ECTI Transactions on Computer and Information Technology
Journal homepage: https://ph01.tci-thaijo.org/index.php/ecticit/
Published by the ECTI Association, Thailand, ISSN: 2286-9131

A Privacy Preservation Model for the URL Query String of Local Links Based on Temporary Tables

Surapon Riyana[1], Kittikorn Sasujit[2] and Nigran Homdoung[3]

ABSTRACT

The URL (Uniform Resource Locator) is a unique identifier for locating a resource online. It generally includes a protocol (e.g., HTTP or HTTPS), a subdomain (e.g., WWW), a domain name, a path or webpage, and parameters in the form of URL query strings (HTTP.GET). The parameters proposed to identify the content of the destination webpage, e.g., the user profile, the user activities, or the details of the specified object. Thus, the destination webpage can pose privacy concerns when made publicly available. To rid these concerns in the local link of websites, a new privacy preservation model is proposed in this work. It is based on a temporary table. Aside from addressing privacy violation issues, a significant aim of the proposed models is to maintain the data utility as much as possible. Furthermore, the proposed model is evaluated by using extensive experiments. The experimental results show that the proposed model is an effective privacy preservation model that can be used to address privacy violation issues in URL query strings.

Article information:

Keywords: Temporary Table, HTTP.GET, URL Query String, Privacy Preservation, Privacy Violation, HTML Website, Web Server

Article history:
Received: July 18, 2024
Revised: September 19, 2024
Accepted: November 28, 2024
Published: December 28, 2024 (Online)

DOI: 10.37936/ecti-cit.2025191.257607

1. INTRODUCTION

A serious concern in data collection is privacy violation issues when the data collection is released. Countries have especially enacted laws to prevent personal data infringement, e.g., Thailand, Japan, Singapore, and the EU countries. For this reason, the identifier values of users are eliminated by an appropriate approach before it will be released. Generally, the identifier values of users in data collection are explicit identifier values and implicit identifier values (quasi-identifier values). The explicit identifier value is the value that the adversary can use to identify the data owner of the specified data directly, e.g., SSN, citizen ID, student code, and name. Thus, they must be removed before the data collection will be released. With the quasi-identifier values, the adversary can use their combined result to identify the owner of the specified data, e.g., age, sex, education, blood group, and zip code. For this reason, before the data collection is released, the unique quasi-identifier values are distorted by their less specific value to be indistinguishable. Data distortion techniques often used to eliminate the unique quasi-identifier values in data collection are Domain Generalization Hierarchy (DGH) [1][2][3], data suppression [4], data shuffling [5][6], data swapping [7], and data anatomization [8]. Aside from explicit identifier and quasi-identifier values, the data collection collects the user's sensitive values, e.g., salaries, diseases, and lawsuits.

To achieve privacy preservation constraints in data collection, the well-known privacy preservation models (e.g., k-Anonymity [9][10][11], l-Diversity [12], and t-Closeness [13]) and their extended versions (i.e., [14], [15], [16], [17], and [18]) are proposed. The principal aims of these privacy preservation models are balancing the data utility and the data privacy of data collection.

With k-Anonymity [9][10][11], data collection does not have any concerns about privacy violation issues when it does not include any explicit identifier values and any unique quasi-identifier values. That is, before the data collection is released, the explicit identifier values of users are removed. The unique quasi-identifier values are distorted by an appropriate data distortion technique to be at least k indistinguishable tuples.

With l-Diversity [12], data collection can satisfy privacy preservation constraints when it can guarantee every sensitive value has the confidence of data

[1,2,3] The authors are with the School of Renewable Energy, Maejo University, Sansai, Chiangmai, Thailand, 50290, Email: surapon_r@mju.ac.th, k.sasujit@yahoo.com and nigranhdg@gmail.com

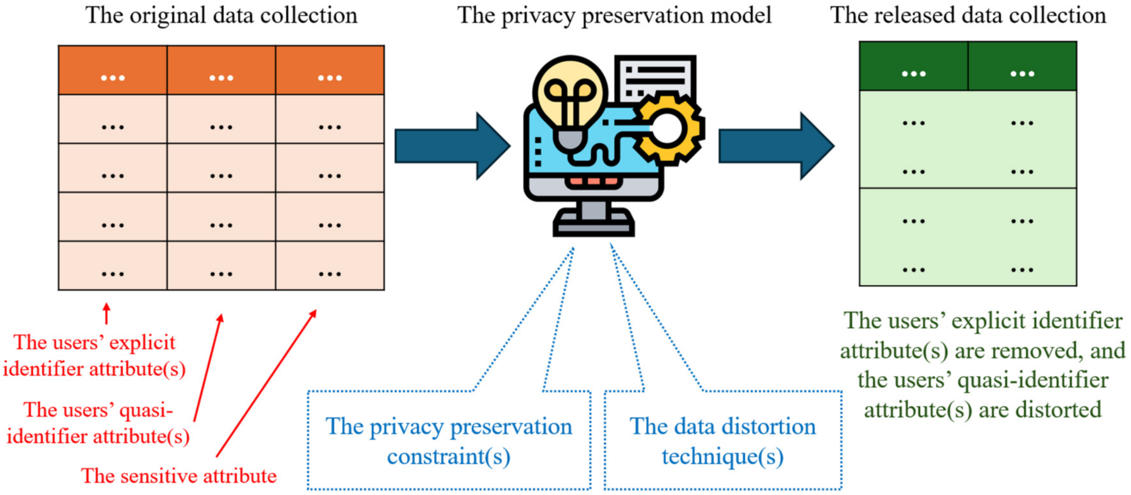

The original data collection
The privacy preservation model
The released data collection
The users' explicit identifier attribute(s)
The users' quasi-identifier attribute(s)
The sensitive attribute
The privacy preservation constraint(s)
The data distortion technique(s)
The users' explicit identifier attribute(s) are removed, and the users' quasi-identifier attribute(s) are distorted

