electronics

MDPI

Article

A Time-Based Dynamic Operation Model for Webpage Steganography Methods

Simun Yuk and Youngho Cho *

Department of Defense Science (Computer Engineering), Graduate School of Defense Management, Korean National Defense University, Nonsan 33021, Korea; 6simun@kndu.ac.kr

* Correspondence: youngho@kndu.ac.kr

Received: 11 November 2020; Accepted: 8 December 2020; Published: 10 December 2020

Abstract: The webpage steganography technique has been used for a covert communication method for various purposes in which a sender embeds a secret message into a plain webpage file like an HTML file by using various steganography methods. With human eyes, it is very difficult to distinguish between the original webpage (cover webpage) and the modified webpage with the secret data (stego webpage) because both are displayed alike in a web browser. In this approach, when two communicating entities want to share a secret message, a sender uploads a stego webpage to a web server or modifies an existing webpage in the web server by using a webpage steganography method, and then a receiver accesses the stego webpage to download and extract the embedded secret data from it. Meanwhile, according to our extensive survey, we observed that most webpage steganography methods focused on proposing or improving steganography algorithms but did not well address how to operate a stego webpage as time passes. However, if a stego webpage is used in a static way such that the stego webpage does not change and is constantly exposed to web clients until the sender removes it, such a static operation approach will limit or badly affect the hiding capacity and undetectability of a webpage steganography method. By this motivation, in this paper, we proposed a time-based dynamic operation model (TDOM) that improves the performance of existing webpage steganography methods in terms of hiding capacity and undetectability by dynamically replacing the stego webpage with other stego webpages or the original webpage. In addition, we designed two time-based dynamic operation algorithms (TDOA-C and TDOA-U), which improve the hiding capacity of existing methods and TDOA-U for improving the undetectability of existing methods, respectively. To validate our model and show the performance of our proposed algorithms, we conducted extensive comparative experiments and numerical analysis by implementing two webpage steganography methods with our TDOM (CCL with TDOA-C and COA with TDOA-C) and tested them in the web environment. According to our experiments and analysis, our proposed algorithms could significantly improve the hiding capacity and undetectability of two existing webpage steganography methods.

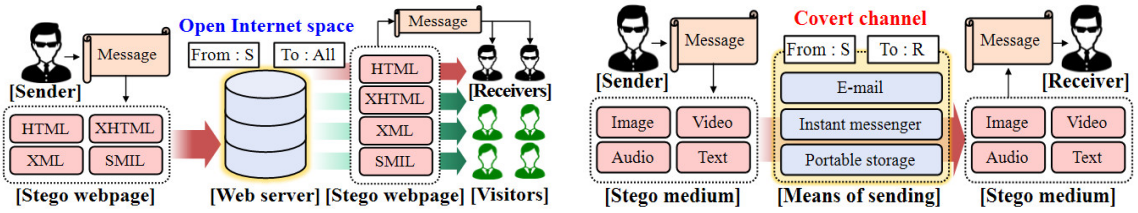
Keywords: webpage steganography; text steganography; HTML steganography; data hiding; covert communication; time-based dynamic operation model

1. Introduction

Steganography techniques have been used for covert communication between a sender and a receiver who want to hide their secret communication even in the presence of unauthorized entities [1,2]. In this approach, the sender creates a stego medium by embedding a secret message into a cover medium (e.g., image, audio, video, text, webpage, etc.) by using various steganography methods depending on the characteristics of cover mediums. Due to its undetectability, the steganography

techniques have been used for terrors, crimes, and espionage, and it is not difficult to find many related cases and reports on news media [3–5].

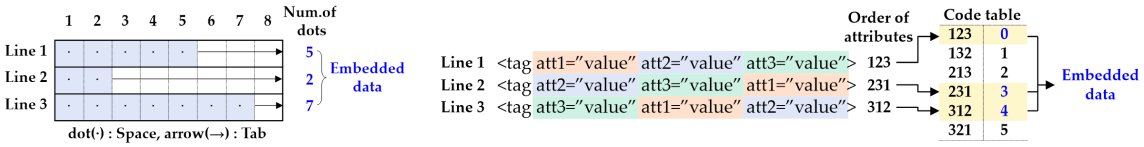
The webpage steganography technique is one of the representative steganography techniques that use a webpage file (e.g., HTML, XHTML, XML, SMIL, etc.) as a cover medium [6]. The created webpage with the secret message is called *stego webpage*. The webpage steganography exploits the characteristic of a webpage file like HTML such that it is less sensitive to the change of its syntax in displayed view in the web browser compared with other web programming languages such as Java or Python. Thus, a sender can hide a secret message easily into a webpage file by manipulating its source codes such that changes to the webpage file do not affect the view of the webpage in the web browser to avoid being detected by a monitoring entity (warden). In addition, the webpage steganography technique can deliver a secret message to more receivers than other types of steganography techniques. For example, while a sender delivers a secret message to one or a group of receivers in other types of steganography techniques (Figure 1b), a secret message can be delivered to many receivers (especially, to anonymous receivers) efficiently because a stego webpage is deployed in a web server, which is accessed by many web users (Figure 1a).



(**a**) Webpage steganography technique (**b**) Other types of steganography techniques

Figure 1. Webpage steganography technique vs. other types of steganography techniques.

As a representative webpage steganography-based cyberattack case, Kaspersky released a report on Platinum in 2019, which is one of the famous hacking groups [7]. According to the report, the Platinum had leaked critical information from governmental and military domains of southeast Asian countries by using two webpage steganography methods to hide their behaviors. For example, one of the methods is to add some special characters such as **whitespace** and **tab** as much as they need to embed secret data into HTML files because those special characters are not visualized when they are parsed by the web browser (Figure 2a). The other method is to change the order of attributes in HTML files. This method uses the fact that the order of attributes in the same tag does not affect the display generated by the web browser (Figure 2b).



(**a**) Adding special characters (**b**) Changing the order of attributes

Figure 2. Webpage steganography methods used by Platinum [7].

According to our extensive survey [6,8–31], most existing studies did not consider how the stego webpage should be operated as time passes. However, how to operate a stego webpage should be well addressed. Thus, if a stego webpage is operated simply in a static manner such that such a static operation feature will make a webpage steganography method very inefficient and inferior in terms of hiding capacity and undetectability because the stego webpage does not change as time passes. We will discuss in detail in Section 2.3.

On the other hand, if we operate a stego webpage in a dynamic manner such that the stego webpage is replaced with other stego webpages containing different secret messages or with its original webpage (cover webpage), we believe that the hiding capacity and undetectability of the

webpage steganography method used for generating the stego webpage can be significantly improved. By this motivation, in this study, we propose and design our time-based dynamic operation model for webpage steganography methods.

Our contributions in this study can be summarized as the following:

- We proposed a time-based dynamic operation model (TDOM) that improves the performance of existing webpage steganography methods in terms of hiding capacity and undetectability.
- We designed two time-based dynamic operation algorithms TDOA-C and TDOA-U, which improve the hiding capacity of existing methods and TDOA-U for improving the undetectability of existing methods, respectively.
- We conducted comparative experiments and numerical analysis to validate our model and show the performance of our proposed algorithms. For this, we implemented two webpage steganography methods with our TDOM (CCL with TDOA-C and COA with TDOA-C) and tested them in the web environment; CCL stands for Changing the Cases of Letters in tags and attributes and COA stands for Changing the Order of Attributes.
- In addition to the above contributions, we hope that this study can provide useful information about webpage steganography techniques and their dynamic operations, which can be used in cyberattacks or cybercrimes, raise an alarm to security engineers and researchers, and, thus, attract them to research defense mechanisms and techniques against them. Meanwhile, we note that our study can be used in positive use cases because the steganography-based covert communication can be used to avoid unauthorized and illegal eavesdroppers.

The rest of this paper is organized as follows. In Section 2, we overview backgrounds and related studies. In Section 3, we propose our time-based dynamic operation model (TDOM) for existing webpage steganography methods and design two time-based dynamic operation algorithms (TDOA-C and TDOA-U) based on two existing methods (CCL and COA). In Section 4, we conduct comparative experiments and numerical analysis. Finally, we conclude in Section 5.

2. Background and Related Works

2.1. Webpage Steganography-Based Covert Communication Model

In steganography-based covert communication [1,2,8], the sender hides a secret message into a cover medium (e.g., image, video, audio, and text file) by using a steganography technique depending on the characteristics of the cover medium. The modified cover medium with a secret message is called the stego medium.

In a webpage steganography-based covert communication model, a webpage is used for a cover medium and various applications and platforms on the Internet are used for transmission channels. In this model, the cover medium and stego medium are called *cover webpage* (the original webpage) and *stego webpage* (the modified webpage), respectively. As shown in Figure 3, a sender hides a secret message into a cover webpage by using a webpage steganography method and then transmits it to the receiver over the Internet. However, a monitor (a passive or active warden) [1,2,8] may watch the transmission channel to detect the existence of covert communication between the communicating parties (sender and receiver). Therefore, a sender must choose a webpage steganography method that can avoid the monitor's detection, and, for this reason, many webpage steganography methods have been devised and proposed in this research area [6,8–31]. We note that a webpage is created by using various web programming languages (e.g., HTML, XHTML, XML, etc.), and, since HTML is the most popular and familiar, we will use an HTML file for a representative webpage example for the rest of our paper without the loss of generality to validate our proposed idea.

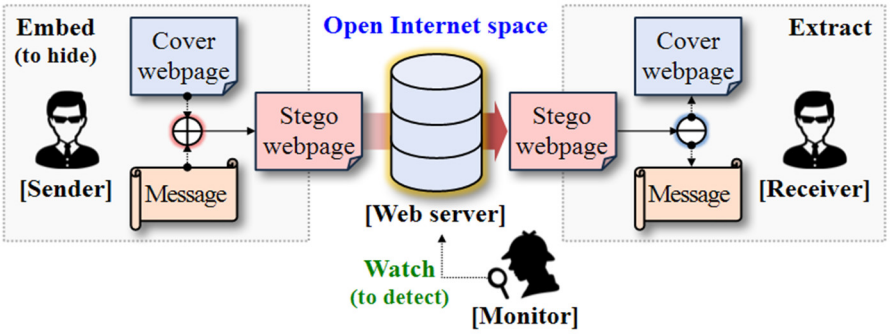


Figure 3. Webpage steganography-based covert communication model.

As shown in Figure 4, a general HTML webpage file consists of several elements, and each element starts with <*tagname*> and end with </*tagname*>. In a real HTML file, <*tagname*> can be replaced with <body>, <table>, <form>, and so on. Each tag can have multiple attributes that have a certain value to provide additional information to the element. Compared to other programming languages such as Java or Python, HTML is insensitive to changing the positions of attributes, inserting invisible characters, or capitalizing (or down-casing) the names of attributes [9]. By using such a characteristic of HTML, a sender can embed a secret message into a webpage file by manipulating the HTML source code while maintaining not only the screen view of the modified webpage displayed in the web browser, which is the same but also the syntax (grammar) of the modified HTML syntax is not captured by a simple HTML markup checker or validator. We will explain more details on embedding methods in Section 2.3.

