Received 24 January 2023, accepted 28 February 2023, date of publication 9 March 2023, date of current version 15 March 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3254586

RESEARCH ARTICLE

Improving ANiTW Performance Using Bigrams Character Encoding and Identity-Based Signature

NABILA B. ROFIATUNNAJAH AND ARI M. BARMAWI, (Member, IEEE)
School of Computing, Telkom University, Bandung 40257, Indonesia

Corresponding author: Ari M. Barmawi (mbarmawi@melsa.net.id)

ABSTRACT Several threats, such as misinformation and hoax, exist in social media as a frequently used technology to exchange information. These threats can be prevented if the reader can verify the message's content integrity and source authenticity. One recent solution to overcome these threats was proposed by Ahvanooey et al. called ANiTW. ANiTW uses an invisible text watermarking scheme to verify social media message integrity and identify the message source. However, ANiTW is still vulnerable to undetected malicious text modification. Moreover, ANiTW has no tools to authenticate the extracted message source. The proposed method aims to overcome those vulnerabilities and improve ANiTW performance as a tool to verify message content integrity and message source authenticity in social media. To overcome vulnerabilities in ANITW, a part of the message hash and IBS-based signature is introduced as the watermark. The message integrity is verifiable by using part of the message hash as a watermark to detect any malicious. Moreover, implementing an IBS-based signature enables any reader to verify the message source's authenticity easily. In addition, to improve ANITW performance in embedding capacity and invisibility, a novel bigram-based character encoding and embedding rule are also introduced. Based on the experiment and analysis result, it is proven that the proposed method can perform better than ANiTW in the security, embedding capacity, and invisibility aspects.

INDEX TERMS Text watermarking, ID-based signature, source authentication, steganography, text integrity.

I. INTRODUCTION

Social media and messaging applications are among the most popular technologies today. Social media users continuously increase yearly, from over 4.26 billion in 2021 to almost a projected 6 billion in 2027 [1]. Data and information sharing via social media have also increased significantly following user growth, such that information security must be considered in social media usage [2]. Amidst the growing pace of information sharing, problems, threats, privacy, and security issues are growing as social media traffic increases. Social media faces several threats, such as fake information or hoax, scamming, sybil, malware, phishing, identity theft, and attacks from malicious users that can exploit the share and reshare functions to spread misinformation [3], [4], [5].

One common method malicious users use to perform these threats is to distribute modified text messages with false information on social media [6]. Distributed malicious text messages may contain false or exaggerated information to herd public opinion. A malicious user may also impersonate an official or a trusted source on social media by distributing fake official texts from a well-known or trusted source. A malicious user can create and distribute a believable fake text message by slightly modifying the official message from a trusted source. The modification may include a modification of vital information, such as phone numbers, emails, or links to bait and scam victims, into an impersonating malicious party. For example, a fake website for phishing can be created by simply modifying the extension from ".com" to ".net" of an official domain.

To prevent these kinds of threats, other users, such as the reader, need to verify two aspects of the received message: the source or author validity and the message content integrity. To ensure the message's source (author), the reader may check the validity of the original author via the in-app feature, if available. However, if the message is forwarded

The associate editor coordinating the review of this manuscript and approving it for publication was Zijian Zhang.

by copy-pasting between different social media applications, there is no tool to verify the message's source (author). Consequently, the reader does not know whether the author is legitimate only from the received text message. Moreover, since there is no modification trace or history on the received message, it is also difficult for the reader to verify the content's integrity.

One solution is to detect the modification and validate the text message's source using text watermarking. Text watermarking can embed data or information (e.g., the original author's name, creation time stamp, or related original information), which is required for the message integrity and source validation process into the message as cover text. A cryptographic function can also be used as a watermark in a text watermarking scheme with a higher level of security. The reader can verify text content integrity and validate the message's source by extracting the embedded data from the received watermarked message. Depending on the design, text watermarking can be used to prove the author's ownership, forgery detection, and text content authentication [2], [7], [8], [9].

Aside from its functionality to verify text integrity and source validity, several aspects need to be considered when applying text watermarking to social media. First, the maximum length of social media messages, depending on the application, is often limited. Following that, the maximum length of the watermark that can be embedded in the message is also restricted. This aspect affects how the watermarking scheme is designed, as the designed scheme is unusable if the length of generated watermark exceeds the social media maximum character count. The second is the applicability of the watermarking scheme across social media platforms. The symbol or method used to represent the watermark has to be supported in the platforms. The information embedded into the message must persist when the message is circulated. The watermark needed to be persistent when the message is forwarded via the forward function or manually reshared via copy-paste. Finally, the watermark used in social media should also be invisible without changing the original text context and purpose.

As one of the latest solutions to verify social media text integrity, Ahvanooey et al. introduced a robust text watermarking technique called ANiTW [10] in 2019. Ahvanooey et al. claimed ANITW as the state-of-the-art text watermarking technique to identify spurious information on social media using invisible signatures. This technique uses zero-width character (ZWC), an invisible character, to embed the author's name and the message's original number of words into the message. The use of ZWC in ANiTW provides a text watermarking scheme that is invisible and applicable to most social media platforms. By implementing ANiTW, the reader can identify the message source by extracting the author's name embedded in the received message. Furthermore, ANiTW also proposed a text integrity checker based on word-counting instances. ANiTW can verify the message content integrity by comparing the number of words calculated from the received message with the extracted number of words from the watermarked message.

However, on ANiTW, there is no tool for the reader to authenticate the validity of the extracted author's name, such that an impersonation attack can be implemented. Suppose that once the ZWC pattern is known, a knowledgeable attacker may imitate the pattern to impersonate another eligible author. There are also some vulnerabilities in the ANiTW text integrity checker caused by the ANiTW's dependencies on word-counting instances. Some modifications that do not change the message's number of words may go undetected, such as character insertion, deletion, and word reordering. As discussed, that kind of modification provides a sufficient opportunity for a malicious user to perform further threats on social media.

Furthermore, in ANiTW, four ZWCs are required to embed one character of the watermark in the message. Considering social media's limited character count, this requirement consumes a lot of character space in social media messages. The maximum embedding capacity of ANiTW is constrained by message length, as we can only use the remaining character space on social media. Thus, it is still challenging for ANiTW to embed other information to verify message content's integrity and source validity.

The main contributions of this research are as follows:

- **We are introducing a watermarking scheme for verifying a message's source authenticity.** The proposed method introduces a pairing-free IBS scheme as a watermark to verify the authenticity of the message source. The signature generated from IBS, verifiable by a signer public ID, is chosen to simplify the verification process. The verification process is conducted using the extracted ID, which acts as a public key in the IBS scheme. The reader can verify the ID as a message source by verifying the signature extracted from the watermarked message. The reader can then easily verify the source of the received message by using only the watermarked message. This capability of the proposed method does not yet exist in ANiTW.
- **We are Overcoming ANiTW vulnerabilities in detecting malicious message modification.** A message hash value as a watermark is introduced to overcome ANiTW vulnerabilities in detecting malicious message modifications. The message hash value is used in the proposed method to verify the watermarked message content integrity. We assume that whitespace and character capitalization modifications can be ignored for the proposed method. This assumption is made because the whitespace and character capitalization are often automatically modified while reformatting the message without changing the context. In the proposed method, the hash value extracted from the watermarked message is compared with the message's hash calculated by the receiver to verify the message's integrity. Thus, the proposed method is able to detect a modification over a watermarked text that goes undetected in ANiTW.

- **We are improving ANiTW performance in embedding capacity for increasing the watermark security level.** To increase the ANiTW embedding capacity, this research proposes a novel bigram character encoding to embed a watermark by mapping it to a combination of ZWC-cover text characters. The proposed method utilizes the existing message character to represent the watermark, such that the number of ZWC needed to represent the watermark is reduced. Using the proposed bigram mapping concept, the proposed method can use the remaining character space on social media more efficiently. Thus, we can attain a higher maximum embedding capacity by using the proposed mapping concept. By improving the embedding capacity, we can achieve a higher level of security because the length of the key used as a watermark on the scheme is also increased.
- **We are overcoming ANiTW vulnerabilities in invisibility and social media entity incorrectness.** A novel embedding location rule and position setting are introduced to increase ANiTW performance in terms of invisibility and usability. In the ANiTW method, a visible trace can be seen in social media special entities, such as hashtag, links, emoticon, and mention on the watermarked text. Moreover, the function of a watermarked entity in ANiTW differs from that of the original keywords, leaving them to work incorrectly (e.g., a watermarked mention in a social media message is linked to a different account). Meanwhile, in the proposed method, we analyze the existence of these entities as an unembeddable location to maintain their functionality. Thus, the proposed method can overcome ANiTW vulnerabilities in invisibility and maintain social media entity correctness.

The rest of the paper is organized as follows: Section II briefly explains previous methods and other related subjects used to improve ANiTW. In Section III, we give implementation details of the proposed method. Section IV shows experimental results and analysis of four watermarking aspects to evaluate ANiTW and the proposed method's performance. Finally, Section V presents our work's research findings and conclusion.

II. BACKGROUND

This section provides a brief description of the text watermarking scheme, previous method: ANiTW [10], Zero Width Character (ZWC), watermarking criteria analysis, Identity-Based Signature (IBS), and character bigram that is being utilized in this research.

A. TEXT WATERMARKING SCHEME

The text watermarking technique is divided into two types based on its approach, linguistic and structural [11]. The syntactic and semantic properties, such as grammar and synonyms, are utilized to embed the watermark for the linguistic-based approach. This technique modifies the cover text following the cover text language rule without changing its original meaning [11], [12]. A machine learning-based algorithm may also be implemented to achieve a natural watermarked text [13].

This technique is robust against watermarking schema attacks such as OCR and retyping. However, since the linguistic-based technique modifies the content of the original cover text, this technique is unusable for a sensitive document that cannot be tempered [11], [12], [13]. Moreover, a high computational cost in a machine learning-based algorithm is not applicable in a low computational environment such as social media.

The second approach, the structural or format-based technique, embeds the hidden watermark by modifying the feature or the structural properties of the cover text [11], [12]. Properties such as word spacing, line spacing, font formatting, and emoticons are used to represent the watermark without changing the original content of the cover text [12].

However, using those properties for this technique will leave a visual trace on the watermarked, making them distinguishable from the original cover text. Following that ZWC-based structural watermarking technique is proposed, allowing an invisible text watermarking scheme [10], [12], [14], [15], [16]. This technique is advantageous regarding invisibility, embedding capacity, and robustness against attack on structural watermarking technique [12]. Thus, following those properties, the ZWC-based structural watermarking technique is suitable for application in the social media platform. The ZWC-based scheme advantages are also the basis for why ANiTW is chosen as the main comparator for the proposed method.

B. PREVIOUS METHOD: ANiTW

Ahvanooey's method, ANiTW (A Novel Intelligent Text Watermarking) [10], is a watermarking technique for social media that uses a Zero-Width Character (ZWC) to embed the watermark. Ahnvanooey also proposes a word-count-based algorithm to check the integrity of received watermarked messages. Three main components are embedded into the cover text (CT) on the ANiTW method, the author's name, the number of words counted on CT, and the mark for each word on CT. Those three components and the mark of each CT's words are represented using ZWC.

The embedding process begins by converting each digit of the number of words and each character of the author's name as watermark text into a 6 bits binary string, as shown in Table 1. The binary string representation of the number of words is nw, and the binary string representation of the author's name is nameW. A complete binary string (HBS) is constructed by concatenating nW with a binary string "101111" as a component separator, and further, this binary string is concatenated with nameW.

Furthermore, a hidden watermark (Wh) is constructed by converting each 3 bits of HBS into two ZWCs based on

TABLE 1. Look-up table to convert watermark character into binary.

A-Z		a-z		0-9 and character	
Watermark Character	6-bits Binary Representation	Watermark Character	6-bits Binary Representation	Watermark Character	6-bits Binary Representation
A	110000	a	010000	0	001010
B	110001	b	010001	1	001011
C	110010	c	010010	2	001100
D	110011	d	010011	3	001101
E	110100	e	010100	4	001110
F	110101	f	010101	5	001111
G	110110	g	010110	6	101010
H	110111	h	010111	7	101011
I	111000	i	011000	8	101100
J	111001	j	011001	9	101101
K	111010	k	011010	!	101111
L	111011	l	011011	" "	101110
M	111100	m	011100		
N	111101	n	011101		
O	111110	o	011110		
P	111111	p	011111		
Q	000000	q	100000		
R	000001	r	100001		
S	000010	s	100010		
T	000011	t	100011		
U	000100	u	100100		
V	000101	v	100101		
W	000110	w	100110		
X	000111	x	100111		
Y	001000	y	101000		
Z	001001	z	101001		

TABLE 2. Look-up table to convert watermark bits binary ZWC representation.