Fig.1: *The infographic of preserving the privacy data in the data collection is based on k-Anonymity, l-Diversity, t-Closeness, and the expanded privacy preservation versions of them.*

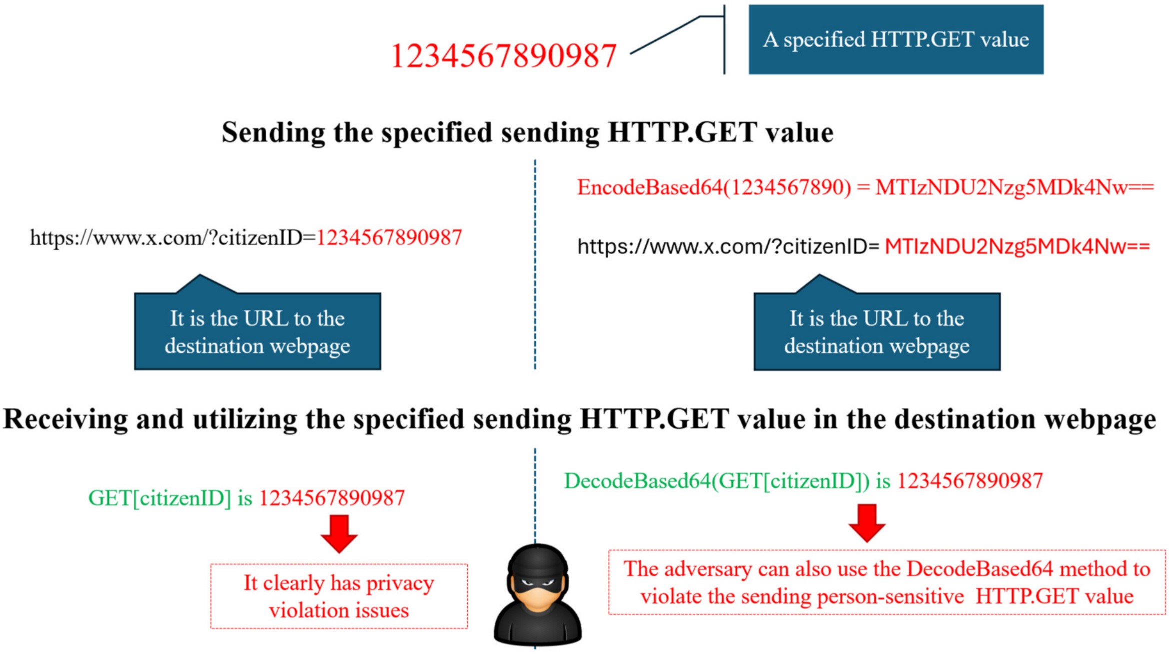

1234567890987
A specified HTTP.GET value
Sending the specified sending HTTP.GET value
https://www.x.com/?citizenID=1234567890987
It is the URL to the destination webpage
EncodeBased64(1234567890) = MTIzNDU2Nzg5MDk4Nw==
https://www.x.com/?citizenID= MTIzNDU2Nzg5MDk4Nw==
It is the URL to the destination webpage
Receiving and utilizing the specified sending HTTP.GET value in the destination webpage
GET[citizenID] is 1234567890987
It clearly has privacy violation issues
DecodeBased64(GET[citizenID]) is 1234567890987
The adversary can also use the DecodeBased64 method to violate the sending person-sensitive HTTP.GET value

Fig.2: *The infographic of sending, receiving, and utilizing the passed person-sensitive URL query string and the scenario of privacy violation issues in the based-64 encryption and decryption privacy preservation models.*

re-identifications to be at least $^{1}/_{l}$. Aside from the unique values, privacy violation issues in data collection can occur by considering the distance of sensitive values. To address these issues, t-closeness [13] is proposed. With this privacy preservation model, data collection does not have any concerns about privacy violation issues when the sensitive values can guarantee the distance between them and their related sensitive values to be at least t.

In brief, the data collection does not have any concerns about privacy violation issues with the above-mentioned privacy preservation models after all explicit identifier values of users are removed. Moreover, the unique quasi-identifier values are distorted by an appropriate data distortion technique to be indistinguishable. In addition, some privacy preservation models further consider the characteristics of sensitive values in their privacy preservation constraints, i.e., the confidence and distance of data re-identifiers. The infographic of preserving the privacy data of these privacy preservation models is shown in Fig.1.

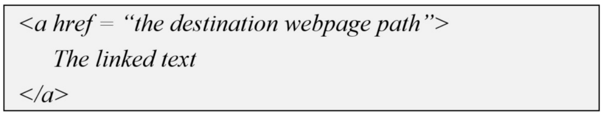
The above-mentioned privacy preservation models are used to address privacy violation issues in data collection that are released. For this reason, they could be ineffective and inefficient in addressing privacy violation issues with other data forms, e.g., URL query strings [22]. In addition, we will explain the privacy violation issues in URL query strings in the "Motivation" section. Moreover, we propose a new privacy preservation model for URL query strings in the "The proposed model" section.

2. CONTRIBUTIONS AND PAPER OUTLINES

The previous section (Section "Introduction") proposed presenting the well-known privacy preservation models for addressing privacy violation issues in data collection that are released. Then, the privacy violation issues in the URL query string are presented in the "Motivation" section. Then, a privacy preservation model for addressing the privacy violation issues in URL query strings will be explained in this section. For this reason, a new privacy preservation model for privacy violation issues in URL query strings is needed. It will be presented in the "The Proposed Model" section. In the "Experiment" section, the experimental results for evaluating the proposed model through extensive experiments are discussed. Finally, the conclusion of this work is discussed.