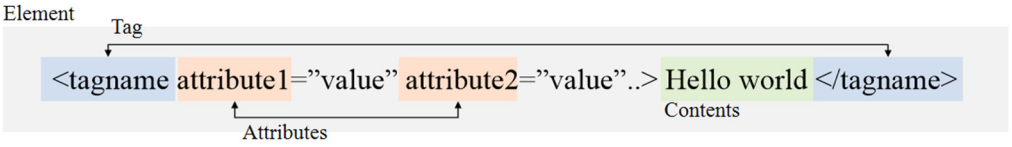


Figure 4. The basic structure of the Hyper Text Markup Language (HTML) webpage element (with a tag, attributes, and values).

The performance of a webpage steganography method can be evaluated in terms of the following three characteristics (hiding capacity, undetectability, and robustness) [1,9,32].

- **Hiding capacity**: The amount of information that can be embedded in a cover webpage, and this depends on the embedding algorithm of the webpage steganography method and a cover webpage.
- **Undetectability**: This refers to a degree that the embedded hidden message is not detected by a monitor (passive warden).
- **Robustness:** This refers to a degree that the embedded hidden message is not destroyed by a monitor (active warden).

2.2. Three Representative Webpage Steganography Approaches and Previous Studies

2.2.1. Adding Invisible Tags or Characters

In a webpage like HTML, special characters (e.g., whitespace) or tags without contents (e.g., <a> or) are not displayed in a web browser even if they exist in the webpage's source codes. In addition, comments (e.g., texts between <!-- and --> in HTML) are not displayed since they are not parsed by the web browser [10,11]. By using them, this approach hides a secret message into a webpage. While this approach can embed a huge amount of data by simply adding those tags and characters into an HTML file, it can be detected easily due to the suspiciously increased file size. Moreover, previous representative studies are below.

- Reference [12] proposed a method of inserting extra spaces and tabs into an HTML file.
- References [13,14] proposed a method of using various characters such as   or &#32 that are displayed as blank spaces in the web browser. In addition to this method, References [15–17] applied cryptography techniques for improving undetectability, Reference [15] used AES(Advanced Encryption Standard), Reference [16] used Caesar cipher, and Reference [17] used RSA(Rivest Shamir Adleman), respectively.
- Reference [18] suggested a method that inserts elements only containing tags without contents (attributes and values). For example, texts between with are bolded, but, if there are no texts between tags, there will be no change in the web browser even though those elements are added.
- There are a couple of webpage steganography tools such as Wbstego [33], Invisible Secret [34], and Snow [35].

2.2.2. Changing the Case of Letters in Tags and Attributes (CCL Approach)

This approach hides a secret message by changing cases of letters in a webpage source code like a case-insensitive HTML file. For example, <HeaD> hides 1001 including 1 for an upper case and 0 for a lower case. Unlike the above approach, this approach does not increase the webpage's file size and, thus, can avoid a simple file checker that examines the file size. Meanwhile, if source codes of a cover webpage and a stego webpage are known to a monitor, this approach can be easily detected. Representative previous studies are below.

- Reference [6] proposed a method that hides a secret message by encoding 0 or 1 according to the case of each character in HTML tags or attributes.
- References [19,20] and [21] proposed a similar method for watermarking.
- Reference [22] proposed a method to create a tag dictionary by gathering all the case-insensitive elements in the HTML specification.

2.2.3. Changing the Order of Attributes (COA Approach)

This approach uses the order of attributes in a tag because, even if the order of attributes is changed, the screen view on the web browser does not change. In addition, this approach does not increase the webpage's file size. Representative previous studies are below.

- Reference [23] proposed methods in which the order of a pair of attributes (two attributes) are used for data hiding. For example, if attribute A is before attribute B, 1 is hidden. Otherwise, 0 is hidden. References [24,25] proposed methods using the permutation of n attributes.
- References [26–29] applied some techniques for improving hiding capacity. Reference [26] used CCL, Reference [27] used Huffman coding, Reference [28] used various types of quotation marks, and Reference [29] proposed a combined approach of References [26] and [28].
- Reference [30] tried to improve the security of this approach by applying cryptography techniques.
- Reference [31] proposed a method that hides 0 or 1 by using whether the attribute belongs to a specific tag. Thus, 1 is hidden when two attributes belong to the same tag. Otherwise, 0 is hidden.
- Reference [36] is a webpage steganography tool, but some HTML tags or attributes newly added after 2006 are not included in their tool.

2.3. Limitations of Previous Studies

In general, all webpage steganography methods have the following covert communication models (see Figure 5).

First, the sender must upload a stego webpage to the web server. Next, when the receiver accesses and requests the stego webpage, which looks like its original webpage (cover webpage), the web server transmits the stego webpage file to the receiver. Lastly, the received webpage file is displayed by the receiver's web browser.

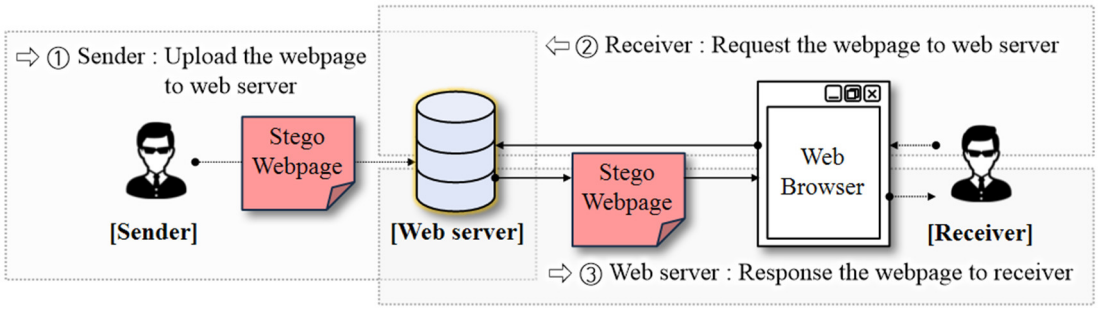


Figure 5. Covert communication model using webpages' steganography techniques.

According to our extensive survey [6,8–31], most existing studies did not consider how the stego webpage should be operated as time passes. However, how to operate a stego webpage should be well addressed. Thus, if a stego webpage is operated simply in a static manner such that it does not change as time passes, such a static operation feature will make a webpage steganography method very inefficient and inferior in terms of hiding capacity and undetectability because of the following reasons.

- *Hiding Capacity (or Capacity)*: Assuming that, given a webpage, the steganography method and the maximum hiding capacity of a stego webpage is m bits, the total amount of a secret message that the sender can deliver to the receiver is limited to the constant value m bits because the stego webpage does not change and, thus, the receiver will always obtain the same m-bit secret message regardless of request times to the stego webpage.
- *Undetectability*: Assuming that there is a perfect monitoring and detection tool, the static stego webpage will be captured by the tool at the time when the tool accesses the stego webpage since the stego webpage does not change and, thus, is constantly exposed to any web client.

Meanwhile, when a stego webpage is operated dynamically such that it is replaced with other stego webpages containing different secret messages or with its original webpage, we believe that the hiding capacity and undetectability of the webpage steganography method used for generating the stego webpage can be significantly improved. By this motivation, in this study, we will propose and design our time-based dynamic operation model for webpage steganography methods.

3. Proposed Model and Algorithms

3.1. Our Approach: Time-Based Dynamic Operation Model (TDOM)

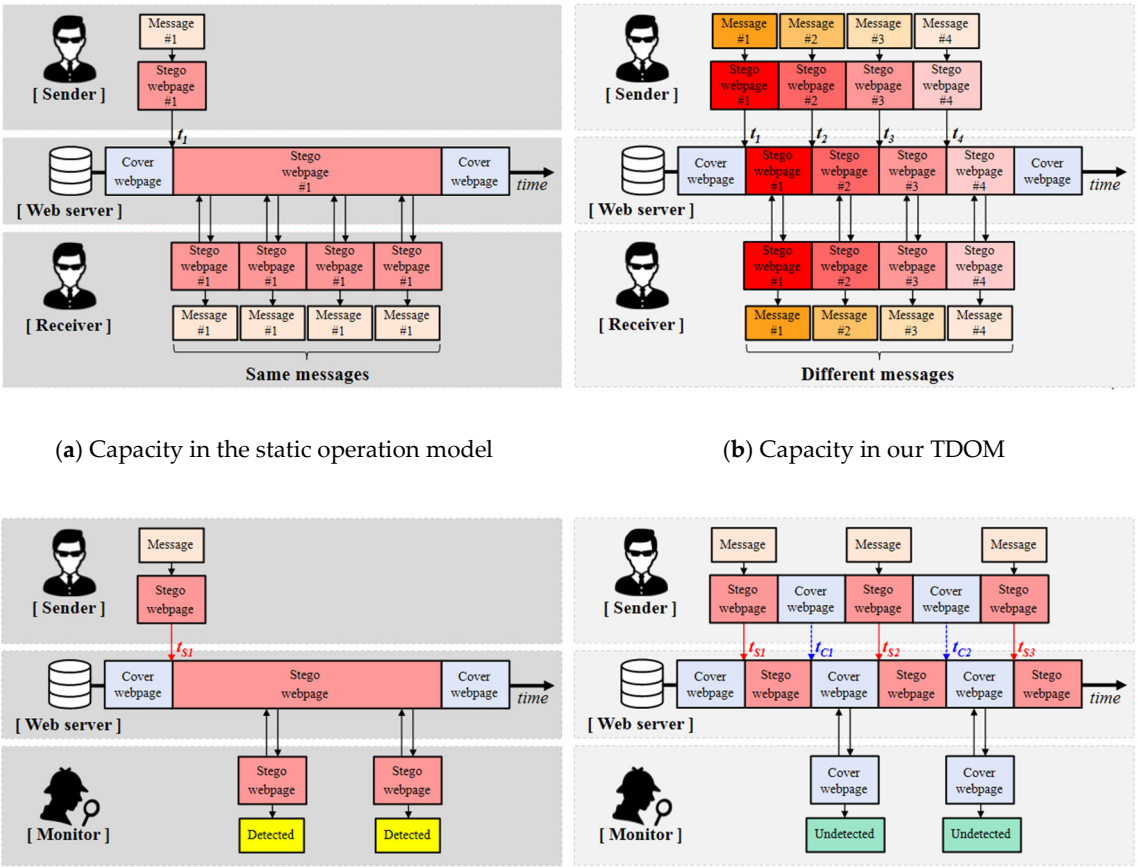
To improve the hiding capacity and undetectability of existing webpage steganography methods, we propose a simple but effective Time-based Dynamic Operation Model (TDOM) for them that controls when a stego webpage is replaced with a new stego webpage with a different secret message for higher capacity or when a stego webpage is exposed to users for higher undetectability.

Figure 6 shows how our TDOM (dynamic operation model) can improve the hiding capacity and undetectability of an existing method, which is operated in a static manner (static operation model). We note that, according to our survey, there is no clear static operation model for existing steganography methods but, for better understanding, we consider the latter case the static operation model in this study. Thus, we here compare how an existing method can work differently when it is operated in the static operation model and our dynamic operation model.

- *For higher hiding capacity (or larger amount of secret message delivery)*: Given the same time slot ($t_2 - t_1$), as shown in Figure 6a, a webpage steganography method delivers only one message (#1) in the static operation model. Meanwhile, as shown in Figure 6b, it can deliver four messages (#1–#4) in our TDOM since four different stego webpages are uploaded in turn in the time slot.
- *For higher undetectability*: Given the same time slot ($t_2 - t_1$), as shown in Figure 6c, a stego webpage keeps exposed to users, and, thus, if a powerful monitor (detection system) visits the

stego webpage, the monitor can detect the existence of the stego webpage. Meanwhile, as shown in Figure 6d, when the stego webpage is deactivated and replaced with the original webpage two times, the chance that the monitor detects the stego webpage will definitely decrease accordingly.

