c\}$-bit AE security, which is a beyond conventional ($c/2$-bit) security. □

5. Discussions

The original intention of designing our APE$^+$ scheme is to achieve higher efficiency, better performance, and stronger security on the lightweight devices. APE$^+$ is an improved version of APE series (including APE, APE^{RI}, APE^{OW}, and APE^{CA}). Therefore, APE$^+$ inherits most of the advantages of APE series. Besides, it has the following advantages in the hardware implementation:

(1) APE$^+$ is a pure permutation-based lightweight online AE mode with concurrent absorption. The rate of

Table 2: Comparison of permutation-based AE modes. Let $X = |A| + |M|$, $n = \lceil |N|/r \rceil$, $a = \lceil |A|/c \rceil$, $m = \lceil |M|/r \rceil$, and $m\prime = \lceil |M| - (c/2)/r \rceil$.

Scheme	APE	APE^{RI}	APE^{OW}	APE^{CA}	Bettle	APE^{+}
Bandwidth	$\|N\| + X + c$	$\|N\| + X + c$	$X + b$	$\|N\| + X + c/2$	$\|N\| + X + b$	$\|N\| + X + c$
Encryption	P	P	P	P	P	P
Decryption	P, P^{-1}	P^{-1}	P^{-1}	P^{-1}	P	P
Encryption cost	$n + a + m$	$n + a + m$	$n + a + m$	$n + a + m\prime$	$1 + a + m$	$1 + \max\{a, m\}$
Decryption cost	$n + a + m$	$n + a + m$	$n + a + m$	$n + a + m\prime$	$1 + a + m$	$1 + \max\{a, m\}$
Security	$c/2$	$c/2$	$\min\{r, c/2\}$	$c/2$	$\min\{b/2, c - \log r, r\}$	$\min\{r, c\}$
Nonce-misuse	Yes	Yes	Yes	Yes	No	Yes
Reference	[17]	[18]	[18]	[18]	[20]	This paper

processing the associated data and the message is faster on hardware devices.

(2) APE^{+} is inverse-free, i.e., its decryption circuit does not invoke the inverse of permutation. Moreover, it is a stream-cipher encryption mode.

(3) APE^{+} is built by the cascade method and has no backward feedback. Therefore, it can be fully pipeline implemented.

(4) To the best of our knowledge, APE^{+} is the first AE mode which supports beyond conventional security against blockwise adaptive adversaries in the lightweight devices.

(5) APE series and APE^{+} are designed and have proven security against nonce-misusing adversaries up to common prefix. Jovanovic et al. showed an attack on APE with a complexity of about $2^{c/2}$ in the nonce-respecting setting (here, "nonce-respecting" means that the nonce is never repeated in the encryption queries) according to the defect $M_i \oplus C_{i-1}$ [26]. If there exists k such that $M_k \oplus C_{k-1} = M_1 = 0$, the adversary breaks the privacy with a complexity of about $2^{c/2}$ in the nonce-respecting setting. In fact, this attack also works for APE series. This defect exists in APE^{RI}, APE^{OW}, and APE^{CA}, while it does not exist in APE^{+}. Therefore, APE^{+} is robust against this kind of attack.

Table 2 shows the comparison of permutation-based lightweight AE modes. From the perspective of hardware implementation costs, APE^{+} just needs the permutation circuit on hardware devices as its encryption and decryption algorithms only call the permutation P. Therefore, the area of the hardware device and the number of hardware footprints are minimized. From the perspective of the efficiency, the bandwidth of implementing is $|N|| + |A| + |M| + c$. Moreover, the computational costs of the encryption and decryption algorithms are $1 + \max\{\lceil |A|/r \rceil, \lceil |M|/r \rceil\}$ as we utilize the method of concurrent absorption to process the associated data and the message. Therefore, the computational complexity is obviously reduced. From the perspective of the security, APE^{+} enjoys at most about $\min\{r, c\}$-bit AE security, which is a great contribution of this paper. Fixing a permutation with recommended parameters $b = 256$, $r = 96$, and $c = 160$, APE series ensure at most about 80-bit security while APE^{+} enjoys at most about 96-bit security. Security levels of permutation-based AE modes using recommended parameters are shown in Table 1.