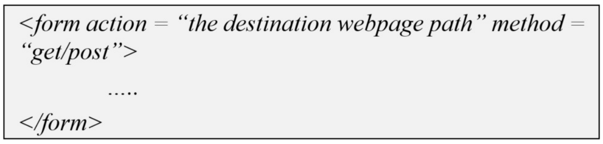
3. MOTIVATION

Websites are a dominant section of the internet. Generally, they are the collection of related data files that can be utilized and shared through web browsers. Examples of data files are often available on websites such as .htm, .html, .docx, .xlsx, .pptx, and .zip. Moreover, we found that some websites are proffered to manage the information for defining organization policies, improving market strategies, and determining customer services. To make it easier to study privacy violation issues on websites from URL query strings, only HTML hyperlinks and webpage links with JavaScript in .html and .htm are focused on this work. They are shown as follows.



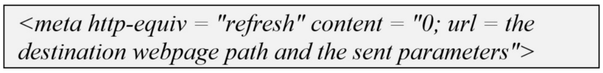
```
<a href = “the destination webpage path”>
    The linked text
</a>
```

Aside from HTML hyperlinks, an HTML form is often used to link from the current webpage to the destination webpage. It is shown as follows.



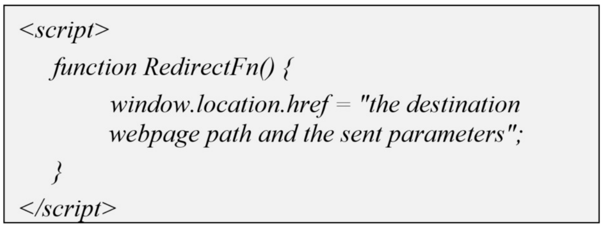
```
<form action = “the destination webpage path” method =
“get/post”>
            .....
</form>
```

Another HTML form, "meta refresh," can also be used to redirect from the current webpage to the destination webpage. Below is a form of using "meta refresh."



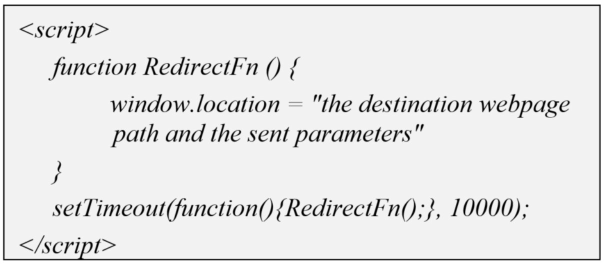
```
<meta http-equiv = "refresh" content = "0; url = the
destination webpage path and the sent parameters">
```

In addition to webpage links based on HTML, webpage links can be constructed from JavaScript. An example of JavaScript links is shown below.



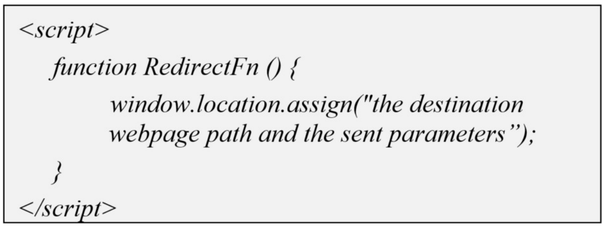
```
<script>
   function RedirectFn() {
         window.location.href = "the destination
         webpage path and the sent parameters";
   }
</script>
```

The second example of JavaScript's linked webpages is "*window.location*" as follows.



```
<script>
   function RedirectFn () {
         window.location = "the destination webpage
         path and the sent parameters"
   }
   setTimeout(function(){RedirectFn();}, 10000);
</script>
```

Another example of JavaScript's linked webpages is "*window.location.assign*". It is as follows.



```
<script>
   function RedirectFn () {
         window.location.assign("the destination
         webpage path and the sent parameters”);
   }
</script>
```

An example of a URL for sending a parameter from the current webpage to the destination webpage is "`https://www.x.com/?citizenID=1234567890987`". That is, `https://www.x.com/` is the path of the destination webpage. The citizenID is a URL query string (or an HTTP.GET parameter). 1234567890987 is the specified value that is presented by citizenID. Moreover, we suppose that 1234567890987 is Alice's citizen ID, which is available at `https://www.x.com`. Furthermore, we assume that "`https://www.x.com/?citizenID=1234567890987`" is the path to access Alice's profile webpage, which consists of Alice's work histories, Alice's income and expense histories, and Alice's health examination histories. In this situation, if the adversary can access this destination webpage, Alice's sensitive information can be violated.

To address privacy violation issues in URL query strings, in [19], [20], and [21], the authors propose the