Our model is designed as a generic architecture to existing webpage steganography methods. Thus, it is not necessary to modify existing methods and our model can be easily combined with them. Although TDOM can be designed and implemented in various ways considering the degree of hiding capacity and undetectability, which are in a trade-off relationship, we show two time-based dynamic operation algorithms: (1) TDOA-C mainly focusing on hiding capacity and TDOA-U mainly focusing on undetectability.



(**a**) Capacity in the static operation model

(**b**) Capacity in our TDOM

(**c**) Undetectability in the static operation model

(**d**) Undetectability in our TDOM

Figure 6. Comparison of the static operation model and our Time-based Dynamic Operation Model (TDOM).

3.2. Time-based Dynamic Operation Algorithm for Hiding Capacity (TDOA-C)

As we briefly explained in Section 3.1, TDOA-C mainly focuses on improving the hiding capacity of an existing webpage steganography method without modifying it. Thus, as shown in Figure 7, when our TDOA-C is combined with an existing webpage steganography method, given a specific time period, the hiding capacity of the method can be significantly improved as the following steps. Let us assume that a sender S wants to send a secret message M_F to a receiver R by using a certain webpage steganography method WSM. In addition, the sender uses a cover webpage CW (original webpage) and the maximum hiding capacity $HC[CW] = c$ bits. Algorithm 1 describes TDOA-C for the sender and receiver.

- **Step 1**. S partitions M_F into n smaller messages M_1, M_2, …, M_n.
- **Step 2**. S creates n stego webpages SW_1, SW_2, …, SW_n by using WSM, CW, and M_1, M_2, …, M_n.
- **Step 3**. S uploads SW_1, SW_2, …, SW_n to the web server, in turn, at a certain time $t_i \in T = \{t_1, t_2, \dots, t_n\}$, which is agreed with R. For this step, the sender and receiver exchange such information by using another covert channel or in person.
- **Step 4**. R receives each stego webpage, in turn, by accessing the URL where the cover webpage is located at the time period agreed with S.
- **Step 5**. R extracts the partitioned secret messages M_1, M_2, …, M_n from received stego webpages.
- **Step 6**. R restores the complete secret message M_F by combining M_1, M_2, …, M_n.

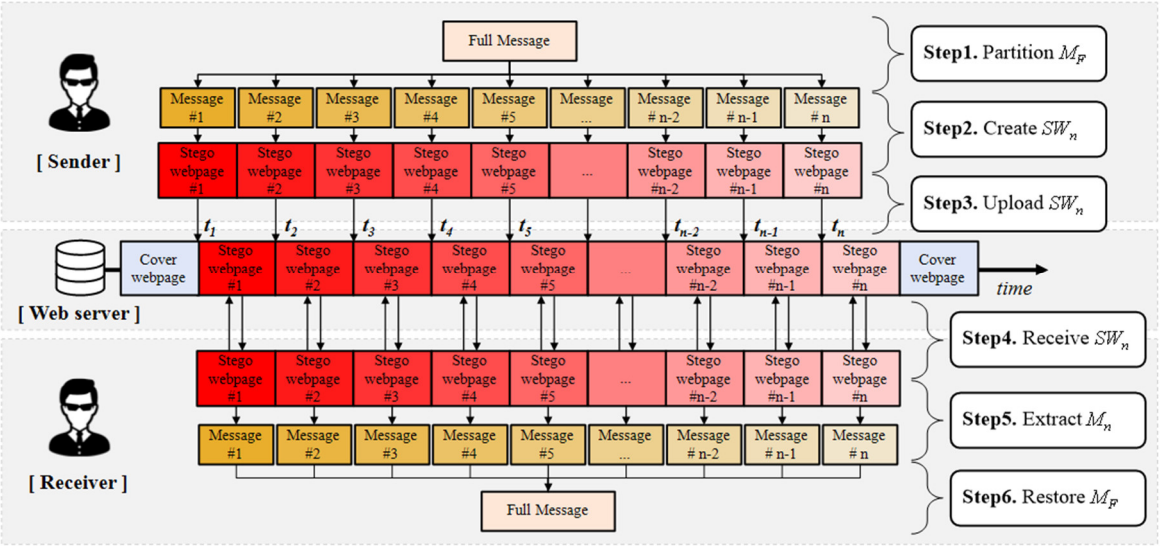


Figure 7. Working steps of Time-based Dynamic Operation Algorithm for hiding Capacity (TDOA-C).

By this manner, the existing method with TDOA-C can deliver $n \times c$ bits, which means our algorithm can improve the hiding capacity of the method in the static operation model by $n \times 100\%$. The hiding capacity (or the amount of delivered secret messages) $HC[TDOA-C]$ can be expressed in Equation (1) below.

$$HC[TDOA-C] = \sum_{i=1}^{n} HC(SW_i) = n \times c \quad (1)$$

Algorithm 1 TDOA-C (Time-Based Dynamic Operation Algorithm for Hiding Capacity)

Definition

Message	M_F: a full secret message to send to R
	$M_S = \{M_1, M_2, \dots, M_n\}$: a set of partitioned messages
	n: the number of partitioned messages
Webpage	CW: cover webpage
	$SW = \{SW_1, SW_2, \dots, SW_n\}$: a set of stego webpages
Time	$T = \{t_1, t_2, \dots, t_n\}$: a set of times that agree with R
	t_{start}, t_{end}: the start and end time of covert communication between S and R
	$t_{current}$: current time
Function	$embed(CW, M_n)$: embeds the secret message M_n into CW. returns SW_n
	$extract(SW_n)$: extracts the embedded secret message from SW_n. returns M_n
	$partition(M_F, n)$: partitions M_F into n smaller messages $M_1, M_2, \dots, M_n$. returns M_S

$restore(M_S)$: restores the M_F by combining $M_1, M_2, \ldots, M_n$. returns M_F

Sender	**Receiver**
Input: M_F, n, CW, T, t_{start}, t_{end}, $t_{current}$	**Input:** n, T, t_{start}, t_{end}, $t_{current}$
Output: M_S, SW	**Output:** SW, M_S, M_F
Functions: $partition(M_F, n)$, $embed(\mathrm{CW}, M_i)$	**Functions:** $extract(SW_n)$, $restore(M_S)$
1: **begin**	1: **begin**
2: $M_S \leftarrow partition(M_F, n)$ #Step1	2: **while** $t_{start} \le t_{current} < t_{end}$
3: **for** i from 1 to n **do** #Step2	3: **for** i from 1 to n **do** #Step4
4: $SW_i = embed(CW, M_i)$	4: **while** True #infinitely repeat
5: $SW \leftarrow SW_i$ #add an element SW_i into set SW	5: **if** $t_{current} == t_i$ **then**
6: **while** $t_{start} \le t_{current} < t_{end}$	6: request SW_i to a webserver
7: **for** j from 1 to n **do** #Step3	7: $SW \leftarrow SW_i$ from a webserver
8: **while** True #infinitely repeat	8: **break**
9: **if** $t_{current} == t_j$ **then**	9: **for** j from 1 to n **do** #Step5
10: upload SW_j to webserver	10: $M_j = extract(SW_j)$
11: **break**	11: $M_S \leftarrow M_j$ #add M_j into set M_S
12: **end**	12: $M_F = restore(M_S)$ #Step6
	13: **end**

3.3. Time-based Dynamic Operation Algorithm for Undetectability (TDOA-U)

We introduce another algorithm (TDOA-U) based on our TDOM and on-off strategy [37] for improving the undetectability of an existing webpage steganography method. Unlike the existing static operation model, by using this algorithm, the sender S controls when a stego webpage is activated (on) and deactivated (off) as time passes. Thus, as shown in Figure 8, only when the stego webpage SW is activated (on), the stego webpage SW can be accessed by the receiver R and, thus, a secret message can be delivered to the receiver R. Meanwhile, when a stego webage is deactivated (off), it is replaced with the original cover webpage CW without any secret message, and, thus, the monitor will not detect the stego webpage when it accesses CW. Consequently, the chance that the monitor detects the stego webpage will decrease according to the deactivated time of SW given a certain time period.

Our TDOA-U works as the following steps (see Algorithm 2). We assume the following. First, a monitor M with steg analysis capability is checking periodically if a webpage (a cover webpage CW) is a stego webpage. Second, a sender uses a dynamic webpage for a cover webpage to avoid a simple webpage change detector (or a file integrity checker) without a steganography detection function. Lastly, there are multiple receivers who want to receive the same secret message from the web server for a certain period of time.

1. **Step 1**. The sender S creates stego webpages SW_1, SW_2, …, SW_n by embedding a secret message into a cover webpage CW. We assume that n stego webpages have the same secret message.
2. **Step 2**. S uploads SW_1, SW_2, …, SW_n to the web server, in turn, at $t_{Si} \in T_S = \{t_{S1}, t_{S2}, \ldots, t_{Sn}\}$ and replace them with CW at $t_{Ci} \in T_C = \{t_{C1}, t_{C2}, \ldots, t_{Cn}\}$ in which both T_S and T_C are agreed in advance between S and R. For this step, the sender and receiver need to exchange such information by using another covert channel or in person. In addition, each SW_i will be replaced with CW again when a small amount of time passes to avoid the monitor. Therefore, the accurate time synchronization between S and R is very important so that this step can succeed. Moreover, to successfully get SW_i even in the presence of a possible time difference between S and R due to the unexpected network or processing delay, the receiver may need to access the web address of SW_i multiple times. Meanwhile, we do not delve into designing a sophisticated time synchronization and guaranteed delivery method for S and R in this study but, instead, we want to leave it for our future research.
3. **Step 3**. R receives each stego webpage, in turn, by accessing the URL where the cover webpage is located at the time period agreed with S.

4. **Step 4**. R can extract the secret message M from received stego webpages SW_1, SW_2, …, SW_n, which means that R can obtain M at any time between t_{Si} and t_{Ci}.

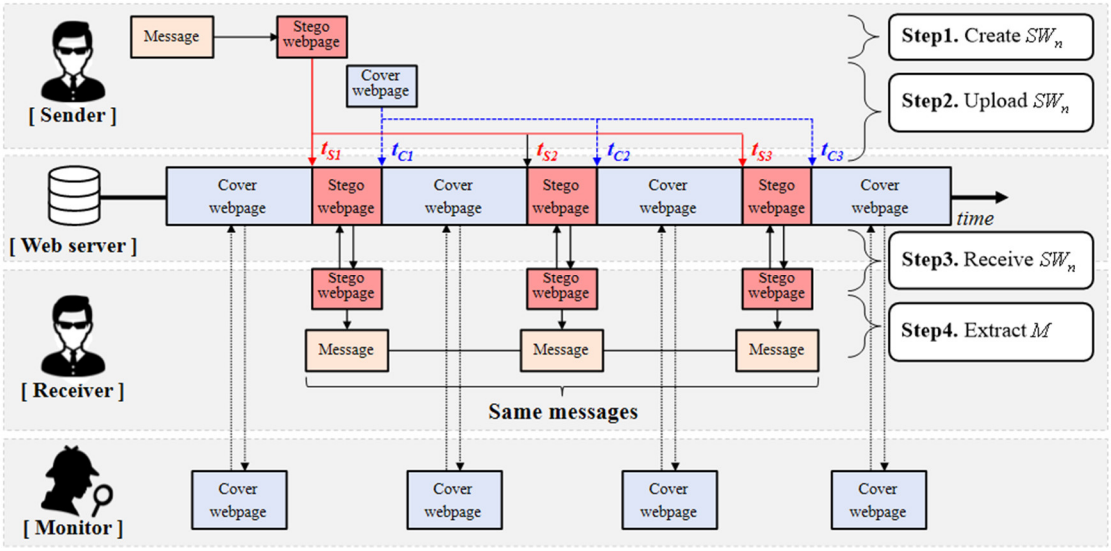


Figure 8. Working steps of Time-based Dynamic Operation Algorithm for Undetectability (TDOA-U).