Watermark 3-Bits	2 ZWC Representation
000	U+200E+"U+200E"
001	U+200C+"U+200E"
010	U+200E+"U+200C"
011	U+200C+"U+200C"
100	U+202C+"U+202C"
101	U+202C+"U+200E"
110	U+200E+"U+202C"
111	U+200C+"U+202C"

Table 2. Wh is inserted before every punctuation mark that ends a CT sentence, such as '.', '!', and '?'. In addition, a ZWC character (ZWC 202D) is also inserted before every last character of each word on CT. The processing is conducted to mark every original word in CT, which will later be used to analyze the watermarked CT content integrity.

Supposes CT is "Stay alert Impersonators are sending QR codes to steal your money. It is important to remember that no one from the government will ever tell you to pay with cryptocurrency. Learn how to avoid falling for QR code and other crypto scams: https://bit.ly/3LHfPHL" where the number of words in the sentence of CT is 42. The account's ID, as the equivalent for the author name, is "100064701336980". For this example, the HBS for nameW + "101111″ + nW has a total length of 108 bits. Thus, there are 42 202D ZWC and 72 Wh ZWC on the watermarked cover text.

The extraction method is straightforward: Wh is extracted from the watermarked CT and converted into a binary string based on Table 2. By detecting the location of separator "101111" in the watermark extracted from CT, the original nW and nameW can be extracted. The original value of the number of words and author names can be obtained by converting nW and nameW to original characters based on Table 1.

The number of words in the received watermarked CT (nW′) is counted manually. The text integrity rate (AC) of received watermarked CT is calculated using (1). The message is considered original if AC equals 1.

$$\mathrm{AC}=\begin{cases}\dfrac{\left(\begin{array}{c}\text{Number of ZWC}\\ \text{202D in Received CT}\end{array}\right)}{\mathrm{nW}}, & \text{if } \mathrm{nW} > \mathrm{nW}' \\[2ex] \dfrac{\left(\begin{array}{c}\text{Number of ZWC}\\ \text{202D in Received CT}\end{array}\right)}{\mathrm{nW}'}, & \text{otherwise}\end{cases} \tag{1}$$

As has been discussed before, text watermarking for social media messages is limited by the associated social media's maximum character count. With the length of the original CT as |CT|, the number of available embedding locations on social media ($\mathrm{EL}_{\mathrm{SocialMedia}}$) is calculated using (2).

$$\mathrm{EL}_{\mathrm{SocialMedia}} = \left(\begin{array}{c}\text{maximum character}\\ \text{count on social media}\end{array} - |\mathrm{CT}|\right) \tag{2}$$

On Ahvanooey's method, since four ZWCs represent each watermark character, the embedding capacity for each $\mathrm{EL}_{\mathrm{SocialMedia}}$ is equal to 1/4 character if the '.' or '?' or '!'

character exists on CT. The maximum embedding capacity for ANiTW can be calculated by (3).

$$\text{Max. EC}_{\text{ANiTW}} = \frac{\text{EL}_{\text{SocialMedia}}}{4} \tag{3}$$

C. ZERO WIDTH CHARACTER (ZWC)

Unicode is a standardized coding system that processes, encodes, and displays almost all modern digital text, including the one used in the social media application. Since 1987, the Unicode standard has been implemented in various internet protocols, programming languages, and operating systems. Moreover, Unicode is also applicable to multi languages [10], [14]. Zero Width Character (ZWC) is a special character in Unicode that can be used as a format or control character such as a Left-to-Right Mark (e.g., LRM makes words written from left to right in the middle of a right-to-left text in a certain language such as Arabic). ZWC has no width, length, or written symbol when deployed in a digital text that uses the Unicode system [10], [17]. ZWCs have been used as a hidden character to hide data into a cover text without leaving a trace [10], [14], [16], [18]. In Ahvanooey's method, 4 ZWCs are used to embed the watermark into the message, while for the proposed method, six ZWCs are employed to embed the watermark into the cover text as shown in Table 3.

TABLE 3. List of used ZWC for the ANiTW and proposed method.

ZWC Symbol	Codepoint
ZWC1	U+200C
ZWC2	U+200E
ZWC3	U+202A
ZWC4	U+202C
ZWC5	U+202D
ZWC6	U+2064

D. WATERMARKING CRITERIA ANALYSIS

There are four aspects of text watermarking criteria, embedding capacity, invisibility, robustness, and security.

1) EMBEDDING CAPACITY

Cryptographic functions such as hash can secure watermarking schemes against estimate-based attacks [19]. The purpose of the text integrity checker, proof of ownership, and temper proofing can also be achieved by implementing a suitable cryptographic function as the text watermark. For those cases where the watermark consists of a cryptographic function, a higher security level means a longer watermark is necessary. Hence, a watermarking technique is desired to embed as many watermark characters as possible into the cover text. Moreover, embedding capacity is one of the most important criteria to evaluate in the text watermarking scheme in social media with a limited text length.

Embedding capacity or payload is defined as the number of bits or characters that can be embedded into one embeddable location. Embedding capacity is calculated using bits-per-location (BPL) or character-per-location (CPL). Embeddable location (EL) refers to how many available positions in CT can be embedded with a watermark following the algorithm designed in the watermarking scheme. Maximum embedding capacity for CT is calculated based on (4) [12], [19].

$$\text{EC} = \begin{cases} \text{BPL} \times \text{EL} & \text{or} \\ \text{CPL} \times \text{EL} \end{cases} \tag{4}$$

2) INVISIBILITY

Aside from embedding capacity, the invisibility aspect is also important in the text watermarking criteria. Evaluating the invisibility aspect of watermarking technique refers to how many perceptual modifications are on watermarked CT [12], [19]. Even though the embedding capacity for one technique is high, it is undesirable if the cover text gets altered profoundly [16], [20], [21], [22], [23]. An invisible or undisguisable alteration on CT is also important to avoid suspicion and to protect the watermark against visual attack [18].

The best way to evaluate the invisibility aspect is by visually comparing the original and watermarked CT [7], [16], [20], [21], [23], [24]. Statistically, the similarity between original and watermarked CT can also be calculated using Jaro–Winkler Distance or Jaro-Similarity (d_{j}). Original and watermarked CT is similar with a d_{j} value of 1. Two characters between st_1 and st_2 can be identified as a match if the distance between the two characters is less or equal than $[\max(|\text{st}_1|, |\text{st}_2|)/2] - 1$ [12], [15], [25]. With m as the number of matching characters and t as half the number of characters transposition, d_{j} between two strings st_1 and st_2 is calculated using (5).

$$d_{\text{j}} = \begin{cases} 0, & \text{if } m = 0 \\ \frac{1}{3}\left(\frac{m}{|\text{st}_1|} + \frac{m}{|\text{st}_2|} + \frac{m-t}{m}\right), & \text{otherwise} \end{cases} \tag{5}$$

3) ROBUSTNESS

Intentional or unintentional changes may break or remove the watermark in watermarked CT circulated in public communication channels such as social media. Watermarking schemes can be fragile, semi-fragile, and robust, depending on the watermark's survivability against such attack [8].

Previous research mentions that the watermark's robustness can be calculated by evaluating the probability of the attacker founding the watermark character on CT such that an attack can be implemented. This probability can be calculated by the number of embedding locations (NL) and the length of CT. The higher the NL value is, the higher the chance that the watermark character can be found. The probability of founding the watermark character P(DR) can be calculated following (6). The higher the P(DR), the more robust watermarking scheme [12], [16], [19].

$$\text{P (DR)} = \frac{\text{NL}}{|\text{CT}|} \tag{6}$$

4) SECURITY

The trade-off between embedding capacity, invisibility, and robustness must be evaluated following the desired properties of each watermarking technique [10], [12], [19]. The security aspect of the watermarking technique as a text integrity verification tool can be evaluated by how well the algorithm detected changes in the watermarked CT.

Even though the hidden symbol used is invisible, an estimate-based attack and other analysis tools can still be used to reveal the existence of a watermark in watermarked CT. The attacker may remove the watermark or create an unauthorized embedding process using the pattern learned from watermarked text [26]. Furthermore, it is also important to evaluate the security aspect by calculating the possibility of obtaining the map of the watermark character [10].

E. IDENTITY-BASED SIGNATURE (IBS)

Identity-Based Signature (IBS) aims to create a signature on a message where any user can verify the signature using the signer's public information (such as email address, ID, or telephone number) instead of using the conventional public key. This method can simplify certificate management [27], [28], [29]. Since Shamir initiated an ID-based cryptosystem in 1984 [30], many IBS schemes have emerged. Pairing-based cryptography, an extension of elliptic curve cryptography (ECC), has been used in most IBS approaches [29], [31], [32], [33]. However, the high cost of pairing computation and long signature size can be a problem for the implementation process in low-bandwidth environments (e.g., cell phones where most users use social media) [34]. A pairing-free scheme has also been proposed as an alternative approach for generating a faster and shorter signature generation over an elliptic curve [34], [35]. Moreover, since the signature is a point element in an elliptic curve, we can shorten the signature size by performing point compression and decompression [36].

F. CHARACTER BIGRAM

N-gram is an n-adjacent sequenced element from a series. This series can include a sequence of letters, words, or syllables. The bigram of character is an n-gram of character sequence for an n = 2. For example, the word "Fraud" can be divided into a bigram of "Fr", "ra", "au", and "ud". The bigram frequency distribution analysis can be used in many fields of application, including natural language processing and cryptography [37], [38]. The proposed method uses the bigram frequency distribution of social media messages to create a novel mapping concept based on the bigram character and ZWC combination to represent the watermark.

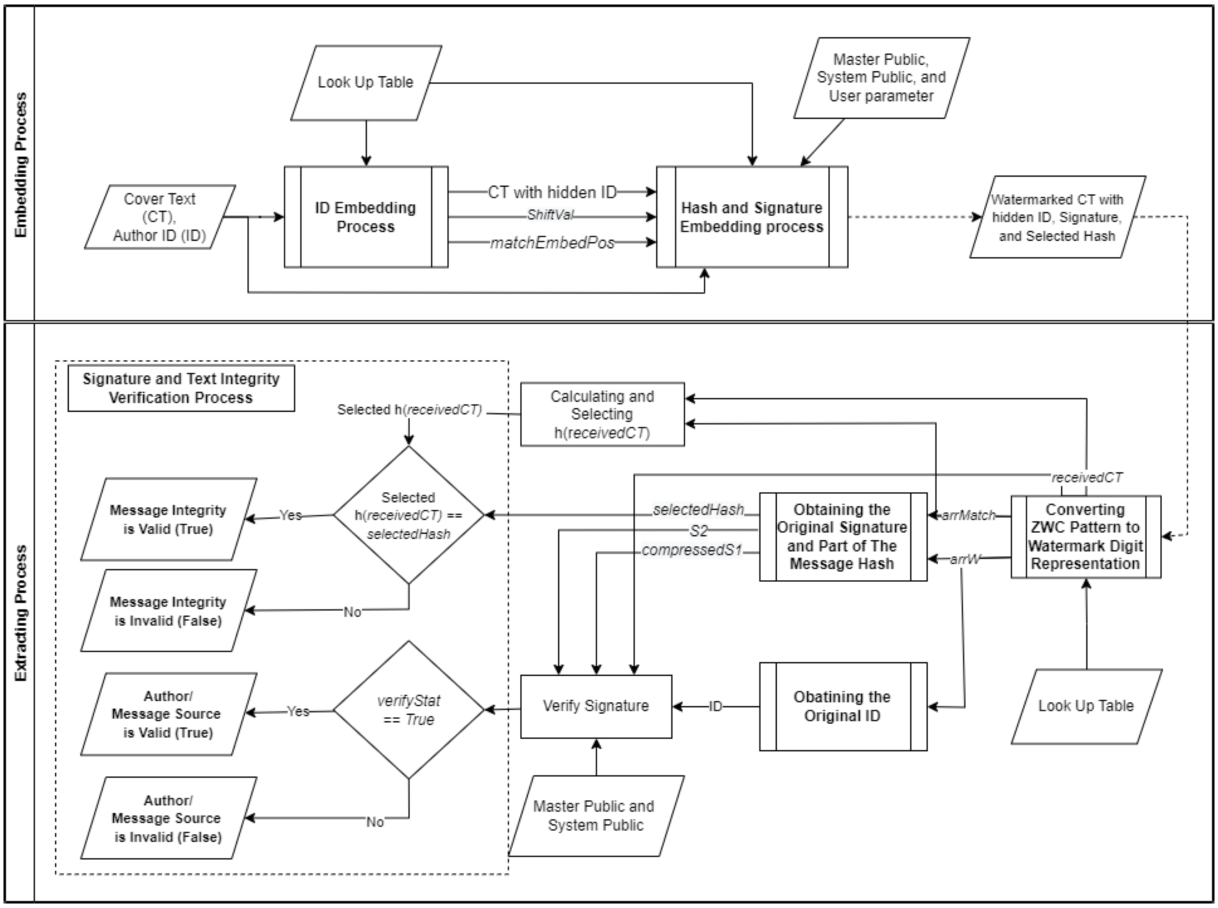
III. PROPOSED METHOD

Social media messages are often shared and forwarded without any information concerning the original message source (e.g., via copy-paste from one social media to another). Consequently, the reader cannot verify the message source or the content's integrity. One of the solutions is to embed necessary information into a message with a watermarking technique. The information embedded can include the message hash to verify the message content and the author's signature to validate the message source. However, the embedded signature must be verifiable without knowing the original sender. Moreover, the embedding capacity of social media messages also has to be maximized following its limited character count. Therefore, a text watermarking scheme using identity-based signature and bigrams character encoding is proposed. In the proposed method, part of the message hash and signature generated by the IBS scheme is introduced as a watermark to increase the security aspect of the watermarked message. A novel bigram-based character encoding is also proposed to increase embedding capacity and invisibility.