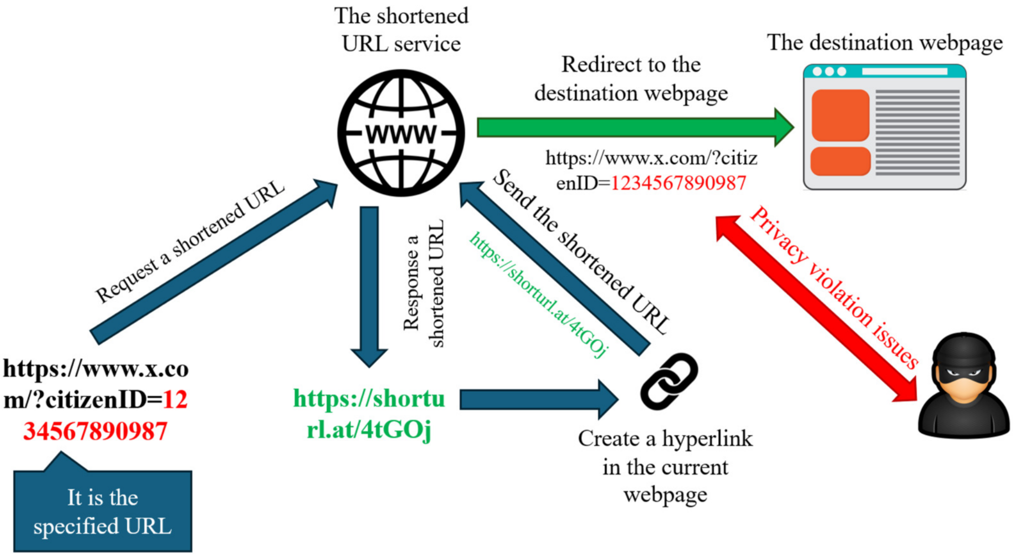

The shortened URL service
Redirect to the destination webpage
The destination webpage
https://www.x.com/?citizenID=1234567890987
Request a shortened URL
Response a shortened URL
Send the shortened URL
https://shorturl.at/4tGOj
Privacy violation issues
https://www.x.com/?citizenID=1234567890987
It is the specified URL
https://shorturl.at/4tGOj
Create a hyperlink in the current webpage

Fig.3: *The infographic of addressing privacy violation issues with the shortened URL and the vulnerability of shortened URL privacy preservation models.*

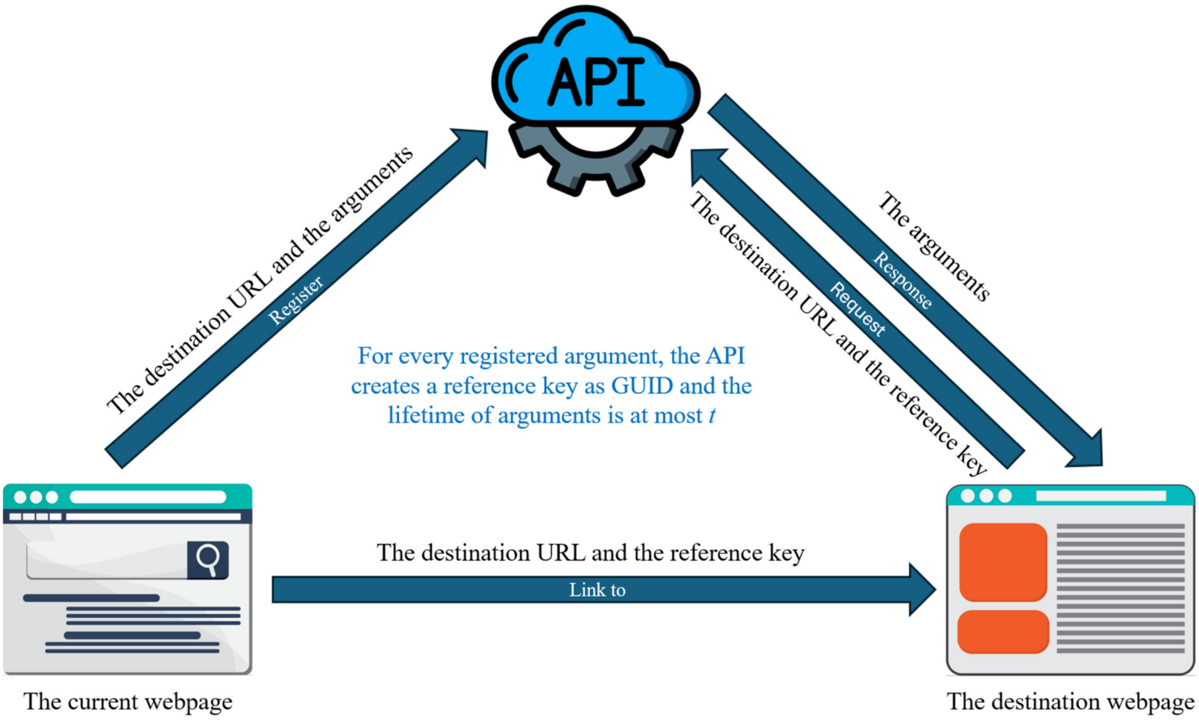

API
The destination URL and the arguments
Register
The arguments
Response
The destination URL and the reference key
Request
For every registered argument, the API creates a reference key as GUID and the lifetime of arguments is at most t
The destination URL and the reference key
Link to
The current webpage
The destination webpage

Fig.4: *The infographic of using API, Globally Unique IDentifier (GUID), and the lifetime of arguments for addressing privacy violation issues.*

based-64 encryption and decryption privacy preservation model. Before the person-sensitive URL query string (e.g., citizen ID, student code, and SSN) is utilized through the web browser and embedded in the webpage source code, it was encrypted as a data form of based-64 encryptions. In addition, the encrypted URL query string can be utilized in the destination webpage by using the based-64 data decryption method. However, in [23], the author demonstrates that the data is based on the based-64 data encryption and decryption; it still has privacy violation issues that must be addressed. That is, the adversary can also use the based-64 data decryption method to violate the passed and embedded encryption value that is available on the webpage. The infographic of sending, receiving, and utilizing the passed person-sensitive URL query string value and the scenario of privacy violation issues in the based-64 encryption and decryption privacy preservation models are shown in Fig.2.