Algorithm 2 TDOA-U (Time-based Dynamic Operation Algorithm for Undetectability)

Definition

Message M: a secret message to send to R
N_S: the number of stego webpage to send to R

Webpage CW: cover webpage
$SW = \{SW_1, SW_2, \ldots, SW_n\}$: a set of stego webpages

Time $T_S = \{t_{S1}, t_{S2}, \ldots, t_{Sn}\}$: a set of times, which S upload SW to the web server
$T_C = \{t_{C1}, t_{C2}, \ldots, t_{Cn}\}$: a set of times, which S upload CW to the web server
t_{start}, t_{end}: the start and end time of covert communication between S and R
$t_{current}$: current time

Function $embed(CW, M_n)$: embeds the secret message M_n into CW. returns SW_n
$extract(SW_n)$: extracts the embedded secret message from SW_n. returns M_n
$restore(M_S)$: restores the M_F by combining $M_1, M_2, \ldots, M_n$. returns M_F

Sender

Input: M, N_S, CW, T_S, T_C, t_{start}, t_{end}, $t_{current}$
Output: SW
Function: $embed(CW, M)$

```
1: begin
2:   for i from 1 to n do #Step1
3:     SW_i = embed(CW, M)
4:     SW ← SW_i #add an element SW_i into set SW
5:   while t_start ≤ t_current < t_end
6:     for j from 1 to N_S do #Step2
7:       while True #infinitely repeat
8:         if t_current == t_Sj then
9:           upload SW_j to webserver
10:        else if t_current == t_Cj then
11:          upload CW to webserver
12:        break
13: end
```

Receiver

Input: T_S, T_C, t_{start}, t_{end}, $t_{current}$
Output: SW, M
Functions: $extract(SW_n)$, $restore(M_S)$

```
1: begin
2:   while t_start ≤ t_current < t_end
3:     for i from 1 to n do #Step3
4:       while True #infinitely repeat
5:         if t_Si ≤ t_current < t_Ci then
6:           request SW_i to webserver
7:           SW ← SW_i from webserver
8:           break
9:   choice j from 1 to n #Step4
10:  M = extract(SW_j)
11: end
```

3.4. The Hybrid Model of Combining TDOA-C and TDOA-U

As described previously, we have designed our TDOA-C and TDOA-U for hiding capacity and undetectability, respectively. Meanwhile, these two algorithms can be combined in a hybrid manner depending on which factor of hiding capacity (the amount of secret message delivery) or undetectability is more necessary. Thus, as shown in Figure 9, the hybrid model can be designed and implemented in various ways by considering the following factors, such as the goal of covert communication between S and R, the expected monitoring and detection capability, the frequency, or cycle, the amount of a secret message or its delivery frequency, and so on. In this paper, we do not implement this hybrid model and consider it in our comparative experiments in Section 4, since our main goal of this study is to show how our idea (dynamic operation model) can improve existing webpage steganography methods rather than showing the optimized performance of our proposed model, which will be conducted for one of our future works.

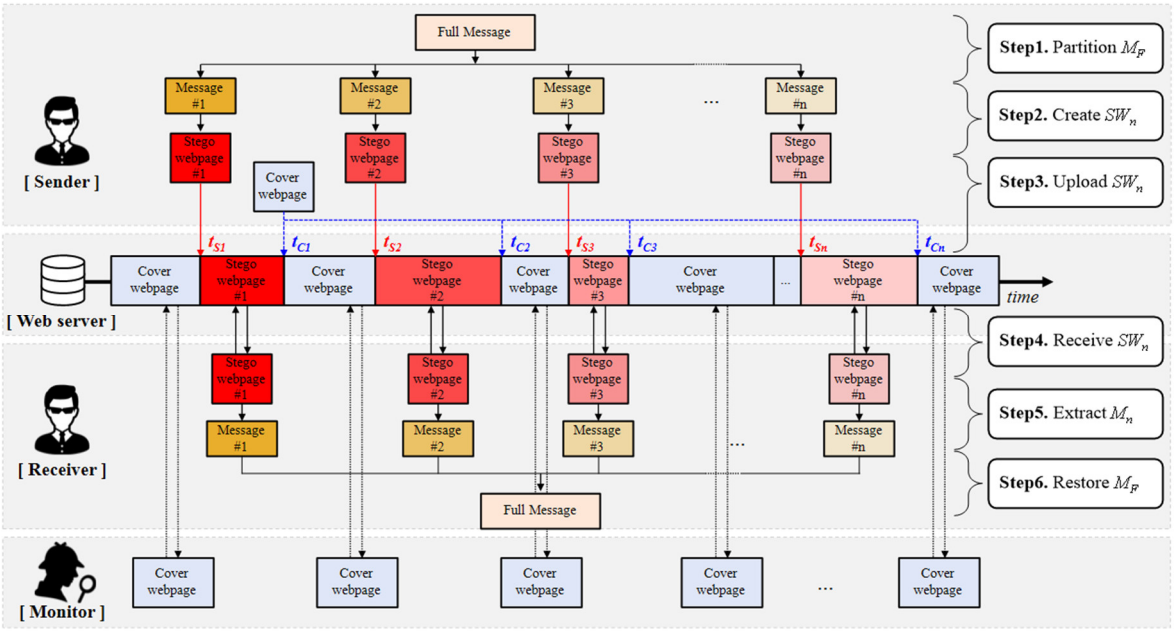


Figure 9. A hybrid approach that combines TDOA-C and TDOA-U.

4. Experimental Results

In this section, we conduct two experiments for the following purposes.

- Experiment 1: Validating TDOA-C has a higher hiding capacity (or a larger amount of secret message delivery) than the existing static model.
- Experiment 2: Validating TDOA-U has a higher undetectability than the existing static model.

We will describe details on the experimental purpose, methods, and results of each experiment. All our experiments were conducted with Python 3 on a laptop (Intel i5 10th GEN and 16GB RAM).

4.1. Experiment 1: Validation of TDOA-C

Experiment 1 consists of two parts (Part 1 and Part 2) as follows. In Part 1, we implement an existing method combined with our TDOA-C in a real web environment and show our model can work. In the second part, we conduct numerical analysis to validate our TDOA-C that will improve the hiding capacity of two representative existing methods (CCL and COA) compared with it in the static operation model.

4.1.1. Experiment 1. Part 1: Implementation of TDOA-C in the Web Environment

To show our TDOA-C works in a real web environment, we built a web server and then implemented TDOA-C according to Algorithm 1 (see Figure 10). We used the Flask web framework [38] to build a web server in our laptop and *urllib.request* library [39] to receive stego webpages from the webserver.

Based on the constructed web environment, we conducted Part 1 of Experiment 1 as follows (see Figure 10). For the secret message M_F, we generated 100 UUID (Universally Unique Identifiers) by using UUID version 4 algorithm [40] and used the combined 100 UUIDs for M_F. Each UUID consists of 32 hexadecimal numbers (16 bytes). Next, we created 100 SWs by embedding each UUID into a cover web page, uploaded each SW_i to the web server every 10 s, and then accessed and received SW_i from the web server. We confirm that our algorithm works in the web environment if the embedded and extracted UUIDs are the same.

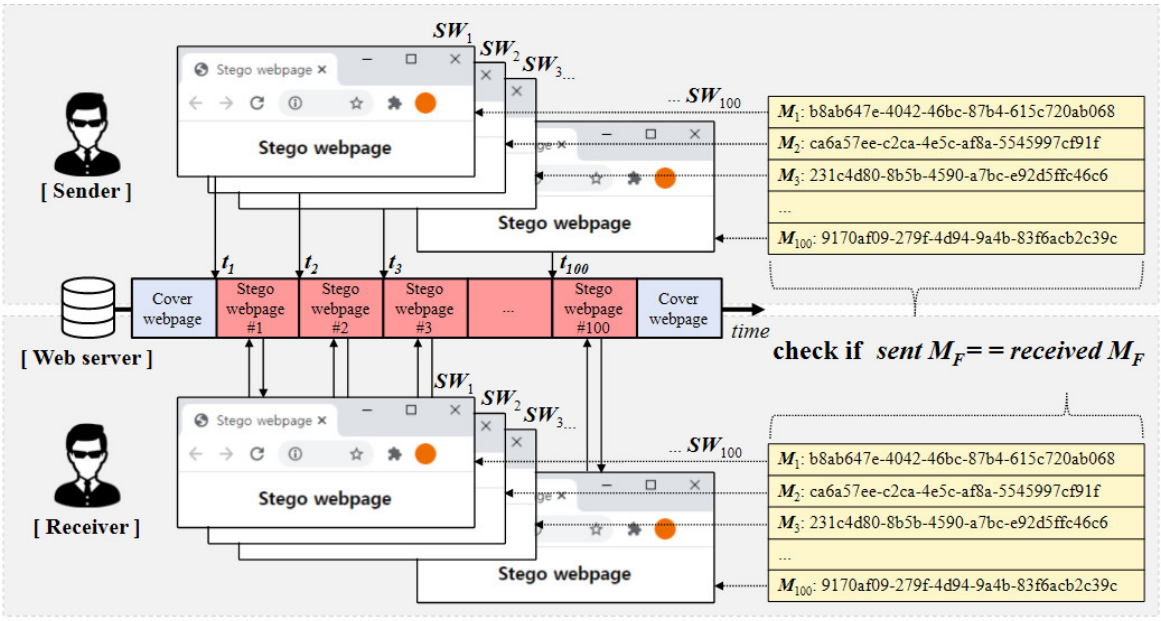


Figure 10. Experimental design of Experiment 1. Part 1.

The experimental results show that all 100 UUIDs are exactly matched. Table 1 shows part of the experimental results of part 1 for $n = 1, 2, 3, 4, 5, 10, 50,$ and 100. For example, for SW_1 ($n = 1$), the sender embedded the first UUID (b8ab647e-4042-46bc-87b4-615c720ab068) into CW and then generated SW_1, and the receiver could extract the same UUID from the received SW_1. According to our experimental results, since all 100 UUIDs are received and extracted successfully at the receiver, M_F (combined 100 UUIDs) is delivered correctly. Therefore, based on the experimental result, we confirmed our TDOA-C works properly in the web environment.

Table 1. A part of the result of experiment 1. Part 1.