Supposes every message from a particular official or a trusted author is watermarked. The watermarked message may get shared and forwarded to the public via social media. Furthermore, there is an official public application or website where any reader can extract the watermark from the watermarked message. The reader may not gain any other information except the watermarked message itself. However, by implementing the proposed method scheme, every reader can check the author's authenticity and message integrity status using the extracted watermark elements without contacting or checking directly with the author.

The watermark used in the proposed method needs to provide enough information to verify the watermarked text content integrity and validate the authenticity of the message source (author). Three watermark elements are embedded into the social media message as the cover text (CT) to achieve that objective. The watermark elements are as follows:

- **ID Number of Author Social Media Account**.
  Every social media account can be identified by a unique ID accessible to the public. Following its unique and public properties, ID is chosen as one of the proposed method watermark elements to identify the message source or author. By getting the ID, any reader can retrieve the associated social media account. Moreover, the ID is also used as the public key in the IBS scheme.
- **Author Signature**.
  The signature is another watermark element used to validate the author's ownership and authenticate the message source. For the proposed method, a signature is generated using the IBS scheme. The IBS is chosen since the signature generated using this method can be verified using the embedded author ID. Thus, the signature verification is simplified.
- **Selected Part of the Message Hash**.
  Message digest or hash is commonly used to validate message integrity. The proposed method uses part of the message hash as a watermark element to authenticate message integrity. Any hash function can be used



Embedding Process
Look Up Table
Master Public, System Public, and User parameter
Cover Text (CT), Author ID (ID)
ID Embedding Process
CT with hidden ID
ShiftVal
matchEmbedPos
Hash and Signature Embedding process
Watermarked CT with hidden ID, Signature, and Selected Hash
Extracting Process
Signature and Text Integrity Verification Process
Selected h(receivedCT)
Calculating and Selecting h(receivedCT)
receivedCT
Message Integrity is Valid (True)
Yes
Selected h(receivedCT) == selectedHash
selectedHash
S2
compressedS1
Obtaining the Original Signature and Part of The Message Hash
arrMatch
arrW
Converting ZWC Pattern to Watermark Digit Representation
Message Integrity is Invalid (False)
No
Author/ Message Source is Valid (True)
verifyStat == True
Verify Signature
ID
Obatining the Original ID
Look Up Table
Author/ Message Source is Invalid (False)
Master Public and System Public

FIGURE 1. Overview of the proposed method.

to generate this third element. However, to prevent a false negative result in the text integrity checker, the hash value is desired to ignore some properties such as whitespace and case. These properties are necessary since unintentional changes may occur in text resharing between applications in social media messages.

The security level of the embedded watermark element depends on the number of embeddable bits in social media messages. To achieve a k-bits level of security, then it is required for the system needs to be able to embed a k-bits long watermark component. The maximum number that can be embedded into the social media message depends on the platform used to post the message. To achieve this objective, we proposed a novel bigram-based character encoding to embed the watermark with a combination of ZWC and CT characters. Each digit is embedded into the CT by searching for a possible character pair in the CT's character. Using the proposed method, we can embed the same length of watermark character with fewer ZWC than ANiTW Method.

The proposed method consists of three main processes: preliminary, embedding, and extracting. The preliminary process is conducted to create the experiment dataset and the look-up table used in the proposed method. The embedding process is conducted on the author's side to embed three watermark elements (ID, signature, and part of message hash) before the message is published online. The embedding process is divided into two subprocesses: the ID embedding process and the hash and signature embedding process. The output of the ID embedding process is used to determine the randomization factor in the hash and signature embedding process. In the extracting process, the reader extracts those elements to check the validity message source or author and content integrity. The overview of those processes is shown in Fig. 1.

Furthermore, the detail of the proposed method is discussed in three subsections, the preliminary process, the embedding process, and the extracting process.

A. PRELIMINARY PROCESS

This process is conducted once, before the embedding and the extracting process, to create a mapping table of bi-gram and prepare the dataset needed for experiments. This process is divided into three sub-processes: social media data scraping, message length distribution analysis, and message's bigram frequency distribution analysis.

1) SOCIAL MEDIA DATA SCRAPING

Social media data scraping is a process for collecting data from social media. Social media data scraping is necessary to achieve a realistic dataset and an efficient mapping in the look-up table used in the proposed method. For this research, we collected 753523 social media messages and author's ID from three social media platforms: Facebook, Instagram, and Twitter. In the data scraping process, the account on each social media is divided into three types, personal, business/cooperate, and official government accounts. The minimum number of accounts for each type is 30, with a minimum of 30 messages for each account. The messages are divided into two languages, Indonesian and English.

2) MESSAGE LENGTH DISTRIBUTION ANALYSIS OF SOCIAL MEDIA MESSAGE

This process aims to analyze and categorize the message collected from the social media data scraping result based on its length. This process is necessary since the performance of both Ahvanooey's and the proposed method on social media is depended on the cover text length. This process takes social media messages as input to analyze social media message length distribution. The result from this process is used to determine message length categories variation for the experiment. Based on the message length, the message will be grouped into three categories, short (message length $<=$ Q1), medium (Q1 $<$ message length $<=$ Q2), and long (Q3 $<$ message length).

3) MESSAGE BIGRAM FREQUENCY DISTRIBUTION ANALYSIS

The proposed mapping concept combines ZWC and CT characters into a digit value 0-9 to represent the watermark. Following that concept, frequency distribution analysis is conducted to achieve an optimal look-up table. This process aims to analyze the bigram frequency analysis from the message in the data scraping result. The unique character value of the first and second characters of CT's bigram is mapped with a ZWC to represent the watermark character. This mapping process is explained further in Section III-B1.d.

B. EMBEDDING PROCESS

The proposed method's watermark components consist of the author's social media ID, signature, and the selected part of the message hash. As shown in Fig. 1, the watermark component is embedded into the CT using two consecutive subprocesses: ID embedding and hash and signature embedding processes.

1) ID EMBEDDING PROCESS

As discussed in the previous section, ID is embedded as the author identifier and the public key for the IBS scheme. Before being embedded into the CT, ID is shifted to randomize the ZWC pattern on watermarked CT results. This process is necessary to prevent attackers from learning the embedding pattern when the message is repeatedly posted. The idea to prevent this attack is to create a unique embedding pattern for each posting time, even for similar messages and IDs. The ID embedding process is conducted as follows:

a: CHOOSING THE CHARACTER REFERENCE

The ID pattern is randomized based on the occurrences of randomly chosen character references (CharRef). A character reference index (indexCharRef) ranging from 0 to 4 is chosen randomly based on the look-up table shown in Table 4.

TABLE 4. Character reference for shifting the watermark.

Index	0	1	2	3	4
Character Reference	a	i	u	e	o

b: SHIFTING THE ID BASED ON THE CALCULATED SHIFTING VALUE.

The shifting value for ID (ShiftVal) is calculated by counting the occurrence of CharRef in CT. The shifted ID (ShiftedID) is generated by shifting(rotating) the ID's character ShiftVal-step to the left.

Suppose we choose a random number, 1, as our indexCharRef with 'i' as our CharRef with nine occurrences within the chosen cover text. With "100064701336980" as our ID and nine as our ShiftVal, the ShiftedID for the example is "33698010006470".

c: COMBINING ShiftedID AND indexCharRef

This step combines indexCharRef and ShiftedID to ensure that the extraction algorithm can shift back the ID. This step is conducted by concatenating indexCharRef at the end of ShiftedID. The combined ShiftedID||indexCharRef is embedded into CT as the watermarking input (W) for the ID embedding process.

d: EMBEDDING THE WATERMARK INPUT (W) INTO CT

The process of embedding the watermark input (W) is conducted to embed each watermark element into the CT. For the ID embedding process, W refers to the ID with the original message from the author's social media as the CT. Embedding the watermark input into the CT begins by converting each character of the W into a 2-digits decimal representation (W_Digits). The determined pattern between the W character and its 2-digits decimal representation is shown in Table 5. For example, the first character '3' from the ShiftedID "33698010006470" is converted into "13" as its decimal representation.

After each character in W is converted, W_Digits is embedded into the CT using the determined rule of embedding mode, embeddable location, and position setting. The details of each rule are explained as follows:

i. **Embedding mode.** Each digit of W_Digits is embedded into the CT using two possible modes. The first combines CT's character with a ZWC called Match Embedding Mode (MM). The second one uses only

TABLE 5. Look-up table to convert watermark character into digits decimal Representation.

Watermark Character	2 Digits Representation	Watermark Character	2 Digits Representation	Watermark Character	2 Digits Representation
A	48	W	06	s	34
B	49	X	07	t	35
C	50	Y	08	u	36
D	51	Z	09	v	37
E	52	a	16	w	38
F	53	b	17	x	39
G	54	c	18	y	40
H	55	d	19	z	41
I	56	e	20	0	10
J	57	f	21	1	11
K	58	g	22	2	12
L	59	h	23	3	13
M	60	i	24	4	14
N	61	j	25	5	15
O	62	k	26	6	42
P	63	l	27	7	43
Q	00	m	28	8	44
R	01	n	29	9	45
S	02	o	30	!	47
T	03	p	31	" "	46
U	04	q	32		
V	05	r	33		

TABLE 6. Mapping type for match embedding mode using the combination of ZWC and CT character.

Embedding Method Name	Embedding Example	Embedded Value	CT Char. and ZWC Combination for MM
BMM	CT[i-1] +ZWC1+CT[i]	-1	ZWC1+CT[i]
AMM	CT[i-1] +ZWC2+CT[i]	2	CT[i-1]+ZWC2
BAMM	CT[i-1] +ZWC3+CT[i]	1	CT[i-1]+ZWC2 and ZWC1+CT[i]

ZWC, called Non-Match Embedding Mode (NM), to represent the digit.

For the MM, every digit in W_Digits is represented by a combination of one CT character and one ZWC. The ZWC can be combined with the character before, after, or before and after the ZWC. As shown in Table 6, MM is divided into three types based on which characters are used in MM. These types can be distinguished by the ZWC type placed between the characters. The first type of MM with ZWC1 is called Before-ZWC Match Embedding Mode or BMM, the second one with ZWC2 as After-ZWC Match Embedding Mode or AMM, and the third one with ZWC3 as Before-ZWC and After-ZWC Match Embedding Mode or BAMM.

Suppose the received watermarked CT consists of 'A' + ZWC + 'N'. If the ZWC type being embedded is ZWC1, then we can take ZWC1 + 'N' as the representation of the digit in W_Digits. If ZWC2 is used, we can take 'A' + ZWC2 as the watermark representation. Lastly, if the ZWC type is ZWC3, we can take both ZWC1 + 'N' and 'A' + ZWC2 for the extraction process.

TABLE 7. Look-up table for matching embedding mode between CT character and ZWC.

Watermark Digits Representation	Possible Character Pair for AMM			Possible Character Pair for BMM		
0	'a'	'j'	'o'	'n'	'u'	'c'
1	'n'	'h'	'.'	' '	'o'	'0'
2	'i'	','	'w'	'g'	'y'	'2'
3	' '	'u'	'^'	'd'	'j'	','
4	'e'	'b'	'c'	'r'	'h'	'w'
5	'k'	'p'	'2'	'a'	'l'	'('
6	't'	'g'	'f'	'p'	't'	'x'
7	'd'	'y'	'q'	'm'	'e'	'v'
8	'l'	's'	'v'	's'	'b'	'f'
9	'm'	'r'	'x'	'i'	'k'	'q'

The CT Character and ZWC pair are being determined based on the data scraping result to achieve a higher possibility of finding a suitable MM, especially BAMM. The bigram of collected messages from data scraping is ranked based on its occurrence frequency. BAMM pattern can be generated if the CT character used for AMM and BMM is adjacent, where the first character from the bigram is mapped for AMM and the second character is mapped for BMM.