To rid the vulnerabilities of based-64 encryption and decryption, in [23], the authors recommend using a shortened URL data version to preserve the privacy data available in web browsers and embedded in the webpage source code. Thus, the webpage seems to have no privacy violation concerns. However, to the best of our knowledge regarding privacy violation issues in URL query strings, we found that the originally person-sensitive version of the shortened URL is exhibited again when the destination webpage is available. In this situation, we can brief that the shortened URL can still have privacy violation concerns that must be addressed. The idea of shortened URL privacy preservation in the URL query string and its vulnerability is shown in Fig.3. To rid the vulnerabilities of shortened URL, in [24], a privacy preservation model is based on API, Globally Unique IDentifier (GUID), and the lifetime of URL query strings. That is, every URL query string is defined for data utilization between the current webpage and the destination webpage. It can be utilized through an API by considering the generated GUID as the data reference and the lifetime of the specified URL query strings. The infographic of preserving URL query strings is based on this privacy preservation model to be shown in Fig.4. For this reason, this URL query string privacy preservation model does not seem to have any privacy violation issues. However, we found that it still has vulnerabilities that the website administrator must consider. For example, there are vulnerabilities in setting up the lifetime of URL query strings. That is, when the website administrator sets up the lifetime of URL query strings that are too short, it could lead to data unusability issues when the speed of internet is low. However, when the lifetime of URL query strings is set too long, it can lead to privacy violation issues by leaking the reference key. Moreover, the privacy preservation processes of this model are separated into two sections, i.e., API and website. If one of these sections is down, all sections are also down. Furthermore, this privacy preservation model does not have any processes that are used to verify the destination webpage. Therefore, the passed URL query string could be utilized in the unwanted webpage.

To rid these vulnerabilities of the privacy preservation models that proposed to address privacy violation issues in URL query strings, a new privacy preservation model is proposed in this work. It will be presented in the “The proposed model” section.

4. THE MODEL FOR PRESERVING PRIVACY DATA IN URL QUERY STRINGS

In this section, we propose a new privacy preservation model that can be used to address the privacy violation issues in URL query strings. Before the proposed privacy is presented, we would like to define its basic definitions first.

4.1 The basic definitions

Definition 1 (The temporary table of URL query strings): *Let W be the website that is proposed to address privacy violation issues that are in the form of URL query strings. Let S be the web server that is proposed to provide W. Let K_W be the token key of W, i.e., it uses to identify W in S. Let P_C and P_D be both of arbitrary related webpages in W. That is, P_C is the current webpage. P_D is the destination webpage. Let $T = \{t_1, t_2, \ldots, t_n\}$ be the temporary table of URL query strings. Let $A = \{a_1, a_2, \ldots, a_i\}$ and $V = \{v_1, v_2, \ldots, v_i\}$ be the set of the argument names and the set of the argument values respectively, i.e., A and V are URL query strings that are proposed to utilize between P_C and P_D. Moreover, let $AV = \{(a_j, v_j) | a_j \in A, v_j \in V\}$, where $1 \leq j \leq i$, be the ordered pairs of A and V. Let $f_{REG}(K_W, W, AV, P_D)$:$\{K_W, W, AV, P_D\}$ $\rightarrow_{K_{AV}, TS_{AV}}$ $\{AV, P_D, K_W, W, K_{AV}, TS_{AV}\}$ be the function to register AV into T of W such that K_{AV} and TS_{AV} are the registered token key and the registered timestamp of AV respectively, i.e., every t_x is presented in the form of $t_x = \{AV, P_D, K_W, W, K_{AV}, TS_{AV}\}$.*

Definition 2 (The privacy violation issue of URL query strings in the current webpage): *Let $Code_{P_C}$ be the HTML syntax or other syntax forms that are used to construct P_C. Let U be the target user of the adversary in W. Let $IDEN_U$ be the U's the identifier value. Let $ENC(IDEN_U) : IDEN_U \rightarrow IDEN'_U$ be a function for encrypting $IDEN_U$ to be $IDEN'_U$. Let $IDEN'_U$ and P_D be built to be a link and available in $Code_{P_C}$. Let $DEC(IDEN'_U : IDEN'_U \rightarrow IDEN_U$ be a function for decrypting $IDEN'_U$ to be $IDEN_U$, i.e., $DEC(IDEN'_U) = IDEN_U$. The meaning of privacy violation issues in P_C is that the adversary can use an appropriate $DEC(IDEN'_U)$ for decrypting $IDEN'_U$ to be $IDEN'_U$. Furthermore, the adversary uses the decrypted value in conjunction with $Code_{P_C}$ to disclose or violate the sensitive data of U from P_D.*

Definition 3 (The privacy violation issue of URL query strings in the destination webpage): *Let t_{AV} be an argument row that is registered by P_C and passed to P_D such that t_{AV} is available in T. The meaning of privacy violation issues in P_D is that the adversary can utilize t_{AV} with a data utilization technique to disclose or violate the sensitive data of U from P_D.*

Definition 4 (The privacy preservation of URL query strings): *Let TS_C represent the current timestamp of S. Moreover, let TS_M be the maximized lifetime of every argument that can be utilized by P_D. Let $f_{GET}(K_W, K_{AV}) : \{K_W, K_{AV}\} \rightarrow AV$ be the function for getting AV such that AV are satisfied* *by* $(TS_C - TS_{AV}) \leq TS_M$.

4.2 The proposed model

This section is devoted to proposing a model for addressing privacy violation issues in URL query strings. The privacy preservation of the proposed privacy preservation model is that it does not allow P_C and P_D to utilize URL query strings directly, i.e., it does not allow P_C and P_D to utilize URL query strings through web browsers directly. For this reason, the processes of the proposed privacy preservation model are separated into two sections (Sections 4.2.1 and 4.2.2).