	Secret Messages (UUID 4)		Match
n	**Sender Side (Embedded)**	**Receiver Side (Extracted)**	**Result**
1	b8ab647e-4042-46bc-87b4-615c720ab068	b8ab647e-4042-46bc-87b4-615c720ab068	matched
2	ca6a57ee-c2ca-4e5c-af8a-5545997cf91f	ca6a57ee-c2ca-4e5c-af8a-5545997cf91f	matched
3	231c4d80-8b5b-4590-a7bc-e92d5ffc46c6	231c4d80-8b5b-4590-a7bc-e92d5ffc46c6	matched
4	f3325710-e852-4bd2-837d-3d9ab67fbd5c	f3325710-e852-4bd2-837d-3d9ab67fbd5c	matched
5	8cf7a48b-72ec-4488-b165-d4c07a82c1fb	8cf7a48b-72ec-4488-b165-d4c07a82c1fb	matched
10	75bf76b3-0bd8-4887-8986-a18e9bad6b18	75bf76b3-0bd8-4887-8986-a18e9bad6b18	matched
50	78ccde2a-dbf4-4ddb-b3b9-111de53caaf3	78ccde2a-dbf4-4ddb-b3b9-111de53caaf3	matched
100	9170af09-279f-4d94-9a4b-83f6acb2c39c	9170af09-279f-4d94-9a4b-83f6acb2c39c	matched

4.1.2. Experiment 1. Part 2: Comparative Numerical Analysis

To show how our TDOA-C can improve the hiding capacity (or the amount of a secret message delivery) of two representative existing methods in the static operation model, we conducted the comparative numerical analysis as follows.

First, we measured the average hiding capacity of existing methods (CCL and COA) by considering the top popular global websites' main webpages. For cover webpages, we collected the top 50 webpages introduced by two popular websites Alexa [41] and SimilarWeb [42] that provide the global website ranks in terms of the number of visitors to websites.

For the CCL method, we implemented it by adopting Sui and Luo's CCL method [6] and measured the hiding capacity of CCL for each of the collected webpages as:

$$HC[CCL] = \sum_{n=1}^{N_{elements}} N_{letters}(e_n) \quad (2)$$

where e_n denotes nth elements in the webpage, and $N_{letters}$ denotesa function that returns to the number of alphabetic letters except to attribute values and contents in an element, and $N_{elements}$ denotes the number of elements in a webpage. We averaged all obtained $HC[CCL]$.

In addition, for the COA method, we implemented it by adopting Huang et al.'s COA method [25], which measured the hiding capacity of COA for each of the collected webpages as:

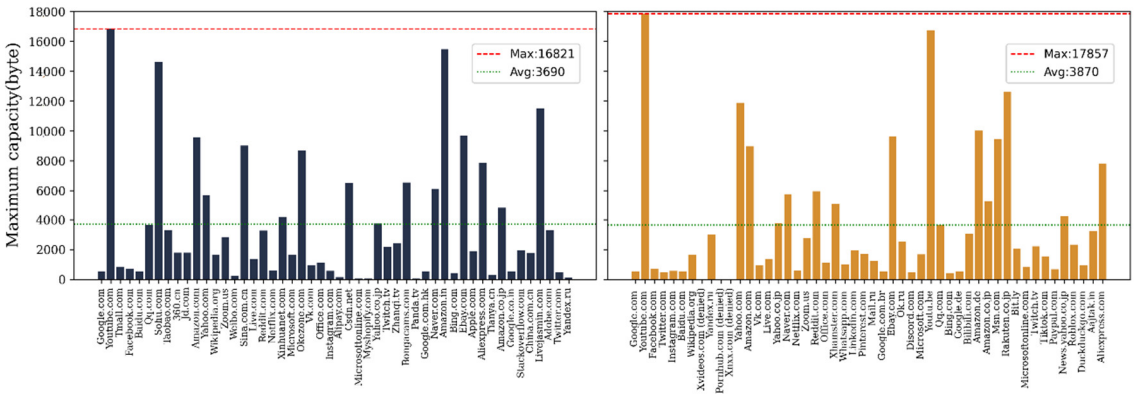
$$HC[COA] = \sum_{n=1}^{N_{tags}} \lfloor \log_2\{N_{attributes}(t_n)\}! \rfloor \quad (3)$$

where t_n is the nth tag in a webpage, $N_{attributes}(t_n)$ is a function that returns the number of attributes in t_n, and N_{tags} is the number of tags within the webpage. If $N_{attribute}(t_n) \geq 2$, the maximum number of permutations is equal to $\{N_{attribute}(t_n)\}!$, and, thus, the hiding capacity of t_n is equal to $\lfloor \log_2\{N_{attributes}(t_n)\}! \rfloor$ bits. We averaged all obtained values of $HC[COA]$.

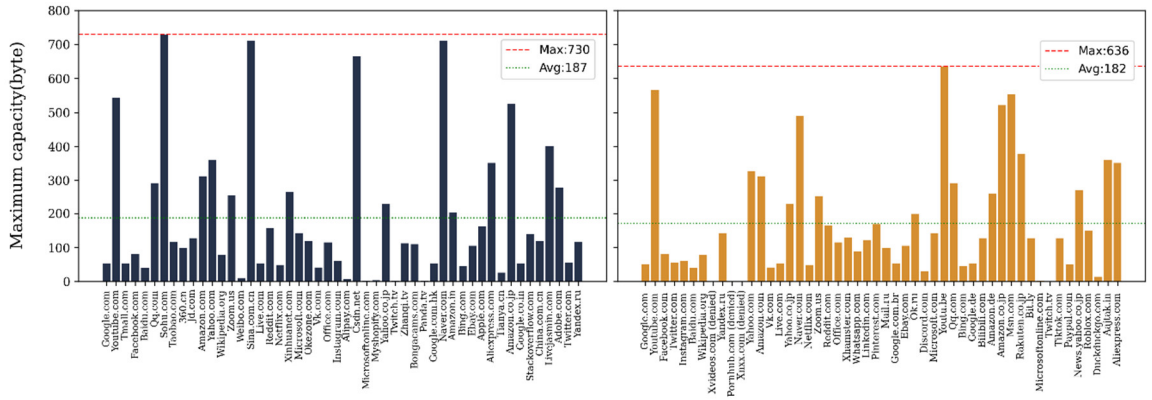
All measured values of the hiding capacity of existing methods (CCL and COA) are shown in Table 2 and Figure 11.

Table 2. The average and max hiding capacity measured for top websites when Changing the Case of Letters (CCL) in tags and attributes and Changing the Order of Attributes (COA) are used.

Hiding Capacity (Bytes)	CCL		COA	
	Alexa	SimilarWeb	Alexa	SimilarWeb
Max (Max_{HC})	16,821	17,857	730	636
Average (μ_{HC})	3690	3870	187	182
Standard Deviation (σ_{HC})	4028	4303	199	163



(**a**) CCL—Alexa (**b**) CCL—SimilarWeb



(**c**) COA—Alexa (**d**) COA—SimilarWeb

Figure 11. The capacity of Changing the Case of Letters (CCL) in tags and attributes and Changing the Order of Attributes (COA) from each website list.

Second, we compared the average hiding capacity calculated of the existing two methods (CCL and COA) in the static operation model with our approaches (CCL with TDOA-C and COA with TDOA-C). For our approaches, we measured the average hiding capacity of our two methods as the change cycle time (C_t) decreases from 100 to 10 by 10. For simplicity, we set C_t to a fixed value between 10 and 100, which means the interval of t_i and t_{i+1} is equally set where $t_i \in T$ *and* $1 \le i \le n-1$ (see Algorithm 1). For example, if C_t is 2 s, an existing stego webpage SW_i will be updated with the next stego webpage SW_{i+1} every two seconds in our dynamic operation model. Thus, for 10 s, five different stego webpages (i.e., the number of stego webpages $N_s = 5$) will be uploaded to the web server while only one stego webpage is exposed in the static operation model regardless of C_t. Based on the above settings, we conducted the comparative numerical analysis in terms of hiding capacity by using various operation times (1000 s, 2000 s, 5000 s, and 10,000 s). The operation time is $t_{end} - t_{start}$.

We now explain our experimental results as follows (see Table 3 and Figure 12).

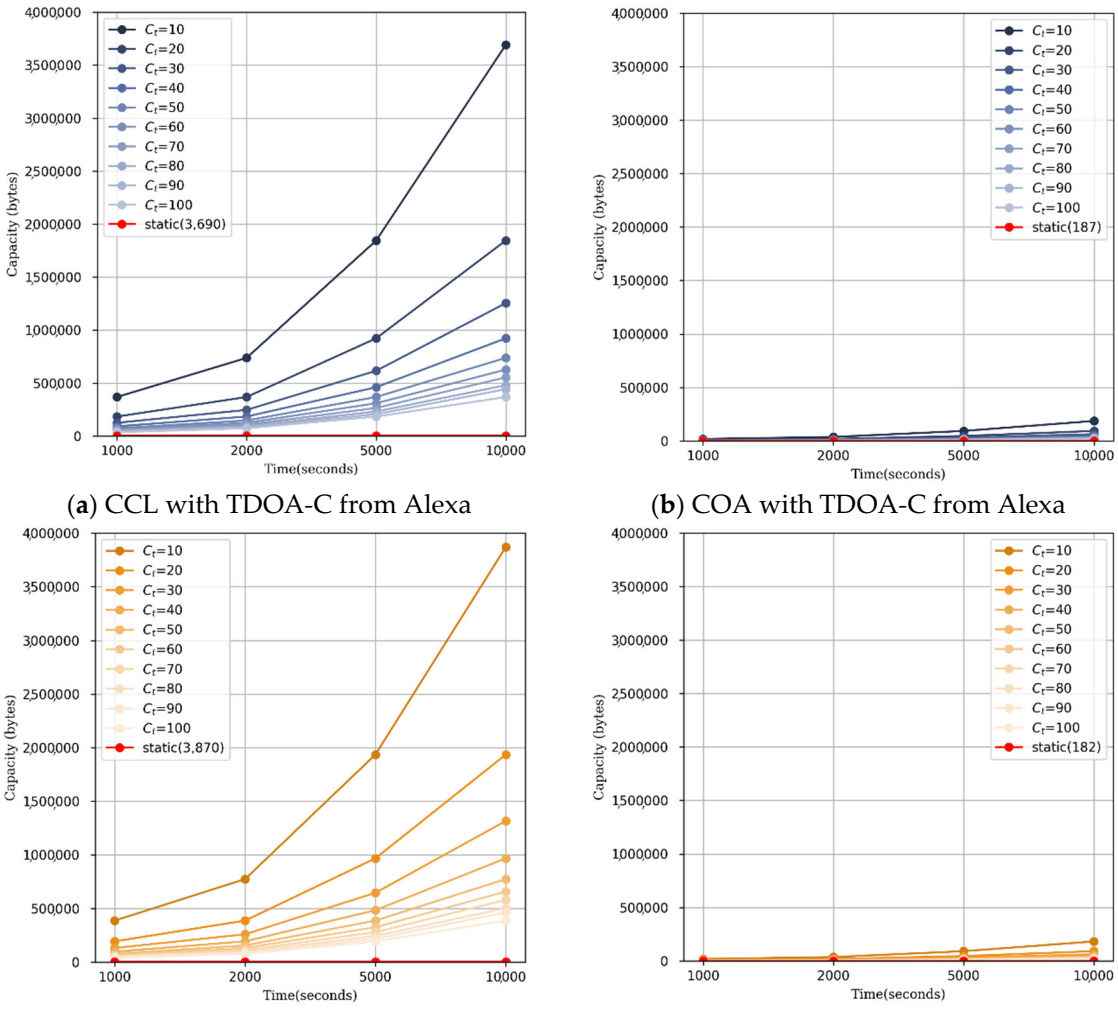
Table 3. The capacity of TDOA-C and existing static model with CCL and COA in 1000 s. The unit of capacity is bytes. N_S denotes the number of transmitted stego webpages. C_t denotes the cycle time that the period of a stego webpage is changed to the next stego webpage.