Based on the number of available character pairs discussed in Section III-A, there are three possible character sets for each digit of the watermark representation. The mapping between each character in the set to the digits is randomized. The mapping between a digit and its possible character pair is shown in Table 7.

As for the NM, every digit in W_Digits of *W* is converted into a ZWC representation based on the non-match look-up table as shown in Table 8. As seen on the look-up table, each digit in W_Digits is represented using one to two ZWCs. Two to four ZWCs are then needed to represent one watermark character in NM.

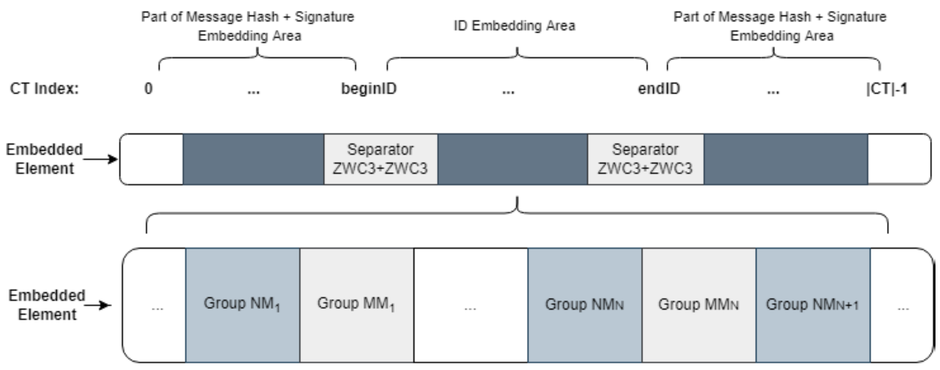


FIGURE 2. The embedding position setting.

TABLE 8. ZWC representation for watermark digits decimal representation in non-match embedding mode.

Watermark Digits Representation	2 ZWC Representation
0	ZWC4
1	ZWC5
2	ZWC6
3	ZWC4+ZWC1
4	ZWC4+ZWC2
5	ZWC4+ZWC3
6	ZWC5+ZWC1
7	ZWC5+ZWC2
8	ZWC5+ZWC3
9	ZWC6+ZWC1
*	ZWC6+ZWC2
Separator	ZWC3+ZWC3

For example, there are two possible ways to embed the first two digits "13" from the W_Digits. The first is by separately representing each digit using either AMM or BMM. The second one uses one ZWC to represent both digits using BAMM. For the digit '1' AMM pattern, we can use either "n + ZWC2" or "h + ZWC2" or ". + ZWC2". The possible BMM patterns for '1' are "ZWC1 + [space]" or "ZWC1 + o" or "ZWC1 + 0". The second digit, '3', can be represented using either "[space] + ZWC2" or "u + ZWC2" or "^ + ZWC2" following the AMM pattern. While for the possible BMM pattern, we can use "ZWC1 + d" or "ZWC1 + j" or "ZWC1 +," to represent the digit '3'.

Suppose there is a CT "Stay alert Impersonators are sending QR codes to steal your money", and the ID is embedded by consequentially searching for a possible match in CT. The digit '1' is embedded using the combination of ZWC1 or "U+200C" + the first space character on CT. Meanwhile, the digit '3' is embedded using the combination of the second space character + ZWC2 or "U+200E" on CT. By using the ZWC mentioned in Table 3, the watermarked CT for this example is "Stay[U+200C] alert [U+200E]Impersonators are sending QR codes to steal your money".

However, if we changed the CT into "Fend off fraudsters by taking this safety measure!" we can use BAMM to represent the '13' with one ZWC. This condition occurs since the position of "n + ZWC2" for 1's AMM pattern is adjacent to "ZWC1 + d" for 3's BMM pattern in the word "Fend". Consequently, following the BAMM pattern, we can use only one ZWC3 or "U+202A" placed between 'n' and 'd' to represent both digits. Therefore, the watermarked CT for the BAMM mode of "13" is "Fen[U+202A]d off fraudsters by taking this safety measure!".

Following those examples, it is clear that combining CT's character and ZWC can reduce the number of ZWC needed to embed the watermark. Contrary to NM, which needs at least three ZWCs, using the AMM or BMM, only two ZWCs are needed to embed one watermark character. Furthermore, only one ZWC is needed to represent one watermark character if the CT condition is fitted for the BAMM pattern.

ii. **The rule for embeddable location**. As mentioned, ZWC is used as a hidden symbol for both embedding modes to represent the watermark. ZWC can be placed anywhere on text messages. However, on social media, some special attributes such as emoticons, hashtags, and links behave differently from regular messages. ZWC in social media is considered or acts as an invisible space, such as placing ZWC on those attributes will affect the functionality of those entities and the invisibility aspect of the embedded watermark. ZWC is also getting cut off if placed at the end of the line on CT. To ensure the watermark is still invisible and to prevent against watermark's accidental data loss on social media messages, there are several rules as follows:

- No ZWC is placed at the beginning or the end of the message.
- No ZWC is placed before, in between, and after an emoji.
- No ZWC is placed in-between social media special attributes such as hashtags, links, and mentions.
- No ZWC is placed before, in-between, and after '\n' or end of line character.

iii. **Position setting.** NM and MM used to represent all watermark elements are placed in a specific pattern on CT based on the position setting shown in Fig. 2. Suppose the CT[k] is used to embed W_Digits[j] using MM, then the NM pattern used to represent W_Digits[i : j − 1] is placed at the CT[k − 1]. The embedding area for the ID embedding process and the signature and selected hash embedding process are also regulated to achieve an efficient embedding process. For this research, since the signature and hash are generally longer, we limit the ID embedding to only a third of the length of CT. The signature and selected hash can be embedded before the first and after the last position of the ID embedding location on CT (marked by separator' ZWC3ZWC3').

Following the embedding mode, the rule of embeddable location, and the position setting discussed above, the embedding process is conducted using steps shown in Algorithm 1. MM is prioritized on the embedding algorithm to achieve a maximum embedding capacity. Following that, the W_Digits embedding process is started by trying to find a matching CT's character that is suitable to embed the digit using MM.

Searching for a suitable location or a matching CT's Char (matchChar) is conducted for each digit in W_Digits within all CT's characters. If no matchChar is found, then the NM representation of the digit based on Table 8 is collected as a group in nonMatchZWC. Meanwhile, if matchChar is found, an embed value depending on the digit's MM type based on Table 6 is added into an array (arrayCT) as a tracker to find a possible BAMM for the next digit. Furthermore, the current group of nonMatchZWC is embedded into the CT before the matchChar position.

After attempting to embed each digit using MM and all groups of nonMatchZWC are already embedded into CT, we evaluate arrayCT for a BAMM possibility. For every MM position in CT, ZWC is embedded into the CT following the arrayCT value based on Table 6.

Finally, a separator character is embedded into CT before the first and after the last embedding position. This embedding position includes either MM or NM position on watermarked CT. This step is only conducted for the ID embedding process to separate the ID from other watermark elements.

For the ID embedding process, the outputs of this algorithm are ID-watermarked CT and the collection of embedding positions using MM (matchEmbedPos), which are used for the next hash and signature embedding process.

Algorithm 1 Embedding the Watermark Input (W) Into CT

```
Function Embed_W_into_CT:
    Input  : Cover Text (CT), watermark input (W)
    Output: Watermarked CT (CTW), all index of CT
            characters used for MM (matchEmbedPos)
    arrayCT ← [0] × length of CT
    CTW ← CT
    foreach char ∈ W do
        W_Digits ← W_Digits + 2-digits decimal
        representation of char based on Table 5
    end foreach
    foreach digit ∈ W_Digits do
        matchChar ← Find an available character in CT
        that matched for digit's MM based on Table 7
        following the embedding rule, position setting,
        and the rule for embedding location as written in
        Section III-B1.d on point i, ii, and iii.
        indexMatch ← index of matchChar in CT
        MMType ← BMM or AMM
        if matchChar is not found then
            nonMatchZWC ← nonMatchZWC + ZWC
            representation of digit using NM based on
            Table 8
        else
            if MMType is BMM then
                indexMatch ← indexMatch-1
            end if
            arrayCT[indexMatch] ←
            arrayCT[indexMatch] + embed value of
            matchChar based on Table 6
            CTW[indexMatch – 1] ←
            CTW[indexMatch – 1] + nonMatchZWC
            nonMatchZWC ← ''
            matchEmbedPos.append(index of matching
            CT)
        end if
    end foreach
    if nonMatchZWC is not empty then
        CTW[indexMatch] ← CTW[indexMatch] +
        nonMatchZWC
    end if
    foreach matchPos ∈ matchEmbedPos do
        zwcMatch ← ZWC representation of
        arrayCT[matchPos] based on Table 6
        CTW[matchPos] ← CTW[matchPos] +
        zwcMatch
    end foreach
    Embed separator (ZWC3+ZWC) before the first and
    last embedding position on CTW (For the ID
    embedding process only)
    return CTW, matchEmbedPos
End
```

2) HASH AND SIGNATURE EMBEDDING PROCESS

The hash and signature embedding process aims to embed part of the message hash and signature into the ID-watermarked CT. The part of the message hash and signature is embedded outside the embedding location on CT. This location refers to before the first and after the last position of the separator character. This process consists of steps as follows:

TABLE 9. Example for embedding process in proposed method.

Component	Values
Cover Text (CT)	Stay alert Impersonators are sending QR codes to steal your money. It's important to remember that no one from the government will ever tell you to pay with cryptocurrency. Learn how to avoid falling for QR code and other crypto scams: https://bit.ly/3LHfPHL
Author Social Media ID	100064701336980
CharRef	i, indexCharRef = 1
ShiftVal	9
ShiftedID	33698010006470
W_Digits of ShiftedID	13134245441011...0421443101111
matchEmbedPos	[3, 10, 16, 39, 40, 54, 63, 74, 84]
Signature	115400916233902548194283578858385448324535826130735354297003145248302*196277...943216
selectedHash	162710959
W_Digits	11111511012641145421243431442114212144 4454...42431313454542151413441512151442424414121
ZWC on CT	Sta[202A][202A]y[200c] alert [200E]Impers[200c]onators are sending QR [200E]c[200E]odes to steal [200c]your mone[200E]y. It's imp[200E]ortant to [200c]r[202c][200E][202d][202c] ... [202A]ember that no one from the government will ever tell you to pay with cryptocurrency. Learn how to avoid falling for QR code and other crypto scams: https://bit.ly/3LHfPHL

a: CALCULATING AND SELECTING HASH

This step begins by cleaning the CT by lowering all the characters and choosing only alphabet characters and digits from the original CT. The cleaned CT (cleanCT) is used to calculate the hash value of the social media message. The hash value is selected by taking the first N digits from the hash calculation result with N equal to the length of matchEmbedPos from the ID embedding process.

b: GENERATING SIGNATURE

Considering the signature length and the computational cost, we choose to implement a modified scheme of Yaduvanshi and Mishra [34] and Mishra's [35]. The scheme is divided into four stages, setup, user key extraction, signature generation, and signature verification. On the proposed method, the setup and key extraction scheme proposed by Yaduvanshi and Mishra [34] is implemented with hash functions proposed by Mishra et al. [35]. Two hash function used on this process are $H_1 : (0, 1)^* \to Z_q*$ and $H_2 : (0, 1)^* \times (0, 1)^* \times G \times G \to Z_q*$. Assume a valid setup by a trusted CA and a key generation stage by a valid author are already performed before the embedding process. The results of the setup process are system master key (s) and system public parameter (G, q, P, Pub, $H_1()$, and $H_2()$). Meanwhile, the results of the key generation process are the user's public parameters (Z_{ID}) and the user's secret parameter (d_{ID}). The signature generation process is conducted as follows:

- Choosing $r_{\text{mID}} \in Z_q*$ randomly, where r_{mID} is a random number corresponding to m and ID using matchEmbedPos and ShiftVal from the ID embedding process.
- Calculating $S_1 = r_{\text{mID}} Z_{ID} \in G$, where G is an additive cyclic group of prime order q, generated by point P. G is defined over a non-singular elliptic curve over a field of q ($E(F_q)$) that satisfies $y^2 = x^3 + ax + b$ with $4a^3 + 27b^2 \neq 0$.
- Calculating $h = H_2(\text{cleanCT}||\text{ID}||Z_{\text{ID}}||\text{Pub}) \in Z_q*$ and $S_2 = \left[(r_{\text{mID}} + h)d_{\text{ID}}\right] \in Z_q*$
- Compressing S_1 by selecting only x-ordinate from the S_1 point coordinate.
- Appending '0' at the beginning of the compressed S_1 (compressedS1) if y-ordinate is even or '1' if y-ordinate is odd.

c: COMBINING SIGNATURE AND SELECTED PART OF MESSAGE HASH

The compressed S_1 (compressedS1), separator character '*', S_2, and the selected part of the message hash (selectedHash) is combined for the second watermark input. The watermark input for this process is $W = \text{compressedS1}||`*'||\text{S}_2||\text{selectedHash}$.

d: EMBEDDING THE WATERMARK INPUT (W) INTO CT

Watermark input (W) is embedded into watermarked CT generated from the ID embedding process, as discussed in Section III-B1.d.