4.2.1 Registering URL query strings

To address privacy violation issues in URL query strings that are proposed to utilize between P_C and P_D, a temporary table is applied. That is, the URL query string AV is utilized between P_C and P_D, they are first registered into the temporary table T that is available in S by Algorithm 1. Finally, they are utilized by Algorithm 2 which is presented in Section 4.2.2.

```
Algorithm 1: $f_{REG}(AV, K_W, W, P_D)$

IF $K_W = GET\_K(W)$ THEN
    $K_{AV} := GENERATE\_GUID()$;
    $TS_{AV} := GET\_CURRENT\_DATETIME()$;
    $T := T \cup \{AV, P_D, K_W, W, K_{AV}, TS_{AV}\}$;
    RETURN $K_{AV}$
ELSE
    RETURN $Failure$;
END IF
```

```
Algorithm 2: $f_{GET}(K_W, K_{AV})$

IF $K_W = GET\_K(W)$ THEN
    $TS_M := c$;
    $TS_C := GET\_CURRENT\_DATETIME()$;
    $GET_{AV} := GET\_AV(K_W, K_{AV})$;
    IF $GET_{AV}[AV] \neq NULL$ THEN
        IF $(TS_C - GET_{AV}[TS_{AV}]) \leq TS_M$ THEN
            $UPDATE_{AV}[TS_{AV}] := \infty$;
            RETURN $GET_{AV}[AV]$;
        ELSE
            RETURN $Failure$;
        END IF
    ELSE
        RETURN $Failure$;
    END IF
ELSE
    RETURN $Failure$;
END IF
```

With Algorithm 1, the inputs are AV, K_{AV}, TS_{AV}, W, and P_D. That is, AV is the set of ordered pairs of the argument names and the argument values, K_W is the token key of W. P_D is the destination webpage. The output of this algorithm is K_{AV}.

To register the argument(s) from P_C into S, the specified input K_W is first investigated by comparing with $GET\_K(W)$. If K_W is equal to $GET\_K(W)$, the next processes for registering the specified argument(s) AV are enabled. But if K_W is not equal to $GET\_K(W)$, the algorithm returns Failure. In addition, $GET\_K(W)$ is the function that is proposed for getting the token key of W from S. Then, the token key of the registered argument(s) is generated such that it is a GUID. In addition, GUIDs are the acronym that stands for Globally Unique Identifier, they are also referred to as UUIDs or Universally Unique Identifiers. Technically they are 128-bit unique reference numbers used in computing which are highly unlikely to repeat when generated despite there being no central GUID authority to ensure uniqueness. Subsequently, the current timestamp of S is getting and it is set to be TS_{AV}. Then, all generated and defined values are stored into T. Finally, K_{AV} is returned.

An example of making P_D and AV in HTML hyperlink syntaxes or building a link from P_C to P_D such that AV is the argument(s) that is proposed to utilize between P_C to P_D. It is shown as follows.

```
<a href="P_D?K_W&K_AV">
    The linked text
</a>
```

4.2.2 Getting URL query strings

This section is devoted to describing the data utilization of AV in P_D such that it does not lead to the concern of privacy violation issues. We know that AV is sent from P_C to P_D indirectly. Thus, an appropriate method or function for utilizing AV in P_D must be defined, i.e., Algorithm 2.

With Algorithm 2, the inputs are K_W and K_{AV} such that they are the token key of W and AV respectively. The output of this algorithm is AV for P_D. For getting AV from T of S to use in P_D, $K_W$is first investigated by comparing with $GET\_K(W)$. If K_W is equal to $GET\_K(W)$ then the next processes for getting AV will be enabled, otherwise, the algorithm returns Failure (the value of K_W is not correct). When the next processes are enabled, the maximized lifetime TS_M of AV is set up to be c seconds (the value of c must set up to accord the runtime of the server S that is used to build P_D). Then, the information about the registered AV is according to the specified website token K_W and the specified token key argument(s) K_{AV} to be get by $GET_{AV}(K_W, K_{AV})$ and it is kept by GET_{AV}. That is, if K_W and K_{AV} are satisfied then the registered argument(s) AV are returned from T and kept by GET_{AV}, otherwise, GET_{AV} is $NULL$. Finally, the

Table 1: *The summary of the vulnerabilities for each experimental privacy preservation model.*

Privacy preservation model	Privacy preservation issue		The related server down issue	Invalid URL query string issue
	Current webpage	Destination webpage		
Based-64 encryption and decryption	✓	✓	–	–
Shortened URL	–	✓	✓	–
API and Globally Unique IDentifier (GUID)	–	–	✓	✓
The proposed model	–	–	–	–

algorithm checks that if $GET_{AV}[TS_{AV}]$ is not null and $(TS_C - GET_{AV}[TS_{AV}])$ is less than and equal to TS_M then the algorithm returns $GET_{AV}[AV]$ to P_D, otherwise, the algorithm returns Failure.

5. RESULTS AND DISCUSSIONS

In this section, the proposed privacy preservation model is evaluated by using data comparison with the privacy preservation models proposed that are presented in [19], [20], and [21]. All proposed evaluations are based on the experiments constructed from 10,000 random URL query strings sent from the current webpage to the destination webpage. With [19], all random URL query strings are further encrypted and decrypted by using the based-64 encryptions and decryption algorithm that is provided on the website "`https://www.base64decode.org`". While the privacy preservation model that is proposed in [22], the URL of the destination webpage is transformed to be a shortened URL by the shortened URL model that is provided by the website "`https://www.shorturl.at`]].

5.1 The privacy violation issue in the current webpage

All experiments show that the privacy preservation model is proposed in [19]. The concern of privacy violation issues still has to be addressed. That is, although every URL query string in the current webpage is decrypted to be a 64-based encryption and decryption data form, the adversary can use a 64-based decryption model to reveal the original data version of the URL query string. However, the proposed privacy preservation model and the URL query string privacy preservation models that are proposed in [22] and [23], they do not have any concerns about privacy violation issues in the current webpage because every URL query string is proposed to utilize between the current webpage and the destination webpage, it is not available in the current webpage, i.e., it is kept in the provided web server.