	Existing Static Model			**CCL with TDOA-C**									
C_t		∞		**100**	**90**	**80**	**70**	**60**	**50**	**40**	**30**	**20**	**10**
N_S		**1**		**10**	**12**	**13**	**15**	**17**	**20**	**25**	**34**	**50**	**100**
Website list	**Alexa**	**Avg**	3690	36,900	44,280	47,970	55,350	62,730	73,800	92,250	125,460	184,500	369,000
		Max	16,821	168,210	201,852	218,673	252,315	285,957	336,420	420,525	571,914	841,050	1682,100
	Similar Web	**Avg**	3870	38,700	46,440	50,310	58,050	65,790	77,400	96,750	131,580	193,500	387,000
		Max	17,857	178,570	214,284	232,141	267,855	303,569	357,140	446,425	607,138	892,850	1,785,700
	Existing Static Model			**COA with TDOA-C**									
C_t		∞		**100**	**90**	**80**	**70**	**60**	**50**	**40**	**30**	**20**	**10**
N_S		**1**		**10**	**12**	**13**	**15**	**17**	**20**	**25**	**34**	**50**	**100**
Website list	**Alexa**	**Avg**	187	1870	2244	2431	2805	3179	3740	4675	6358	9350	18,700
		Max	730	7300	8760	9490	10,950	12,410	14,600	18,250	24,820	36,500	73,000
	Similar Web	**Avg**	182	1820	2184	2366	2730	3094	3640	4550	6188	9100	18,200
		Max	636	6360	7632	8268	9540	10,812	12,720	15,900	21,624	31,800	63,600

First, for all value of C_t, the average hiding capacities of our proposed algorithms (CCL with TDOA-C and COA with TDOA-C) are much higher than those of the existing two methods (CCL and COA). Table 3 shows the analysis results when the operation time = 1000 s. For example, when the top 50 websites from Alexa is considered, the average hiding capacities (μ_{HC}) of CCL and COA

are fixed as 3690 bytes and 187 bytes. $\mu_{HC}[CCL]$ = 3690 bytes and $\mu_{HC}[COA]$ = 187 bytes. This is because only one stego webpage is exposed ($N_s = 1$) regardless of the operation time. On the other hand, when $C_t = 100$, our methods (CCL with TDOA-C and COA with TDOA-C) are 10 times better than CCL and COA in terms of the average hiding capacity because, when $C_t = 100$, 10 different stego webapges are uploaded for 1000 s. As a result, $\mu_{HC}[CCL\ with\ TDOA-C]$ = 36,900 bytes and $\mu_{HC}[COA\ with\ TDOA-C]$ = 1870 bytes. Moreover, the difference of the average hiding capacity of existing methods and our methods keep increasing as C_t decreases because N_s grows as C_t decreases, as we can see in Table 3 and Figure 12. In case of a SimilarWeb, the analysis results are shown similarly.

Second, the difference between the hiding capacity of existing methods and our methods increases as the operation time increases from 1000 s, 2000 s, 5000 s, and 10,000 s. For example, Figure 12 shows $\mu_{HC}[CCL]$, $\mu_{HC}[COA]$, and $\mu_{HC}[CCL\ with\ TDOA-C]$, and $\mu_{HC}[COA\ with\ TDOA-C]$ as the increment of operation time grows when the top 50 websites from Alexa (Figure 12a,b) and SimilarWeb (Figure 12c,d)) are considered. We can see that, as the operation time increases from 1000 s to 2000 s, the $\mu_{HC}[CCL\ with\ TDOA-C]$ and $\mu_{HC}[COA\ with\ TDOA-C]$ doubles. Likewise, when the operation time increases from 1000 s to 5000 s and from 1000 s to 10,000 s the $\mu_{HC}[CCL\ with\ TDOA-C]$ and $\mu_{HC}[COA\ with\ TDOA-C]$ grow by five times and 10 times, respectively. However, no matter how much operation time increases, $\mu_{HC}[CCL]$ and $\mu_{HC}[COA]$ were constant.



(**a**) CCL with TDOA-C from Alexa

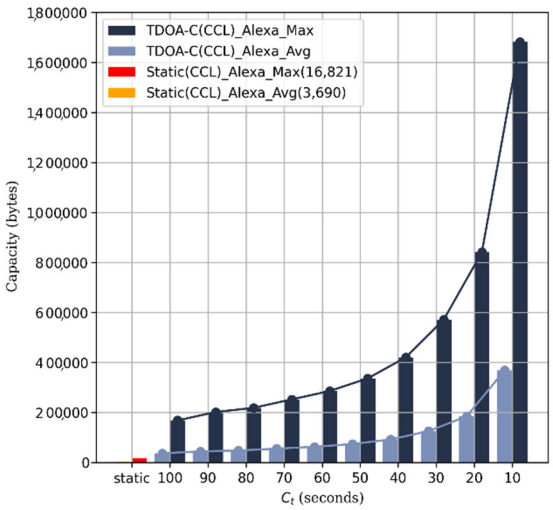
(**b**) COA with TDOA-C from Alexa

(**c**) CCL with TDOA-C from a SimilarWeb

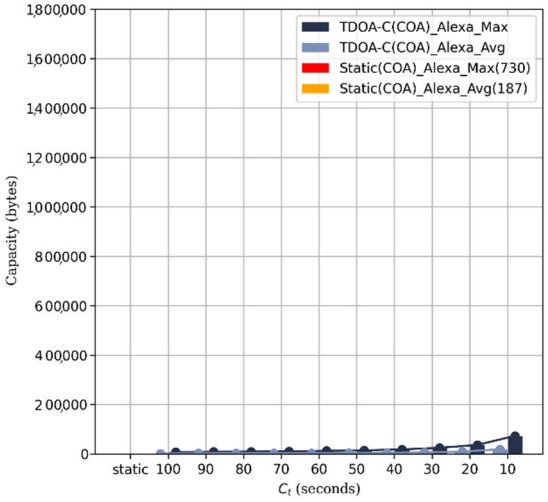
(**d**) COA with TDOA-C from a SimilarWeb

Figure 12. Average hiding capacities of our methods (CCL with TDOA-C and COA with TDOA-C) and existing methods (CCL and COA) as the value of the operation time increases from 1000 s to 10,000 s.

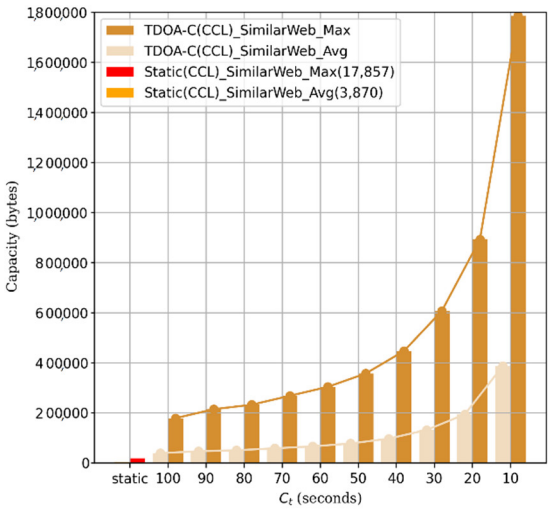
Third, the hiding capacity of CCL with TDOA-C is much larger than the capacity of COA with TDOA-C. For example, Figure 13a,b shows $Max_{HC}[CCL\ with\ TDOA-C], \mu_{HC}[CCL\ with\ TDOA-C]$ and $Max_{HC}[COA\ with\ TDOA-C], \mu_{HC}[COA\ with\ TDOA-C]$ when the top 50 websites from Alexa are considered. For all values of C_t, $Max_{HC}[CCL\ with\ TDOA-C] > Max_{HC}[COA\ with\ TDOA-C]$ and $\mu_{HC}[CCL\ with\ TDOA-C] > \mu_{HC}[COA\ with\ TDOA-C]$. In addition, as C_t decreases, both $Max_{HC}[CCL\ with\ TDOA-C] - Max_{HC}[COA\ with\ TDOA-C]$ and $\mu_{HC}[CCL\ with\ TDOA-C] - \mu_{HC}[COA\ with\ TDOA-C]$ also increase. In the case of SimilarWeb as shown in Figure 13c,d, such a tendency holds as well. This is because the CCL method can embed data more into a cover webpage than the COA method, as we discussed above (see Table 3).



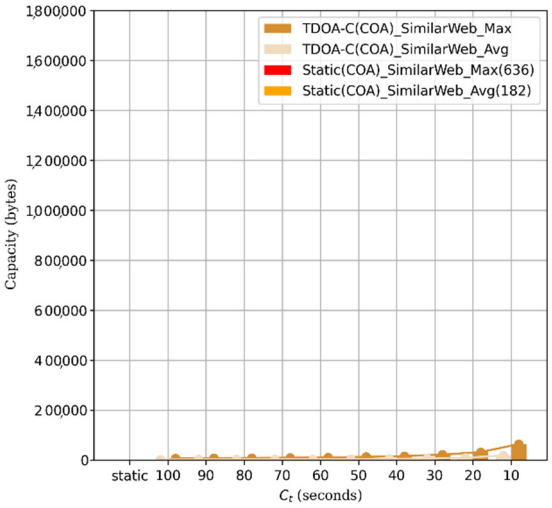
(**a**) CCL with TDOA-C from Alexa



(**b**) COA with TDOA-C from Alexa



(**c**) CCL with TDOA-C from SimilarWeb

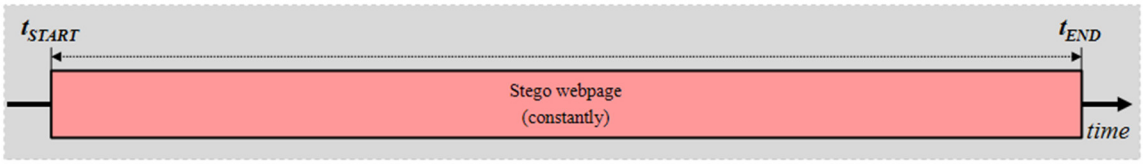


(**d**) COA with TDOA-C from SimilarWeb

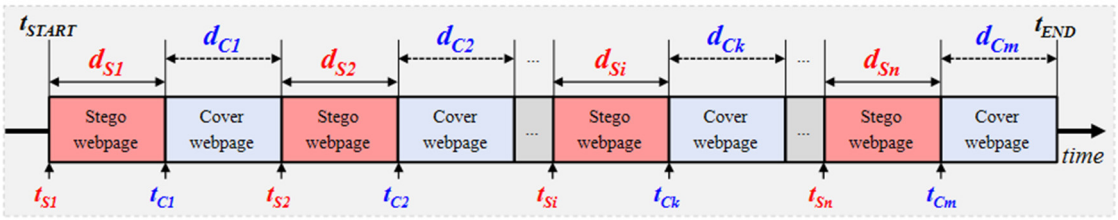
Figure 13. The capacity of our methods (CCL with TDOA-C and COA with TDOA-C) and existing methods (CCL and COA) according to C_t.