This paper just focuses on the single-key security of APE^{+}. Recently, the multikey or multiuser security and related-key security are also very hot research topics of lightweight ciphers. The implementation of APE^{+} on the hardware circuit and the security under the multikey or multiuser and related-key settings are our next important works.

6. Conclusions

Most of the devices widely used in smart home and Internet of Things are resource constrained. The privacy security and authenticity security of data from these devices are crucial in the process of data transmission. The lightweight AE modes designed by permutations have more advantages and attractions for the protection of data security due to its simple structure, convenient lookup table, and fast running speed. However, almost all of permutation-based lightweight AE modes enjoy conventional security. In this paper, we discuss the problem of whether can we design an efficient lightweight AE mode to achieve beyond conventional security bound for permutation-based lightweight ciphers. We propose a new permutation-based lightweight AE mode APE^{+} with beyond conventional security, derive its security proof, and discuss the properties of APE^{+}. APE^{+} has proven AE security up to about $2^{\min\{r,c\}}$ adversarial queries and it is robust, where r and c are, respectively, the rate and the capacity of the permutation. APE^{+} is an improved version of APE series and inherits most of the advantages of APE series. It is well suited for the protection of the data security in some special environments, such as an insufficiently large capacity of the permutation or the partial information leakage of permutation by side channel attacks.

Data Availability

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

Conflicts of Interest

The author declares that there are no conflicts of interest.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (Grant no. 61902195), Natural Science Fund for Colleges and Universities in Jiangsu Province (General Program, Grant no. 19KJB520045), and NUPTSF (Grant no. NY219131).

References

[1] C. Beierle, G. Leander, A. Moradi, and S. Rasoolzadeh, "CRAFT: lightweight tweakable block cipher with efficient protection against DFA attacks," *IACR Transactions on Symmetric Cryptology*, vol. 2019, no. 1, pp. 5–45, 2019.

[2] A. Bogdanov, L. R. Knudsen, and G. Leander, "Present: an ultra-lightweight block cipher," in *Lecture Notes in Computer Science 4727*, P. Paillier and I. Verbauwhede, Eds., pp. 450–466, Springer-Verlag, Berlin, Germany, 2007.

[3] W. Wu and L. Zhang, "Lblock: a lightweight block cipher," in *Lecture Notes in Computer Science*, J. López and G. Tsudik, Eds., Springer-Verlag, Berlin, Germany, pp. 327–344, 2011.

[4] A. R. Raza, K. Mahmood, M. F. Amjad, H. Abbas, and M. Afzal, "On the efficiency of software implementations of lightweight block ciphers from the perspective of programming languages," *Future Generation Computer Systems*, vol. 104, pp. 43–59, 2020.

[5] B. Rashidi, "High-throughput and flexible ASIC implementations of SIMON and SPECK lightweight block ciphers," *International Journal of Circuit Theory and Applications*, vol. 47, no. 8, pp. 1254–1268, 2019.

[6] P. Li, S. Zhou, B. Ren et al., "Efficient implementation of lightweight block ciphers on volta and pascal architecture," *Journal of Information Security and Applications*, vol. 47, pp. 235–245, 2019.

[7] Y. Wei, P. Xu, and Y. Rong, "Related-key impossible differential cryptanalysis on lightweight cipher TWINE," *Journal of Ambient Intelligence and Humanized Computing*, vol. 10, no. 2, pp. 509–517, 2019.

[8] D. Dinu, Y. L. Corre, D. Khovratovich, L. Perrin, J. Großschädl, and A. Biryukov, "Triathlon of lightweight block ciphers for the internet of things," *Journal of Cryptographic Engineering*, vol. 9, no. 3, pp. 283–302, 2019.

[9] T. Hiscock, O. Savry, and L. Goubin, "Lightweight instruction-level encryption for embedded processors using stream ciphers," *Microprocessors and Microsystems*, vol. 64, pp. 43–52, 2019.

[10] F. Farnoud, A. Abubakr, K. J. Peter, and G. Kris, "Faceoff between the CAESAR lightweight finalists: ACORN vs. Ascon," in *Proceedings of the International Conference on Field-Programmable Technology*, pp. 330–333, Naha, Japan, December 2018.