5.2 The privacy violation issue in the destination webpage

All experiments show that the proposed privacy preservation models of [19] and [22] still concern privacy violation issues that the website administrator must consider. With the privacy preservation model of [19], the adversary can reveal the decrypted URL query string by using a 64-based encryption model. With the privacy preservation model proposed in [22], we found that the original data version of the URL is shown in web browsers when the destination webpage is available. Also, the proposed privacy preservation model and the privacy preservation model proposed in [23] do not have any concerns about privacy violation issues because every URL query string is proposed to be utilized between the current webpage and the destination webpage, it is not available in the current webpage and the web browser.

5.3 The privacy violation issue from an unsuitably defined lifetime of URL query strings

With all experiments that are proposed to evaluate the experimental privacy preservation models about defining their lifetimes that are unsuitable, we found that the privacy preservation model is proposed in [23], it still has privacy violation issues and invalid URL query string issues that must be addressed. That is, the default lifetime of registered URL query strings is set to ∞ when it is active. Moreover, we found that the lifetime of registered URL query strings is set to 0 (zero) when it responds to the destination webpage successfully. Therefore, if an arbitrary URL query string is sent to the destination webpage and it is not utilized by the destination webpage successfully, it could be violated by the adversary when the token key is disclosed. However, the proposed privacy preservation model of this work cannot have any concern about these issues. That is because every registered URL query string is clearly configured about its lifetime of data utilization in the destination webpage. In addition, the proposed privacy preservation models of [19] and [22] do not have any concern about the unsuitably defined URL query strings in the destination webpage is not considered.

5.4 The data utility issue of URL query strings is in the destination webpage when the related web server is down

All experiments are proposed to evaluate the data utility issue of URL query strings that are available in the destination webpage when the related web server is down. Aside from 10,000 random URL query strings that are sent from the current webpage to the destination webpage, we suppose that the server that is providing the shortened URL model and the API of URL query strings, it is down. With this situation, we found that the privacy preservation model that is proposed in [22] and [23], they cannot provide the sent URL query strings to the destination webpage. However, the proposed privacy preservation model and the URL query string privacy preservation that is based on the 64-based encryption and decryption cannot have any effect on this situation because they are only proposed to run on a provided server.

In addition, the summary of the vulnerabilities for each experimental privacy preservation model is shown in Table 1.

6. CONCLUSION

In this work, a privacy preservation model for addressing privacy violation issues in URL query strings is proposed. It is based on temporary tables. The experimental results can indicate that the proposed privacy preservation model is more effective than the based-64 encryptions and decryption privacy preservation model, the shortened URL privacy preservation model, and the privacy preservation model that is based on API, Globally Unique IDentifier (GUID), and the lifetime of URL query strings (arguments). That is, the proposed URL query string privacy preservation model can address privacy violation issues and data utility issues in the current webpage, the destination webpage, the unsuitably defined lifetime of URL query strings, and the data utility issue of URL query strings is in the destination webpage when the related web server is down. In addition, although the proposed privacy preservation model can address privacy violation issues that are in URL query strings, an adversary will discover a new privacy violation approach that can be used to violate the privacy data that is available in URL query strings in the future. Thus, an appropriate privacy preservation model that can address the newly discovered privacy violation issues should also be proposed in the future.

AUTHOR CONTRIBUTIONS

Conceptualization, Surapon Riyana; methodology, Surapon Riyana; software, Surapon Riyana; validation, Surapon Riyana, Kittikorn Sasujit, and Nigran Homdoung; formal analysis, Surapon Riyana; investigation, Surapon Riyana; data curation, Surapon Riyana; writing—original draft preparation, Surapon Riyana; writing—review and editing, Kittikorn Sasujit and Nigran Homdoung; visualization, Surapon Riyana, Kittikorn Sasujit, and Nigran Homdoung; supervision, Surapon Riyana. All authors have read and agreed to the published version of the manuscript.

References

[1] Y. Huang, M. Milani and F. Chiang, "Privacy-aware data cleaning-as-a-service," *Information Systems*, vol. 94, p. 101608, 2020.

[2] M. Guerriero, D. A. Tamburri and E. Di Nitto, "Defining, Enforcing and Checking Privacy Policies In Data-Intensive Applications," *2018 IEEE/ACM 13th International Symposium on Software Engineering for Adaptive and Self-Managing Systems (SEAMS)*, Gothenburg, Sweden, pp. 172-182, 2018.

[3] G. A. Afzali and S. Mohammadi, "Privacy preserving big data mining: association rule hiding using fuzzy logic approach," *IET Information Security*, vol. 12, no. 1, pp. 15-24, 2018.

[4] S. Riyana and N. Riyana, "Achieving anonymization constraints in high-dimensional data publishing based on local and global data suppressions," *SN Computer Science*, vol. 3, no. 3, 2022.

[5] Ú. Erlingsson, V. Feldman, I. Mironov, A. Raghunathan, K. Talwar and A. Thakurta, "Amplification by shuffling: From local to central differential privacy via anonymity," in *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 2468-2479, 2019.

[6] V. Balcer, A. Cheu, M. Joseph and J. Mao, "Connecting robust shuffle privacy and pan-privacy," in *Proceedings of the 2021 ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pp. 2384-2403, 2021.

[7] D. Rankin, M. Black, R. Bond, J. Wallace, M. Mulvenna and G. Epelde, "Reliability of supervised machine learning using synthetic data in health care: Model to preserve privacy for data sharing," *JMIR medical informatics*, vol. 8, no. 7, e18910, 2020.

[8] L. Yao, Z. Chen, X. Wang, D. Liu and G. Wu, "Sensitive Label Privacy Preservation with Anatomization for Data Publishing," in *IEEE Transactions on Dependable and Secure Computing*, vol. 18, no. 2, pp. 904-917, 1 March-April 2021.