An example of the proposed method embedding process can be seen in Table 9. Based on the watermarked CT in the example, an example of position setting is shown in Fig. 3. An example of the final result of watermarked CT in social media (Facebook) is shown in Fig. 4.

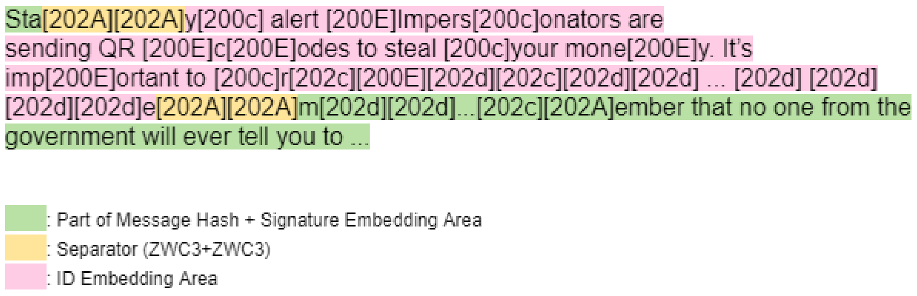
3) CALCULATING MAXIMUM EMBEDDING CAPACITY

As has been discussed in Section II-D1, in the proposed method, the payload for each $\text{EL}_{\text{SocialMedia}}$ will differ based on the embedding mode used to embed the watermark character. Suppose nCEaB is the number of CT characters that can be embedded with ZWC before and after the character. Furthermore, the number of CT characters that can be embedded with ZWC only before or after the character is nCEoB. The total value of nCEaB is used as the embedding location for BAMM EL_{BAMM} as shown in (7).

$$\text{EL}_{\text{BAMM}} = \max\left(\text{EL}_{\text{SocialMedia}},\ \text{nCEaB}\right) \tag{7}$$

The total value of nCEoB within unused space in the social media is used as a location for AMM or BMM $\text{EL}_{\text{AMMorBMM}}$ as shown in (8).

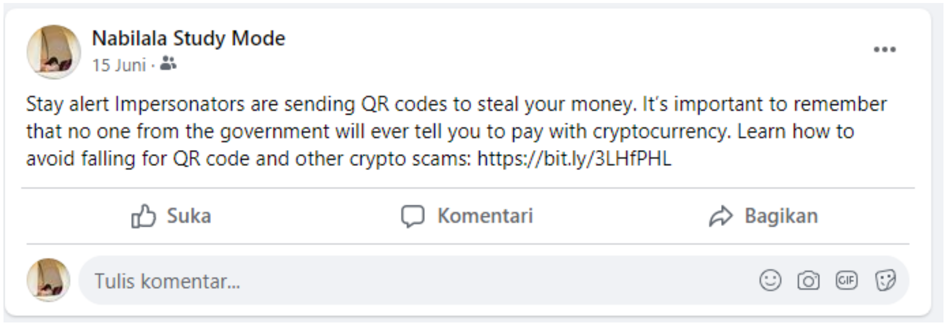
$$\text{EL}_{\text{AMMorBMM}} = \max(\text{EL}_{\text{SocialMedia}} - \text{EL}_{\text{BAMM}},\ \text{nCEoB}) \tag{8}$$



Sta[202A][202A]y[200c] alert [200E]Impers[200c]onators are sending QR [200E]c[200E]odes to steal [200c]your mone[200E]y. It's imp[200E]ortant to [200c]r[202c][200E][202d][202c][202d][202d] ... [202d] [202d] [202d][202d]e[202A][202A]m[202d][202d]...[202c][202A]ember that no one from the government will ever tell you to ...

🟩 : Part of Message Hash + Signature Embedding Area
🟨 : Separator (ZWC3+ZWC3)
🟥 : ID Embedding Area

FIGURE 3. Example of position setting implementation in watermarked CT.



Nabilala Study Mode
15 Juni ·

Stay alert Impersonators are sending QR codes to steal your money. It's important to remember that no one from the government will ever tell you to pay with cryptocurrency. Learn how to avoid falling for QR code and other crypto scams: https://bit.ly/3LHfPHL

FIGURE 4. Example of watermarked text posted on facebook. the result of the proposed method is invisible on social media.

The leftover character count of social media is used as an embedding location for NM mode (EL_{NM}) as shown in (9).

$$EL_{NM} = EL_{SocialMedia} - (EL_{AMMorBMM} + EL_{BAMM}) \tag{9}$$

As has been discussed in Section III-B1.d, the embedding capacity for each EL_{BAMM} is equal to 1 character. Meanwhile, the embedding capacity for each $EL_{AMMorBMM}$ is 1/2 and for each EL_{NM}) is 1/4 character. Following that, for the worst-case scenario, the maximum embedding capacity of the proposed method is equal to ANITW. As for the best-case scenario, the maximum embedding capacity of the proposed method can be calculated using (10).

$$\text{Max. } EC_{Proposed} = (EL_{BAMM}) + \left(\frac{EL_{AMMorBMM}}{2}\right) + \left(\frac{EL_{NM}}{4}\right) \tag{10}$$

C. EXTRACTING PROCESS

All watermark elements are extracted and used to verify the message's source authentication and content integrity in the extraction process. As shown in Fig. 1, the extracting process consists of four subprocesses, converting the ZWC pattern to watermark digit representation, obtaining the original signature and part of the message hash, obtaining the original ID, and signature and text integrity verification process.

TABLE 10. Lookup table to decode CT Bigram.

		CT[i+1]						
		ZWC 1	ZWC 2	ZWC 3	ZWC 4	ZWC 5	ZWC 6	Other Char.
CT[i]	ZWC1	x	x	x	x	x	x	BMM
	ZWC2	x	x	x	x	x	x	AMM
	ZWC3	x	x	Sep.	x	x	x	BAMM
	ZWC4	'3'	'4'	'5'	'0'	'0'	'0'	'0'
	ZWC5	'6'	'7'	'8'	'1'	'1'	'1'	'1'
	ZWC6	'9'	'*'	'3'	'2'	'2'	'2'	'2'
	Other Char.	x	x	x	x	x	x	x

1) CONVERTING ZWC PATTERN TO WATERMARK DIGIT REPRESENTATION

This process aimed to rebuild the original CT, collect all parts of the W_Digits, and store the number of MM found on watermarked CT. This process analyses each bigram in the watermarked CT (CTW). First, the original CT can be retrieved by evaluating whether bigram's first character is a ZWC. If the first character in the current bigram is not a ZWC, then the first character is collected to rebuild the original CT.

Second, we retrieve each W_Digits digit by evaluating a possible embedding pattern on each bigram. This evaluation is conducted by analyzing bigram's decode value. The bigram's decode value (decodeValue) is determined by the bigram's first and second character combination as shown in Table 10.

If decodeValue is 'X', that means the current bigram is not included in our embedding pattern such that the current

Algorithm 2 Converting ZWC Pattern to Watermark Digit Representation

```
Function Extracting_the_Watermark_Character:
    Input  : Watermarked CT (CTW)
    Output: received CT without watermark
            (receivedCT), an array of W_Digits in
            CTW(arrW), the total MM pattern found in
            CTW for all W_Digits (arrMatch)

    foreach bigram ∈ CTW do
        if the first character ∈ bigram is not a ZWC then
            receivedCT ← receivedCT + first character
        endif

        decodeValue ← decode value of bigram based
        on Table 10.
        if decodeValue is 'X' then
            continue
        else if decodeValue is a decimal or '*' then
            receivedW.append(decodeValue)
        else if decodeValue is BMM/AMM/BAMM then
            digitValue ← watermark digit representation
            of bigram based on Table 7
            receivedW.append(digitValue)
            matchCount ← matchCount +1
        else if decodeValue is 'Sep.' then
            arrW.append(receivedW)
            arrMatch.append(matchCount)
            receivedW ← ''
            matchCount ← 0
    end foreach
    return receivedCT, arrW, arrMatch
End
```

bigram is ignored. If the decodeValue is a decimal '0'..'9' or '*', then the digit value is equal to the decodeValue. If the decodeValue is 'BMM' or 'AMM' or 'BAMM', the digit value can be retrieved by looking at the look-up table. The digit value is grouped into one temporary variable called receivedW. In addition, the number of decode values with 'BMM', 'AMM', or 'BAMM' is also collected to keep track of MM found in CT.

The W_Digits are grouped into three parts based on two separator characters in watermarked CT. The first part of W_Digits is placed before the first separator character. The second part of W_Digits is placed between two separator characters. The third part of W_Digits is placed after the second separator character. Following that, if the bigram decodeValue is 'sep.' which marked a separator, the existing W_Digits is grouped by appending receivedW into an array arrW and matchCount into an array arrMatch.

This detail of this process is shown in Algorithm 2. There are three results for this process, received CT without watermark (receivedCT), an array of W_Digits parts in CTW (arrW), and the collection of the number of MM patterns found from each W_Digits part in watermarked CT (arrMatch).

2) OBTAINING THE ORIGINAL SIGNATURE AND PART OF THE MESSAGE HASH

This process aims to extract and separate the original signature and part of the message hash character from an array of W_Digits. The input of this process is arrW and arrMatch extracted from III-C1. The outputs of this process are selectedHash, compressedS1, and S_2 as extracted hash and signature value. This process consists of the following steps:

- Getting the W_Digits of signature and part of the message hash by combining the first and second elements of arrW with W_Digits = arrW[0] + arrW[2].
- Converting each two digits of W_Digits into one watermark character based on Table 5.
- Separating the signature (compressedS1 and S_2) and selectedHash by taking the Watermark character's N-last character as selectedHash, with N equal to the length of arrMatch[1].
- Separating the compressedS1 and S_2 by using the character '*' position as a reference.

3) OBTAINING THE ORIGINAL ID

This process aims to extract the original author's ID from an array of W_Digits. The input of this process is arrW and arrMatch extracted from III-C1. The outputs of this process are ID as the original author identifier. This process consists of the following steps:

- Getting the W_Digits of Signature and part of the message hash by combining the first and second elements of arrW with W_Digits = arrW[1]
- Converting every two digits of W_Digits into one watermark character based on Table 5.
- Separating ShiftedID and indexCharRef by taking the last character from the watermarks as indexCharRef.
- Shifting the ShiftedID to the right using ShiftVal. ShiftVal is calculated using the steps mentioned in Section III-B1.

4) SIGNATURE AND TEXT INTEGRITY VERIFICATION PROCESS

The watermark element extracted from the previous process is evaluated to prevent information manipulation by a malicious attack such as a tampering attack. Two aspects can be evaluated on the watermarked CT: text integrity and the author or source integrity.

The text content integrity is evaluated by comparing the calculated and extracted part of the message hash from the watermarked CT. The hash value of receivedCT is calculated and selected using the same step mentioned in III-B2 with matchEmbedPos = arrMatch[1]. The message integrity status (integrityStat) is considered valid (TRUE), or the message content is original if the comparison results between a part of the message hash from receivedCT and selectedHash are equal.

The received message source can be checked by using the extracted ID. Moreover, the reader can check the author or user's validity by verifying the signature component. Using cleaned receivedCT (see Section III-B2.a), receivedID, compressedS1, S_2, and other parameters as has been discussed in Section III-B2, the signature verification process is conducted as follows:

- Recalculating S_1 by decompressing compressedS1 by recalculating y-ordinate from x-ordinate value by considering the '0' or '1' mark at the start of compressedS1.
- Calculate $h = H_2(\text{cleaned receivedCT}||\text{receivedID}||Z_{\text{ID}}||\text{Pub}) \in Z_q*$
- The signature verification phase refers to Yaduvanshi's method [34]. However, the variable x_{ID} is replaced by r_{mID}. The replacement is conducted because the proposed method uses different randomization factors and hash functions, as discussed in Section III-B2.b. Thus the signature verification is done by comparing S_2P and $S_1 + hZ_{\text{ID}}$. If the comparison holds, then the signature is valid.

Based on the result of the hash comparison and signature verification process, there are three possible evaluation results:

- The content is valid, and the author is legitimate if the message integrity and the signature verification result are TRUE.
- The author or message's source is not legitimate if the signature verification result is FALSE.
- The message has been modified if the message integrity is FALSE.