4.2. Experiment 2: Validation of TDOA-U

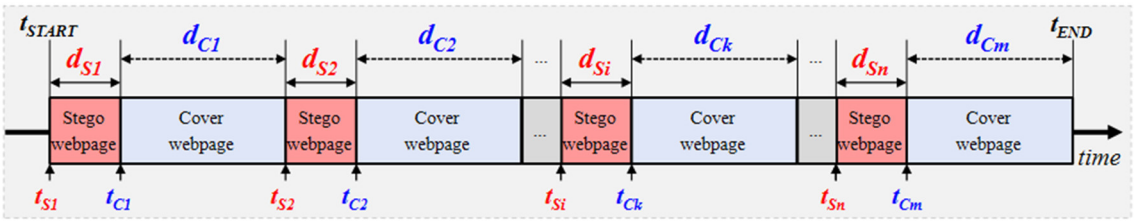
The purpose of Experiment 2 is to show how our TDOA-U can improve the undetectability of an existing method in the static operation model. The concept to measure undetectability is depicted in Figure 14.



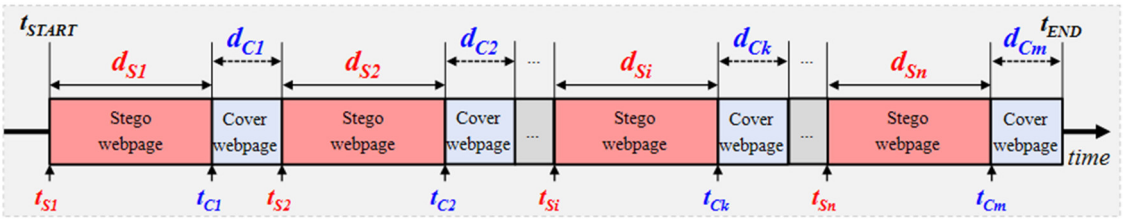
(**a**) Existing method in the static operation model.



(**b**) Case 1. ($\sum_{i=1}^{|D_S|} d_{Si} = \sum_{k=1}^{|D_C|} d_{Ck}$)



(**c**) Case 2. ($\sum_{i=1}^{|D_S|} d_{Si} < \sum_{k=1}^{|D_C|} d_{Ck}$)



(**d**) Case 3. ($\sum_{i=1}^{|D_S|} d_{Si} > \sum_{k=1}^{|D_C|} d_{Ck}$)

Figure 14. Concept to measure undetectability.

We consider that an existing method in the static operation model has zero undetectability because, as shown in Figure 14a, it will always be detected by a monitor since there is no concealed time period between t_{start} and t_{end}. In addition, we consider that a method in Case 2 has higher undetectability than a method in Case 3 because Case 2's exposure time to users including a monitor is smaller than Case 3 given the same operation time period ($= t_{end} - t_{start}$). Thus, the chance that Case 2 is not detected by the monitor is lower than the chance that Case 3 is not detected. To compare two methods quantitatively in terms of undetectability, we define undetectability M_U as:

$$M_U = 1 - \left(\frac{\sum_{i=1}^{|D_S|} d_{Si}}{\sum_{i=1}^{|D_S|} d_{Si} + \sum_{k=1}^{|D_C|} d_{Ck}} \right) = \frac{\sum_{k=1}^{|D_C|} d_{Ck}}{t_{end} - t_{start}} \quad (4)$$

where d_{Si} is the exposure time period of the ith stego webpage, D_S is a set of $d_{Si}\ for\ i \in [1, n]$, d_{Ck} is the exposure time period of the kth cover webpage, D_C is a set of $d_{Ck}\ for\ k \in [1, m]$, and $M_U \in [0, 1]; |D_S| = n\ and\ |D_C| = m$. This metric indicates the ratio of the amount of time that a stego webpage is not exposed to the total operation time. Thus, a method has its maximum undetectability when $M_U = 1$, and, for the existing method in the static operation model, $M_U = 0$.

To see how differently our proposed methods has M_U depending on D_S and D_C, we conducted the comparative numerical analysis as follows. For simplicity, we set $d_{S1} = d_{S2} = \cdots = d_{Si} = \cdots =$

d_{Sn} and $d_{C1} = d_{C2} = \cdots = d_{Ck} = \cdots = d_{Cm}$, and the total operation time ($= t_{end} - t_{start}$) = 1000 s. In addition, we used various values for d_{Si} and d_{Ck} from 10 s to 100 s by 10 s.

In addition, we implemented our CCL with TDOA-U and conducted an experiment in a real web environment to see if its M_U measured in the real web environment is similar with M_U calculated in our numerical analysis (see Figure 15). The experimental environment is the same as Experiment 1, Part 1. To create an actual stego webpage, we used the main webpage of Google as a cover webpage, and then embedded the secret string "STEGANOGRAPHY" (91 bits) by using the CCL method. The total operation time period ($= t_{end} - t_{start}$) is 100 s, and we considered three cases: Case 1 (d_{Si}= 50 s and d_{Ck}= 50 s), Case 2 (d_{Si}= 30 s and d_{Ck}= 70 s), and Case 3 (d_{Si}= 70 s and d_{Ck}= 30 s). In addition, we implemented a monitor such that it accesses the webpage one time randomly during the operation time of 100 s and then check if the accessed time point is in the stego webpage's exposed time period. The experiment was repeated 100 rounds, and then we measured M_U as the total number of detections of the stego webpage over the total number of accesses.

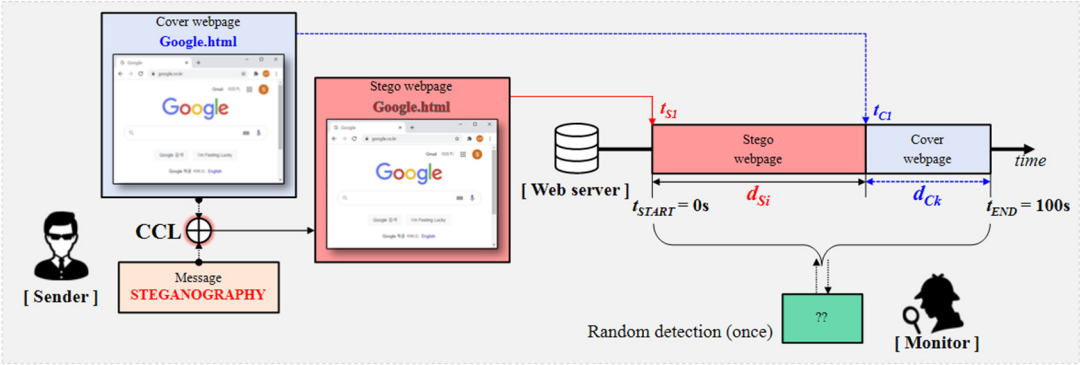


Figure 15. An experiment to measure M_U in the real web environment.

We now explain the results of Experiment 2 (see Tables 4 and 5).

First, our proposed method TDOA-U has higher undetectability than the existing method in the static operational model. As we can see in Table 4, for all values of d_{Si} and d_{Ck}, all measured values of M_U were higher than zero in our experimental settings. That means that our proposed method has higher undetectability than the existing method in the static operation model. Therefore, we confirmed that our TDOA-U can improve the undetectability of the existing webpage steganography methods.

Second, depending on the values of d_{Si} and d_{Ck} of TDOA-U, the undetectability M_U varies. As shown in Table 4, M_U is blue-colored when $M_U > 0.6$ and M_U is red-colored when $M_U < 0.4$. As we expected, when $d_{Si} > d_{Ck}$, $M_U \geq 0.5$ because the total amount of exposure time of CW is greater than the total amount of exposure time of SW during the operation time of 1000 s. On the other hand, when $d_{Si} < d_{Ck}$, $M_U \leq 0.5$ because the total amount of exposure time of CW is lower than the total amount of exposure time of SW during the operation time of 1000 s. In addition, when $d_{Si} = d_{Ck}$, $M_U \approx 0.5$ because the total amount of exposure time of CW is almost equal to the total amount of exposure time of CW during the operation time of 1000 s. There were some small deviations from 0.5 due to the fixed operation time of 1000 s (e.g., when $d_{Si} = 30$ s and $d_{Ck} = 30$ s, $M_U = 0.49$). We can use d_{Si} and d_{Ck} for various purposes. For example, for higher undetectability, we can set $d_{Ck} > d_{Si}$ and, for higher secret message delivery, we can set $d_{Ck} < d_{Si}$.

Table 4. Measured undetectability M_U of our TDOA-U according to values of d_{Si} and d_{Ck}. M_U is blue-colored when $M_U > 0.6$ and MU is red-colored when $M_U < 0.4$.

		d_{Si}									
		10	**20**	**30**	**40**	**50**	**60**	**70**	**80**	**90**	**100**
d_{Ck}	**10**	0.50	0.33	0.25	0.20	0.16	0.14	0.12	0.11	0.10	0.09
	20	0.66	0.50	0.40	0.32	0.28	0.24	0.22	0.20	0.18	0.16
	30	0.75	0.60	0.49	0.42	0.36	0.33	0.30	0.27	0.24	0.21
	40	0.80	0.66	0.56	0.48	0.44	0.40	0.36	0.32	0.28	0.28
	50	0.83	0.70	0.61	0.55	0.50	0.45	0.40	0.36	0.35	0.30
	60	0.85	0.74	0.66	0.60	0.54	0.48	0.44	0.42	0.37	0.36
	70	0.87	0.77	0.70	0.63	0.56	0.52	0.49	0.44	0.42	0.40
	80	0.88	0.80	0.72	0.64	0.60	0.56	0.51	0.48	0.46	0.40
	90	0.90	0.81	0.73	0.68	0.63	0.58	0.54	0.52	0.46	0.45
	100	0.90	0.82	0.76	0.70	0.65	0.60	0.58	0.52	0.50	0.50

Third, the undetectability $M_{U,WEB}$ measured in the real web environment is similar to the $M_{U,NA}$ calculated in our numerical analysis (see Table 5). For Case 1, Case 2, and Case 3, the measured $M_{U,WEB}$ is 0.49, 0.67, and 0.29, respectively, and they are similar to the $M_{U,NA}$ calculated for Case 1, Case 2, and Case 3. The small difference $M_{U,NA}$ and $M_{U,WEB}$ between each case can be ignored.

Table 5. $M_{U,NA}$ and $M_{U,WEB}$.

Case	Numerical Analysis			Web Experiment		
	Set (s)		$M_{U,NA}$	Number of Detections		$M_{U,WEB}$
	d_{Si}	d_{Ck}		*SW*	*CW*	
Case 1. ($\sum_{i=1}^{\lvert D_S \rvert} d_{Si} = \sum_{k=1}^{\lvert D_C \rvert} d_{Ck}$)	50	50	0.5	51/100	49/100	0.49
Case 2. ($\sum_{i=1}^{\lvert D_S \rvert} d_{Si} < \sum_{k=1}^{\lvert D_C \rvert} d_{Ck}$)	30	70	0.7	33/100	67/100	0.67
Case 3. ($\sum_{i=1}^{\lvert D_S \rvert} d_{Si} > \sum_{k=1}^{\lvert D_C \rvert} d_{Ck}$)	70	30	0.3	71/100	29/100	0.29

5. Conclusions and Future Works

In this paper, to improve the hiding capacity or undetectability of existing webpage steganography, we proposed a time-based dynamic operation model (TDOM) that dynamically replaces the stego webpage with other stego webpages or the original webpage. We designed two time-based dynamic operation algorithms (TDOA-C and TDOA-U), which improve the hiding capacity of existing methods and TDOA-U for improving the undetectability of existing methods, respectively. In addition, we validated and showed the performance of our proposed methods, conducted extensive comparative experiments and numerical analysis by implementing two webpage steganography methods with our TDOM (CCL with TDOA-C and COA with TDOA-C), and tested them in the web environment.