[9] D. Slijepčević, M. Henzl, L. D. Klausner, T. Dam, P. Kieseberg and M. Zeppelzauer, "*k*-Anonymity in practice: How generalisation and suppression affect machine learning classifiers," *Computers & Security*, vol. 111, p. 102488, 2021.

[10] R. Wei, H. Tian and H. Shen, "Improving *k*-anonymity based privacy preservation for collaborative filtering," *Computers & Electrical Engineering*, vol. 67, pp. 509-519, 2018.

[11] S. Zhang, X. Li, Z. Tan, T. Peng and G. Wang, "A caching and spatial K-anonymity driven privacy enhancement scheme in continuous location-based services," *Future Generation Computer Systems*, vol. 94, pp. 40-50, 2019.

[12] A. Machanavajjhala, J. Gehrke, D. Kifer and M. Venkitasubramaniam, "L-diversity: privacy beyond k-anonymity," *22nd International Conference on Data Engineering (ICDE'06)*, Atlanta, GA, USA, pp. 24-24, 2006.

[13] N. Li, T. Li and S. Venkatasubramanian, "t-Closeness: Privacy Beyond k-Anonymity and l-Diversity," *2007 IEEE 23rd International Conference on Data Engineering*, Istanbul, Turkey, pp. 106-115, 2007.

[14] L. Yao, X. Wang, X. Wang, H. Hu and G. Wu, "Publishing Sensitive Trajectory Data Under Enhanced l-Diversity Model," *2019 20th IEEE International Conference on Mobile Data Management (MDM)*, Hong Kong, China, pp. 160-169, 2019.

[15] M. Xue, P. Kalnis and H. K. Pung, "Location diversity: Enhanced privacy protection in location based services," *Location and Context Awareness*, vol. 5561, pp. 70-87, 2009.

[16] H. Zhu, H. -B. Liang, L. Zhao, D. -Y. Peng and L. Xiong, "τ -Safe (l, k)-Diversity Privacy Model for Sequential Publication With High Utility," in *IEEE Access*, vol. 7, pp. 687-701, 2019.

[17] B. Meden *et al.*, "Privacy–Enhancing Face Biometrics: A Comprehensive Survey," in *IEEE Transactions on Information Forensics and Security*, vol. 16, pp. 4147-4183, 2021.

[18] L. Yao, X. Wang, X. Wang, H. Hu and G. Wu, "Publishing Sensitive Trajectory Data Under Enhanced l-Diversity Model," *2019 20th IEEE International Conference on Mobile Data Management (MDM)*, Hong Kong, China, pp. 160-169, 2019.

[19] A. R. Pathak, S. Deshpande and M. Panchal, "A secure framework for file encryption using base64 encoding," *Computing and Network Sustainability*, vol. 75, pp. 359-366, 2019.

[20] P. Kalia, D. Bansal and S. Sofat, "Privacy preservation in cloud computing using randomized encoding," *Wireless Personal Communications*, vol. 120, pp. 2847-2859, 2021.

[21] J. Kaur, A. Agrawal and R. A. Khan, "Encryfuscation: A model for preserving data and location privacy in fog based IoT scenario," *Journal of King Saud University-Computer and Information Sciences*, vol. 34, no. 9, pp. 6808-6817, 2022.

[22] A. G. West and A. J. Aviv, "Measuring Privacy Disclosures in URL Query Strings," in *IEEE Internet Computing*, vol. 18, no. 6, pp. 52-59, Nov.-Dec. 2014.

[23] S. Orhpomma, V. Mekthanavanh, S. Soukhavong, N. Homdoung, K. Sasujit and S. Riyana, "Model for Preserving Privacy Data in URL Query Strings," *2024 10th International Conference on Engineering, Applied Sciences, and Technology (ICEAST)*, Luang Prabang, Laos, pp. 57-60, 2024.

[24] A. A. Khan, M. Abou El-Ela and M. A. Al-Turaigi, "Current-mode Precision Rectification," *International Journal of Electronics*, vol. 79, No.2, pp.853-859, 1995.

[25] D. Endo, T. Hiyane, K. Atsuta and S. Kondo, "Fractal Image Compression by the Classification in the Wavelet Transform Domain," *Proceeding of 5th IEEE International Conference on Image Processing (ICIP 98)*, pp.190-193, 1998.

[26] C. Phongcharoenpanich, "Slot Array Antenna," D. Eng. dissertation, King Mongkut's Institute of Technology Ladkrabang, Bangkok, Thailand, pp. 138-145, 2001.

Surapon Riyana received a B.S. degree in computer science from Payap University (PYU), Chiangmai, Thailand, in 2005. Moreover, He further received a M.S. degree and a Ph.D. degree in computer engineering from Chiangmai University (CMU), Thailand, in 2012 and 2019 respectively. Currently, he is a lecturer in Smart Farming and Agricultural innovation Engineering (Continuing Program), School of Renewable Energy, Maejo University (MJU), Thailand. His research interests include data mining, databases, data models, privacy preservation, data security, databases, and the internet of things.

Kittikorn Sasujit received a B.Eng (Environmental Engineering) in 2004 from Rajamangala University of Technology Lanna, Thailand, and an M. Eng and Ph.D. (Energy Engineering) in 2008 and 2020, respectively, from Chiang Mai University, Thailand. His studies will include biomass technology, wind energy technology, NTP applications for biomass tar removal, and renewable energy.

Nigran Homdoung received a B.S. degree in mechanical engineering from King Mongkut's University of Technology Thonburi (KMUTT), Thailand, in 2001. He received a M.Eng. in Energy Engineering from Chiang Mai University (CMU), Thailand, in 2007. Moreover, he received a D.Eng. In Mechanical Engineering from Chiang Mai University (CMU), Thailand, in 2015. His research interests include biomass technology (gasification and pyrolysis process) and application Internal combustion Engine to biofuels. machine learning, data science, and artificial intelligence.