IV. EXPERIMENT RESULT AND DISCUSSION

This section discusses the experiment scenario, result, analysis, and discussion of four watermarking criteria: invisibility, embedding capacity, robustness, and security.

A. EXPERIMENT SCENARIO

The proposed method applies to any Latin-based platform that implements the Unicode system. However, to analyze how both methods perform in a real-life application, an experiment is conducted on Facebook and Instagram as one of the most frequently used social media platforms [39]. The cover text and ID data are taken from the social media data scraping. For the experiment, we used 180 social media covers (Instagram's caption and Facebook's post) as the input CT with its associated ID. Based on the message length distribution, the CT is divided into three categories, short, medium, and long. This categorization is necessary since the embedding capacity in social media will depend on the cover text length. The experiment evaluates ANiTW and the proposed method's performance in four watermarking criteria: embedding capacity, invisibility, robustness, and security.

The maximum embedding capacity of ANITW and the proposed method is calculated based on (4), as mentioned in Section II-D1. Two scenarios are used to evaluate the maximum embedding capacity. The first scenario is taken from the original calculation used in ANITW, assuming an optimal embedding condition. The available embedding location is calculated solely based on the maximum character count on social media (see (2) in Section II-B). All of CT's character is assumed to be a suitable embedding location for ANiTW and the proposed method. On the proposed method, all available embedding locations are assumed to be suitable for BAMM (see (7) in Section III-B3). For this scenario, the embedding capacity is calculated when the cover text used 25%, 50%, and 75% of the social media maximum character count. However, in a real-life implementation, the maximum embedding capacity in social media depends on the details of the cover text. The maximum embedding capacity is influenced by the message length, character combination, punctuation marks, and whitespaces. Following that, for the second scenario, the embedding location rule of both methods is implemented within the dataset of real-life social media messages. The rule will determine the number of available embedding locations for each message. The embeddable location for ANiTW depends on the existence of a punctuation mark, as mentioned in Section II-B. While the embeddable location for the proposed method depends on the message character occurrence as mentioned in Section III-B1.d.

The invisibility aspect of the ANiTW and the proposed method performance is compared statistically using the Jaro-Similarity calculation following (5) and by visual comparison between the watermarked CT with the original CT posted on social media as discussed in Section II-D2. The evaluation is conducted since there are two possible ways for the reader or an attacker to obtain the watermarked text, by seeing it visually (e.g., directly seen on social media message posted on a digital device) and as a string data (e.g., obtain the data by scraping process or when the message is being transmitted). Both methods should be visually and statistically invisible to avoid suspicion. For calculating Jaro-Similarity, following (5) in Section II-D2, the original CT is used as the input st_1, and the watermarked CT is used as st_2. For the second evaluation, both methods' original and embedded cover text is posted on the associated social media in the same order. The invisibility aspect of the uploaded embedded cover text is evaluated visually using the original cover message as the reference. The embedded cover text for ANiTW and the proposed method is considered invisible if there is no visual difference between the watermarked text and the original cover text.

The robustness of the ANiTW and the proposed method performance is compared statistically using (6) mentioned in Section II-D3 following the original evaluation used in ANiTW. In addition, the robustness of both methods is also evaluated using a simulation-based experiment to see how watermark performs under deletion attack (e.g., from an

TABLE 11. Maximum embedding capacity comparison between ANiTW and the proposed method in social media.

Social Media	Maximum Char Capacity (UTF-8 Char)	Maximum Embedding Capacity					
		CT Length is 25% Social Media's Max. Char. Capacity		*CT Length is 50% Social Media's Max. Char. Capacity*		*CT Length is 75% Social Media's Max. Char. Capacity*	
		ANiTW	*Proposed*	*ANiTW*	*Proposed*	*ANiTW*	*Proposed*
Facebook	31603	5925	23701	3950	15800	1975	7899
LinkedIn	29718	5572	22287	3714	14858	1857	7428
Google+	50.000	9375	37499	6250	24999	3125	12499
Instagram	1100	206	824	137	549	68	274
Pinterest	250	46	186	31	124	15	61
YouTube	2.500	468	1874	312	1249	156	624
WhatsApp	30000	5625	22499	3750	14999	1875	7499
Gmail	35.000.000	6562500	26249999	4375000	17499999	2187500	8749999
WeChat	16.207	3038	12154	2025	8102	1012	4050
Imo	Virtually Unlimited	Virtually Unlimited	Virtually Unlimited	Virtually Unlimited	Virtually Unlimited	Virtually Unlimited	Virtually Unlimited
Hangouts	Virtually Unlimited	Virtually Unlimited	Virtually Unlimited	Virtually Unlimited	Virtually Unlimited	Virtually Unlimited	Virtually Unlimited
Telegram	4096	768	3071	512	2047	256	1023
Line	10000	1875	7499	1250	4999	625	2499
Tango	520	97	389	65	259	32	129
QQ	16.207	3038	12154	2025	8102	1012	4050

intentional deletion in an attempt of malicious modification). For the simulation-based experiment, all the watermarked CT is truncated at the sentence's beginning, middle, and end, where the cropped location area varies from 20% to 80%. The robustness of this simulation-based experiment is calculated from the percentage of remaining ZWC over the total number of the original watermark's ZWC.

The security aspect evaluation focuses on the method's text integrity verification system. It is evaluated whether the ANiTW and the proposed method can detect the modification applied on watermarked CT. Moreover, the possibility of obtaining the map of the watermark character in ANiTW and the proposed method is also calculated to evaluate the probabilities for the attacker to convert the ZWC pattern to the original hidden watermark character using brute force.

B. EXPERIMENT RESULT

This section will discuss the experiment result and the analyses of both ANiTW and the proposed method based on the experiment scenario.

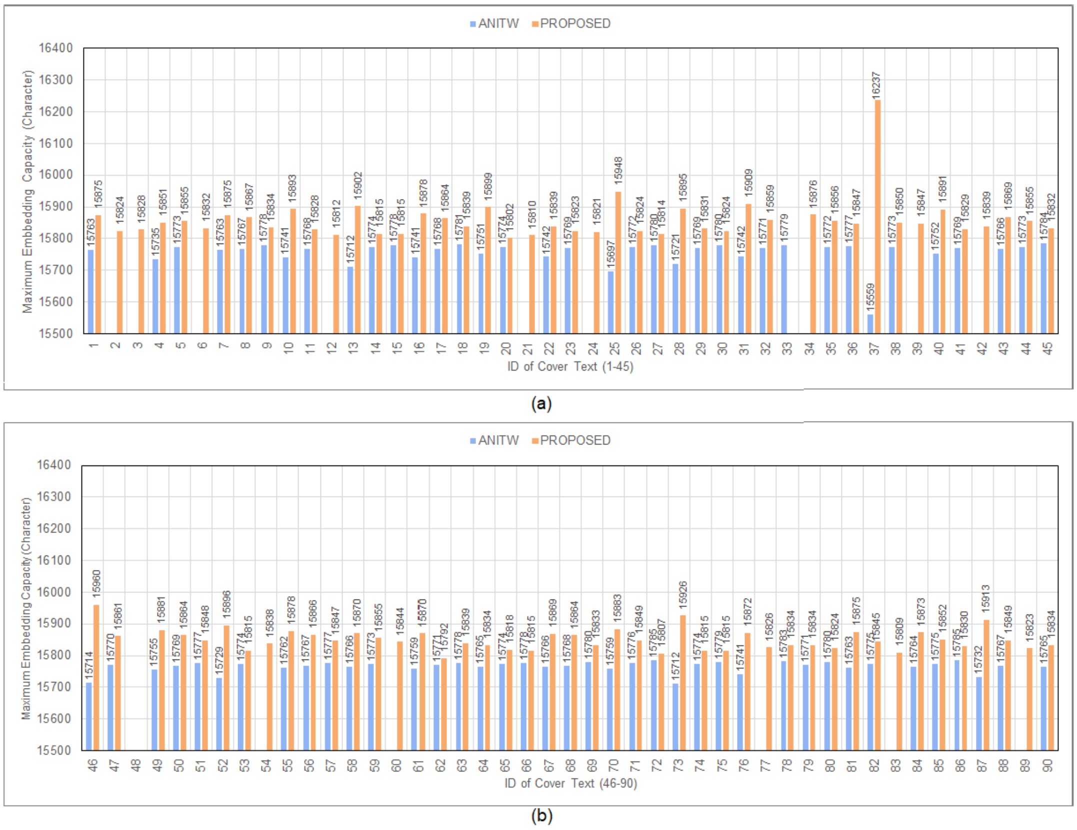
1) MAXIMUM EMBEDDING CAPACITY ANALYSIS BETWEEN ANiTW AND PROPOSED METHOD

For the first scenario, where all CT's character is considered an embeddable location, the maximum embedding capacity comparison between ANiTW and the proposed method is shown in Table 11. As shown in Table 11, the embedding capacity of the proposed method is higher than ANiTW, with an improvement of at least 400%. Thus it is shown that, in the optimum condition, the proposed method is able to increase the maximum embedding capacity of ANITW.

The embedding capacity of the proposed method is also higher than ANITW for the second experiment scenario, where the embeddable location depends on a social media message condition. For this second scenario, before calculating the maximum embedding capacity, the CT has to be evaluated to determine whether it is embeddable. The maximum embedding capacity comparison between ANiTW and the proposed method for each CT in the dataset is shown in Fig. 5. As shown in the experiment result, when the CT is embeddable, the embedding capacity of the proposed method is higher than ANiTW, with a maximum improvement of around 400% for CT with ID 160 and 163. This condition occurred because, with the same CT length, the proposed method can embed more characters than ANiTW by using the proposed bigram-based character encoding.

However, there are cases where the CT is unembeddable using the proposed method, as shown in CT with ID 33 and 48. This condition occurs since CT with ID 33 consists of only one link. Meanwhile, the CT with ID 48 consists of only one hashtag. Therefore, the CT has no available embedding location following the rule discussed in III-B1.d. Meanwhile, on ANiTW, the CT with ID 2, 3, 6, 12, 21, 24, 34, 39, 42, 48, 54, 60, 77, 83, 89, 108, 132, 141, and 153 is considered unembeddable using ANiTW since no punctuation mark end a sentence ('.', '!', and '?') on the CT.

The maximum embedding capacity of the proposed method is greater than the ANiTW method in most cases of the experiment. The proposed method can achieve a greater maximum embedding capacity by prioritizing the MM method to embed the watermark. The higher the number of matching characters found on CT, the lesser the number of ZWC needed to represent watermark characters. Therefore, the higher the number of MM methods found, the higher the maximum embedding capacity. Thus, it is proved that the



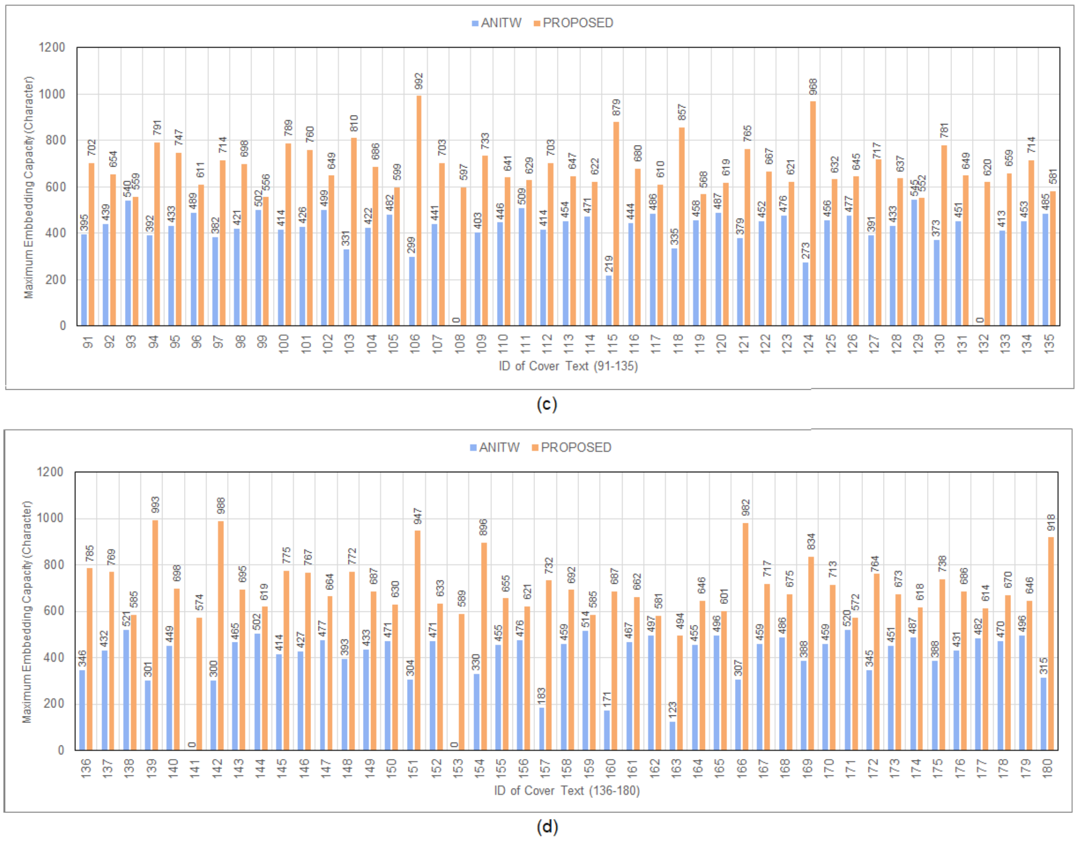


FIGURE 5. Maximum embedding capacity comparison between ANiTW and proposed method (a), (b) on Facebook and (c), (d) on Instagram.

proposed method's watermark mapping is more efficient than the ANiTW.