[11] Z. Bao, J. Guo, T. Iwata, and K. Minematsu, "ZOCB and ZOTR: tweakable blockcipher modes for authenticated encryption with full absorption," *IACR Transactions on Symmetric Cryptology*, vol. 2019, no. 2, pp. 1–54, 2019.

[12] Y. Naito and T. Sugawara, "Lightweight authenticated encryption mode of operation for tweakable block ciphers," *IACR Trans. Cryptogr. Hardw. Embed. Syst.*vol. 2020, no. 1, pp. 66–94, 2020.

[13] J. Mohsen, B. Nasour, and N. Zeinolabedin, "Lightweight implementation of SILC, CLOC, AES-JAMBU and COLM authenticated ciphers," *Microprocessors and Microsystems*, vol. 72, Article ID 102925, 2020.

[14] N. Yusuke, S. Yu, and S. Takeshi, "Lightweight authenticated encryption mode suitable for threshold implementation," in *Lecture Notes in Computer Science 12106*, C. Anne and I. Yuval, Eds., pp. 705–735, Springer-Verlag, Berlin, Germany, 2020.

[15] B. Andrey, M. Florian, R. Francesco, R. Vincent, and T. Elmar, "ALE: AES-based lightweight authenticated encryption," in *Lecture Notes in Computer Science 8424*, M. Shiho, Ed., pp. 447–466, Springer-Verlag, Berlin, Germany, 2013.

[16] W. E. Daniel, O. S. J. Markku, S. Peter, and M. S. Eric, "The hummingbird-2 lightweight authenticated encryption algorithm," in *Lecture Notes in Computer Science 7055*, M. Shiho, Ed., pp. 19–31, Springer-Verlag, Berlin, Germany, 2011.

[17] E. Andreeva, B. Bilgin, A. Bogdanov et al., "APE: authenticated permutation-based encryption for lightweight cryptography," in *Lecture Notes in Computer Science 8540*, C. Cid and C. Rechberger, Eds., pp. 168–186, Springer-Verlag, Berlin, Germany, 2014.

[18] Y. Sasaki and K. Yasuda, "Optimizing online permutation-based ae schemes for lightweight applications," *IEICE - Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. 102, no. 1, pp. 35–47, 2019.

[19] H. Kim and K. Kim, "Preliminary design of a novel lightweight authenticated encryption scheme based on the sponge function," in *Proceedings of the 10th Asia Joint Conference on Information Security*, pp. 110-111, Kaohsiung City, Taiwan, May 2015.

[20] A. Chakraborti, N. Datta, M. Nandi, and K. Yasuda, "Beetle family of lightweight and secure authenticated encryption ciphers," *IACR Transactions on Cryptographic Hardware and Embedded Systems*, vol. 2018, no. 2, pp. 218–241, 2018.

[21] P. Zhang and Q. Yuan, "Minimizing key materials: the even-mansour cipher revisited and its application to lightweight authenticated encryption," *Security and Communication Networks*, vol. 2020, Article ID 41801391, 2020.

[22] P. Rogaway and T. Shrimpton, "A provable-security treatment of the key-wrap problem," in *Lecture Notes in Computer Science 4004*, S. Vaudenay, Ed., Springer-Verlag, Berlin, Germany, pp. 373–390, 2006.

[23] C. Namprempre, P. Rogaway, and T. Shrimpton, "Reconsidering generic composition," in *Lecture Notes in Computer Science 8441*, P. Q. Nguyen and E. Oswald, Eds., Springer-Verlag, Berlin, Germany, pp. 257–274, 2014.

[24] R. Granger, P. Jovanovic, B. Mennink, and S. Neves, "Improved masking for tweakable blockciphers with applications to authenticated encryption," in *Lecture Notes in Computer Science 9665*, M. Fischlin and J. S. Coron, Eds., Springer-Verlag, Berlin, Germany, pp. 263–293, 2016.

[25] Y. Sasaki and K. Yasuda, "How to incorporate associated data in sponge-based authenticated encryption," in *Lecture Notes in Computer Science 9048*, K. Nyberg, Ed., Springer-Verlag, Berlin, Germany, pp. 353–370, 2015.

[26] P. Jovanovic, A. Luykx, B. Mennink, Y. Sasaki, and K. Yasuda, "Beyond conventional security in sponge-based authenticated encryption modes," *Journal of Cryptology*, vol. 32, no. 3, pp. 895–940, 2019.