Our future research directions are as follows. First, we will consider a spatial factor for our dynamic operation model by studying the concept of recent moving target defense techniques [43,44] to better improve the hiding capacity or the undetectability of existing webpage steganography methods. Second, we will devise a new webpage steganography method that overcomes the limitations of existing webpage steganography methods, and then combine it with our dynamic operation model. Third, we will design a secured and sophisticated time synchronization method for the sender and receiver. Fourth, we will study randomizing the change-replacement pattern of the stego webpage and the cover webpage to improve the undetectability of our dynamic operation model. Lastly, we will design the hybrid model to combine TDOA-C and TDOA-U and examine how the hybrid approach can improve the performance of existing methods.

Author Contributions: Conceptualization, S.Y. and Y.C. Methodology, Y.C. Software, S.Y. Validation, S.Y. Formal analysis, S.Y. and Y.C. Investigation, S.Y. Writing—original draft preparation, S.Y. Writing—review and editing, Y.C. Visualization, S.Y. Supervision, Y.C. Project administration, Y.C. Funding acquisition, Y.C. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Republic of Korea Air Force Academy Research Fund, grant number ROKAFA 20-A-1.

Acknowledgments: An earlier version of this paper was presented and selected as one of the outstanding presentation papers at the KIISE Korea Software Congress 2019 (KSC 2019) in December 2019, South Korea [45]. The authors would like to thank the editor and reviewers for their valuable comments and constructive suggestions.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Johnson, N.F.; Jajodia, S. Exploring steganography: Seeing the unseen. *Computer* **1998**, *31*, 26–34.
2. Bender, W.; Gruhl, D.; Morimoto, N.; Lu, A. Techniques for data hiding. *IBM Syst. J.* **1996**, *35*, 313–336.
3. Steganography: A Close View of the Traditional Attack Technique that Has Created Chaos in the Cybersecurity World. Available online: https://cyware.com/news/steganography-a-close-view-of-the-traditional-attack-technique-that-has-created-chaos-in-the-cybersecurity-world-d412d190 (accessed on 2 December 2020).
4. Steganography Anchors Pinpoint Attacks on Industrial Targets. Available online: https://threatpost.com/steganography-pinpoint-attacks-industrial-targets/156151/ (accessed on 2 December 2020).
5. Steganography in Attacks on Industrial Enterprises. Available online: https://ics-cert.kaspersky.com/reports/2020/06/17/steganography-in-attacks-on-industrial-enterprises/ (accessed on 2 December 2020).
6. Sui, X.G.; Luo, H. *A new steganography method based on hypertext. In Proceedings of 2004 Asia-Pacific Radio Science Conference, Qingdao, China, 24–27 August 2004*; IEEE: New York, NY, USA, 2004; pp.181–184.
7. Platinum Is Back. Available online: https://securelist.com/platinum-is-back/91135/ (accessed on 2 December 2020).
8. Cox, I.; Miller, M.; Bloom, J.; Fridrich, J.; Kalker, T. *Digital Watermarking and Steganography*; Morgan kaufmann: San Francisco, CA, USA, 2007.
9. Rafat, K.F. Cutting Edge Steganography Using HTML Document-An Appraisal. *Int. J. Comput. Sci. Inf. Secur.* **2016**, *14*, 960.
10. Odeh, A.; Elleithy, K.; Faezipour, M.; Abdelfattah, E. Novel Steganography over HTML Code. In *Innovations and Advances in Computing, Informatics, Systems Sciences, Networking and Engineering*; Springer: Cham, Switzerland, 2015; pp. 607–611.
11. Kis, D.; Pataki, N. Source Code-based Steganography. In Proceedings of the 10th International Conference on Applied Informatics, Eger, Hungary, 30 January–1 February, 2017; pp. 157–162.
12. Katzenbeisser, S.; Petitcolas, F.A.P. *Digital Watermarking*; Artech House: London, UK, 2000; Volume 2.
13. Lee, I.S.; Tsai, W.H. Secret communication through webpages using special space codes in HTML files. *Int. J. Appl. Sci. Eng.* **2008**, *6*, 141–149.
14. Chou, Y.C.; Huang, C.Y.; Liao, H.C. A reversible data hiding scheme using cartesian product for HTML file. In Proceedings of 2012 Sixth International Conference on Genetic and Evolutionary Computing, Kitakyushu, Japan, 25–28 August 2012; IEEE: New York, NY, USA, 2012; pp. 153–156.
15. Imran, S.; Khan, A.; Ahmad, B. Text Steganography Utilizing XML, HTML and XHTML Markup Languages. *Int. J. Inf. Technol. Secur.* **2017**, *9*, 99–116.
16. Tariq, M.A.; Khan, A.T.A.A.; Ahmad, B. Boosting the Capacity of Web based Steganography by Utilizing Html Space Codes: A blind Steganography Approach. *IT Ind.* **2017**, *5*, 29–36.
17. Bajaj, I.; Aggarwal, R.K. RSA Secured Web Based Steganography Employing HTML Space Codes and Compression Technique. In Proceedings of the 2019 International Conference on Intelligent Computing and Control Systems (ICCS), Madurai, India, 15–17 May 2019; IEEE: New York, NY, USA, 2019; pp. 865–868.

18. Jaiswal, R.J.; Patil, N.N. Implementation of a new technique for web document protection using unicode. In Proceedings of the 2013 International Conference on Information Communication and Embedded Systems (ICICES), Chennai, India, 21–22 February 2013; IEEE: New York, NY, USA, 2013, pp. 69–72.
19. Zhao, Q.; Lu, H. A PCA-based watermarking scheme for tamper-proof of webpages. *Pattern Recognit.* **2005**, *38*, 1321–1323.
20. Zhao, Q.; Lu, H. PCA-based webpage watermarking. *Pattern Recognit.* **2007**, *40*, 1334–1341.
21. Wu, C.C.; Chang, C.C.; Yang, S.R. An efficient fragile watermarking for webpages tamper-proof. In *Advances in Web and Network Technologies, and Information Management*; Springer: Berlin/Heidelberg, Germany, 2007; pp. 654–663.
22. Junling, R.; Chengquan, W. A Webpage information hiding algorithm based on tag dictionary. In Proceedings of the 2012 International Conference on Computer Science and Electronics Engineering, Hangzhou, China, 23–25 March 2012; IEEE: New York, NY, USA, 2012; pp. 546–550.
23. Ghosh, S. *StegHTML: A message hiding mechanism in HTML tags*; Technical Report: Charlottesville, VA, USA, 10 December 2007.
24. Shen, D.; Zhao, H. A novel scheme of webpage information hiding based on attributes. In Proceedings of the 2010 IEEE International Conference on Information Theory and Information Security, Austin, TX, USA, 13–18 June 2010; IEEE: New York, NY, USA, 2010; pp. 1147–1150.
25. Huang, H.; Zhong, S.; Sun, X. An algorithm of webpage information hiding based on attributes permutation. In Proceedings of the 2008 International Conference on Intelligent Information Hiding and Multimedia Signal Processing, Harbin, China, 15–17 August 2008; IEEE: New York, NY, USA, 2008; pp. 257–260.
26. Reddy, B.S.; Kuppusamy, K.S.; Sivakumar, T. Towards Web page steganography with Attribute Truth Table. In Proceedings of the 2016 3rd International Conference on Advanced Computing and Communication Systems (ICACCS), Coimbatore, India, 22–23 January 2016; IEEE: New York, NY, USA, 2016; pp. 1–5.
27. Singh, R.K.; Alankar, B. A Novel Approach For Data Hiding In Web Page Steganography Using Encryption With Compression Based Technique. *IOSR J. Comput. Eng.* **2016**, *18*, 73–77.
28. Yang, Y.J.; Yang, Y.M. An efficient webpage information hiding method based on tag attributes. In Proceedings of the 2010 Seventh International Conference on Fuzzy Systems and Knowledge Discovery, Yantai, China, 10–12 August 2010; IEEE: New York, NY, USA, 2010; pp. 1181–1184.
29. Yong, X.; Juan, L.; Yilai, Z. A high capacity information hiding method for webpage based on tag. In Proceedings of the 2012 Third International Conference on Digital Manufacturing & Automation, Guilin, China, 31 July–2 August 2012; IEEE: New York, NY, USA, 2012; pp. 62–65.
30. Garg, M. A novel text steganography technique based on html documents. *Int. J. Adv. Sci. Technol.* **2011**, *35*, pp. 129–138.
31. Mahato, S.; Yadav, D.K.; Khan, D.A. A modified approach to text steganography using HyperText markup language. In Proceedings of the 2013 Third International Conference on Advanced Computing and Communication Technologies (ACCT), Rohtak, India, 6–7 April 2013; IEEE: New York, NY, USA, 2013; pp. 40–44.
32. Fridrich, J. Applications of data hiding in digital images. In Proceedings of the Fifth International Symposium on Signal Processing and its Applications (IEEE Cat. No. 99EX359), ISSPA'99, Brisbane, Queensland, Australia, 22–25 August; IEEE: New York, NY, USA, 1999; Volume 1.
33. Wbstego. Available online: http://wbstego.wbailer.com/ (accessed on 2 December 2020).
34. Invisible Secret. Available online: http://www.invisiblesecrets.com/ (accessed on 2 December 2020).
35. Snow. Available online: http://www.darkside.com.au/snow/ (accessed on 2 December 2020).
36. Deogol. Available online: https://hord.ca/projects/deogol/ (accessed on 2 December 2020).
37. Cho, Y. Intelligent On-Off Web Defacement Attacks and Random Monitoring-Based Detection Algorithms. *Electronics* **2019**, *8*, 1338.
38. Flask Web Framework. Available online: https://flask.palletsprojects.com/en/1.1.x/ (accessed on 2 December 2020).
39. Python url.request Library. Available online: https://docs.python.org/3/library/urllib.request.html (accessed on 2 December 2020).
40. UUID_RFC4122. Available online: https://www.ietf.org/rfc/rfc4122.txt (accessed on 2 December 2020).

41. Alexa Top 500 Sites on the Web. Available online: https://www.alexa.com/topsites (accessed on 2 December 2020).
42. SimilarWeb Top Websites Ranking. Available online: https://www.similarweb.com/top-websites/ (accessed on 2 December 2020).
43. Tan, J. Optimal strategy selection approach to moving target defense based on Markov robust game. *Comput. Secur.* **2019**, *85*, 63–76.
44. Kanellopoulos, A.; Vamvoudakis, K.G. A moving target defense control framework for cyber-physical systems. *IEEE Trans. Autom. Control* **2019**, *65*, 1029–1043.
45. Yuk, S.; Cho, Y. A New Covert Communication Method based on Webpage Steganography. In Proceedings of the KIISE Korea Software Congress, Pyeongchang, South Korea, 18–20 December 2019; Volume 12, pp. 794–796.

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (http://creativecommons.org/licenses/by/4.0/).