2) INVISIBILITY ANALYSIS BETWEEN ANiTW AND PROPOSED METHOD

Based on the statistical and visual comparison results, the proposed method performed better in the invisibility aspect in most experiments. For the statistical comparison, since ANiTW and the proposed method are embedding the watermark by inserting the ZWC into the CT, there is no character transposition or *t* in the Jaro-Similarity calculation. Consequently, the statistical similarity between original and watermarked CT for both methods depends on the number of matching characters and the length of each CT. The more ZWC is embedded between the original characters, the less possibility that two original characters are considered a match. Thus, it can be concluded that in most cases, the more ZWC used to embed the watermark, the smaller the Jaro-Similarity.

The assumption above is proven following the experiment result, as shown in Fig. 6, where the proposed method scored higher than ANiTW for the Jaro-Similarity calculation between the watermarked CT and the original CT. For the case where the message is embeddable, the proposed method scored an average of 0.936, whereas ANiTW scored 0.86. This condition occurred because the proposed method mapping concept is more efficient than ANiTW, such as the number of ZWC used to embed the same watermark length is smaller than ANiTW. However, there is a case where the Jaro-Similarity of the proposed method is less than ANiTW, as seen on CT with ID 129. This condition occurred because the length of CT is relatively short, such that any additional inserted character will increase the number of unmatched characters and decrease the similarity.

For the visual comparison experiment, the watermark in ANiTW and the proposed method are considered invisible if there is no visual difference between the watermarked CT and the original CT. Using the proposed method, 89.44% of 180 data experimented with is invisible, whereas, by ANiTW, only 29.44% of all data is invisible. Most of the visual differences for ANiITW are found in the hashtags, links, and mentions entities on the watermarked CT.

The ZWC placement of ANiTW can cause visual differences, as shown in Fig. 7. This condition occurs because all the entities embedded with ZWC are considered two separate instances at the last character of a word and the '.'. For example, the hashtag "#EnergizingYou" is sliced into two, "#EnergizingYo" and 'u' and the links " https://bit.ly/3LHfPHL" is sliced into two, "https://bit" and ".ly//3LHfPHL". Furthermore, visual differences can also occur in emoticons. This difference is occurred because the ZWC embedded will contradict the existing ZWC used to format the emoticon, erasing the existing ZWC function to show the color choice from the original emoticon. Aside from the visual, the function will change since those watermarked entities are linked to a different endpoint.

Thus, it is proven that statistically and visually, the invisibility aspect of watermarked text using the proposed method is better than ANiTW.

3) ROBUSTNESS ANALYSIS BETWEEN ANiTW AND PROPOSED METHOD

The CT length is the same for both ANiTW and the proposed method. Following that, by using (6), the probability of the attacker founding the watermark character on watermarked CT (P(DR)) can be calculated using the number of embedding locations. The higher the number of embedding locations, the lower the P(DR) is, and the less robust the watermarked CT is. Based on the experiment result shown in Fig. 8, for the case where CT is embeddable, ANiTW scored higher than the proposed method in 152 cases. The proposed method P(DR) average is 0.95, whereas the ANiTW P(DR) average is 0.98. This condition occurred because the proposed method prioritizes a higher maximum embedding capacity by maximizing the usage of MM, such as the embedding location of each ZWC is more widely distributed on CT than ANiTW.

All the watermarked CT is truncated at the sentence's beginning, middle, and end, where the cropped location area varies from 20% to 80%. The robustness of this simulation-based experiment is calculated from the number of remaining ZWC over the total number of the original watermark's ZWC. The result of the simulation-based experiment is shown in Fig. 9. In the case where the watermarked CT is truncated from the end of the sentence, the proposed method scored higher than ANITW with an average of 80.12%, whereas the ANiTW only scored 45.59%. However, where the watermarked CT is truncated from the beginning and the middle of the sentence, the proposed scored lower than ANiTW. The proposed method's average score is 38.20% and 39.63%, whereas the ANiTW score is 61.55% and 42.90%. This condition occurred because the proposed method embedding algorithm is designed to find the possible embedding location for MM starting from the beginning of CT. This design is necessary since the end of the message in social media gets truncated when the maximum character count is exceeded.

4) SECURITY ANALYSIS BETWEEN ANiTW AND PROPOSED METHOD

For the text integrity verification, the watermarked CT is modified to simulate the scenario where an attacker has obtained the watermarked CT, manipulated it, and then reshared the manipulated watermarked CT on social media. The manipulation includes the insertion, deletion, reordering, and replacement of words or characters into the watermarked CT (e.g., an attacker might do some character insertion to perform a phishing attack by redirecting other users using the wrong links as shown in modified watermarked CT with ID CTE 1 as shown in Table 12). We evaluated both ANiTW and the proposed method's ability to detect modification over the CT, such that it can prevent an attacker from gaining a

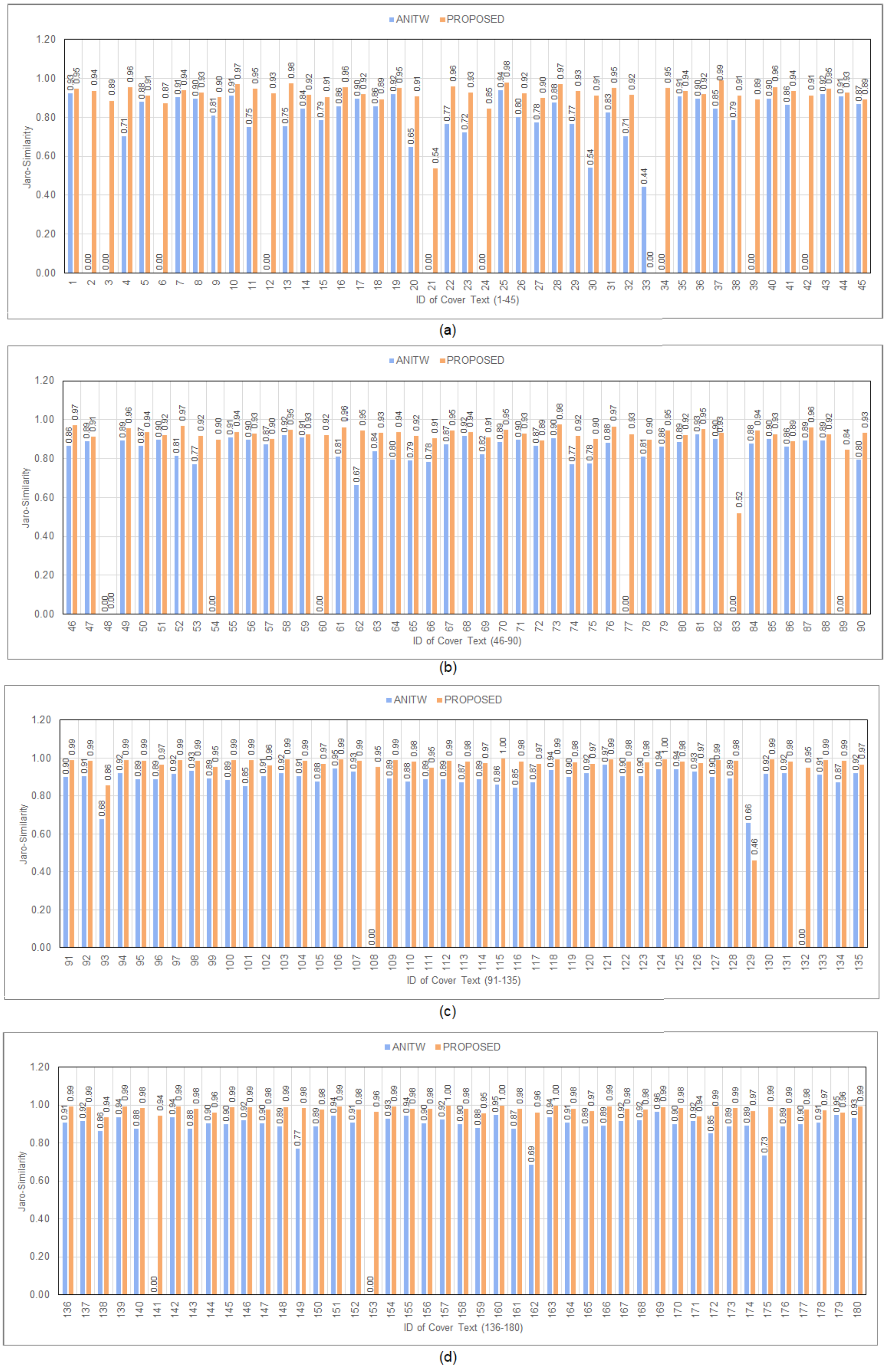
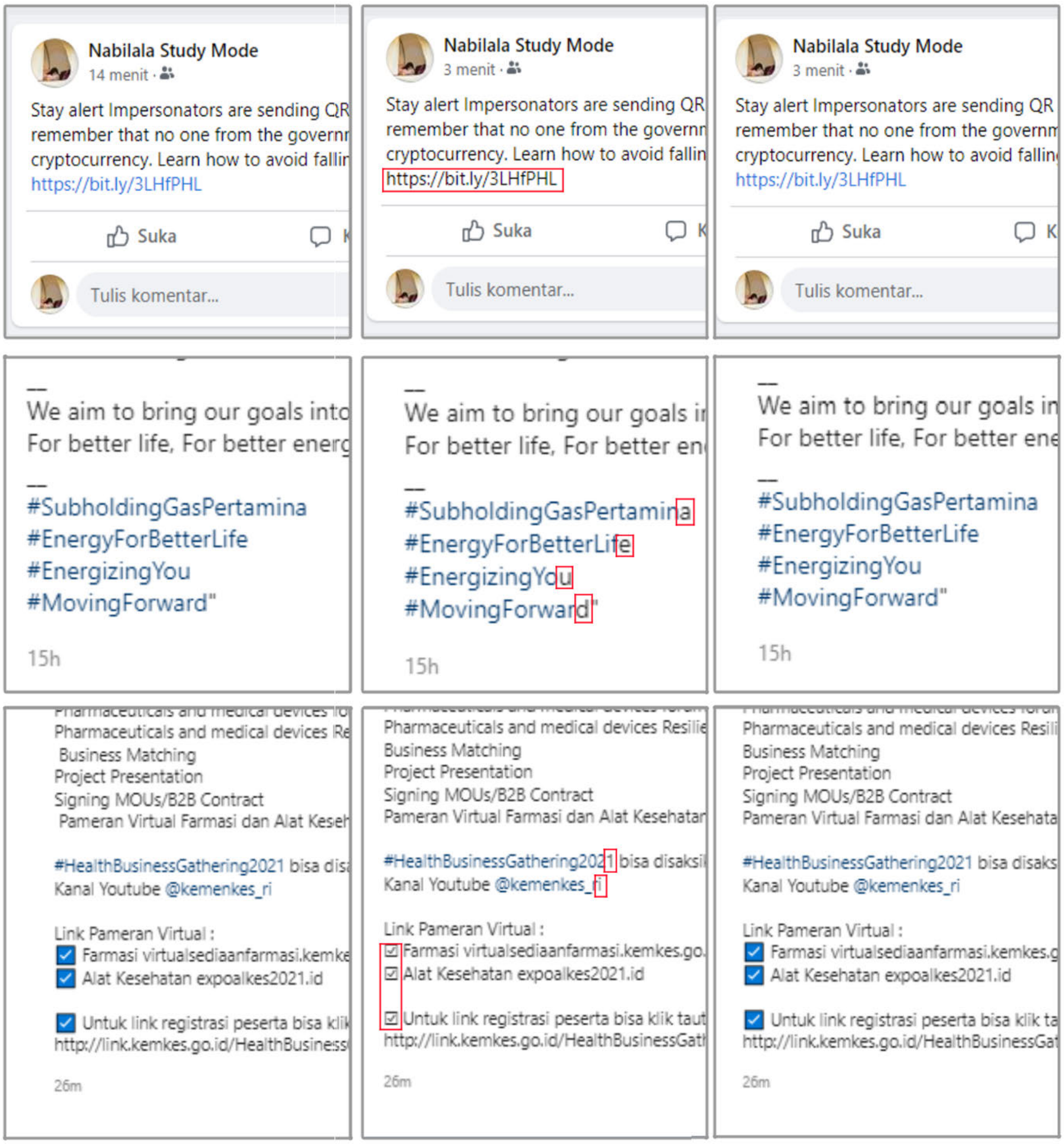


FIGURE 6. Watermarked cover text statistical similarity comparison between ANiTW and proposed method (a), (b) on Facebook and (c), (d) on Instagram.



Nabilala Study Mode
14 menit ·
Stay alert Impersonators are sending QR
remember that no one from the governr
cryptocurrency. Learn how to avoid fallir
https://bit.ly/3LHfPHL
Suka
Tulis komentar...

Nabilala Study Mode
3 menit ·
Stay alert Impersonators are sending QR
remember that no one from the governn
cryptocurrency. Learn how to avoid fallin
https://bit.ly/3LHfPHL
Suka
Tulis komentar...

Nabilala Study Mode
3 menit ·
Stay alert Impersonators are sending QR
remember that no one from the governm
cryptocurrency. Learn how to avoid fallin
https://bit.ly/3LHfPHL
Suka
Tulis komentar...

—
We aim to bring our goals into
For better life, For better energ
—
#SubholdingGasPertamina
#EnergyForBetterLife
#EnergizingYou
#MovingForward"
15h

—
We aim to bring our goals i
For better life, For better en
—
#SubholdingGasPertamina
#EnergyForBetterLife
#EnergizingYou
#MovingForward"
15h

—
We aim to bring our goals in
For better life, For better ene
—
#SubholdingGasPertamina
#EnergyForBetterLife
#EnergizingYou
#MovingForward"
15h

Pharmaceuticals and medical devices Re
Business Matching
Project Presentation
Signing MOUs/B2B Contract
Pameran Virtual Farmasi dan Alat Keseh
#HealthBusinessGathering2021 bisa dis
Kanal Youtube @kemenkes_ri
Link Pameran Virtual :
☑ Farmasi virtualsediaanfarmasi.kemke
☑ Alat Kesehatan expoalkes2021.id
☑ Untuk link registrasi peserta bisa klik
http://link.kemkes.go.id/HealthBusiness
26m

Pharmaceuticals and medical devices Resilie
Business Matching
Project Presentation
Signing MOUs/B2B Contract
Pameran Virtual Farmasi dan Alat Kesehatar
#HealthBusinessGathering2021 bisa disaksi
Kanal Youtube @kemenkes_ri
Link Pameran Virtual :
☑ Farmasi virtualsediaanfarmasi.kemkes.go.
☑ Alat Kesehatan expoalkes2021.id
☑ Untuk link registrasi peserta bisa klik taut
http://link.kemkes.go.id/HealthBusinessGat
26m

Pharmaceuticals and medical devices Resili
Business Matching
Project Presentation
Signing MOUs/B2B Contract
Pameran Virtual Farmasi dan Alat Kesehata
#HealthBusinessGathering2021 bisa disaks
Kanal Youtube @kemenkes_ri
Link Pameran Virtual :
☑ Farmasi virtualsediaanfarmasi.kemkes.g
☑ Alat Kesehatan expoalkes2021.id
☑ Untuk link registrasi peserta bisa klik ta
http://link.kemkes.go.id/HealthBusinessGat
26m

FIGURE 7. Visual comparison of social media original CT (Left), watermarked CT using ANiTW (Middle) and watermarked CT using the proposed method (Right).

malicious advantage. As discussed in Section II-B, ANiTW considers the received text invalid if the AC < 1. Meanwhile, the proposed method considers the text invalid if the text integrity and user authentication verification status are false.

Based on the experiment result, the proposed method performed better than ANiTW in detecting a modification implemented on the watermarked CT. ANiTW can only detect 46.67% out of the total of 180 modified watermarked CT, while the proposed method is able to detect 93.89%. This condition occurred because the ANiTW integrity checker only depends on the changes in the number of words in watermarked CT. Therefore, modification without changing the watermarked CT's number of words will go undetected. Meanwhile, the proposed method used the hash value of the message that is able to detect any changes in the original CT's alphabet character and digits sequences.

However, there are 6.11% of cases where the proposed method cannot detect the manipulation applied on

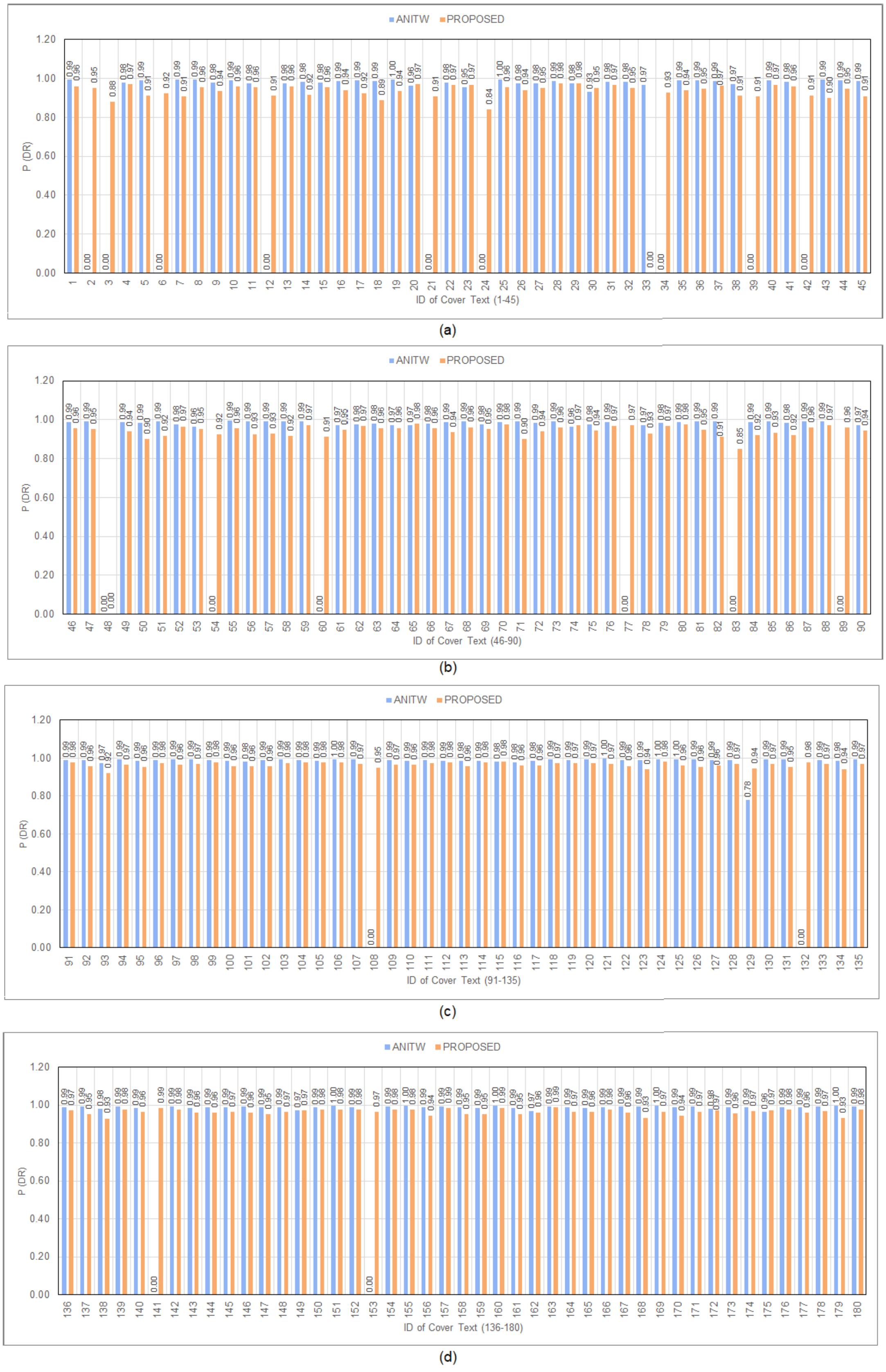


FIGURE 8. Distortion robustness comparison between ANiTW and proposed method (a), (b) on Facebook and (c), (d) on Instagram.

TABLE 12. **Comparison between ANiTW and proposed method on detecting modification on watermarked text.**

CT ID	Original Watermarked CT	Modified Watermarked CT	AC	Integrity Verification Status	Signature Verification Status	Detected Change?	
						ANiTW	Proposed
CTE 1	... Learn how to avoid falling for QR code and other crypto scams: https://bit.ly/3LHfPHL	... Learn how to avoid falling for QR code and other crypto scams: https://bit.ly/3UVXLHfPHL	1	FALSE	FALSE	FALSE	TRUE
CTE 2	...Learn more through CDC.gov/Features/TurkeyTime thanks to @cdcgov. //...	...Learn more through CDC.gov/Features/TurkeyTime thanks to @USAcdcgov. //...	1	FALSE	FALSE	FALSE	TRUE
CTE 3	... pasien positif terbanyak, dari 424 pasien menjadi 472 pasien dalam sehari.	... pasien positif terbanyak, dari 4024 pasien menjadi 4720 pasien dalam sehari.	1	FALSE	FALSE	FALSE	TRUE
CTE 4	...Tidak hanya saksi, Haydar juga menegaskan jika ...	...Selain saksi, Haydar juga menegaskan jika ...	0.96	FALSE	FALSE	TRUE	TRUE

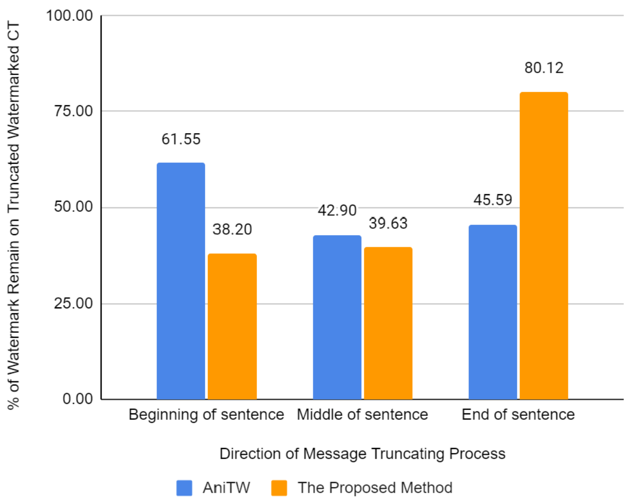


FIGURE 9. **Comparison of percentage of watermark remain on truncated watermarked CT.**

watermarked CT. Two reasons cause the undetected 6.11% case using the proposed method, the modification ignorance and format changes of watermarked CT once posted. The modification of characters such as '!' or '?' on the proposed method is ignored. In addition, some links from the watermarked CT on Facebook can automatically be converted into a linked webpage, which results in negative false in the extraction process.

From the result, we can conclude that in most cases, any modification of character or number in watermarked CT can be detected by the proposed method using the part of message hash and the signature from IBS. While on the ANiTW method, the text integrity checker based on word as CT'S instances can still be bypassed, especially using a character insertion attack.

Another thing to discuss in the security evaluation is the probability of obtaining the watermark character mapping. Some attackers may learn the ZWC pattern from the watermarked text. For example, the attacker may see the repeating pattern of ZWC representing the user's name or ID as proof of text ownership. Using that knowledge, an attacker can perform an impersonation attack by copy-pasting the ZWC into their own CT, pretending to be the valid user. This attack will depend on the probability of obtaining the watermark character mapping. The security aspect for ANiTW and the proposed method for this evaluation depends on the look-up table used to encode the watermark character. The security evaluation for both methods is conducted based on the probability of obtaining the map of the watermark character. Suppose there are n ID characters that are mapped into x unique code. Then there is a P_n^x possible map.

As discussed in II-B, two look-up tables are used in ANiTW. These look-up tables convert a character into a 6-bits binary string and a 3-bits binary string into 2 ZWC. Thus, if there is an attack attempted on the ANiTW method, the probability of obtaining the correct mapping is $1/[P_{64}^{64} \cdot (P_8^9)^2]$. In the proposed method, instead of only guessing the ZWC-Character map, the attacker must also guess the embedding mode used. As discussed in III-B1.d, there are four embedding modes, NM, AMM, BMM, and BAMM. Following that, the probability of obtaining the correct map in the proposed method is $1/\{P_{64}^{100} \cdot \text{shiftVal} \cdot [P_{10}^{36} + (P_{10}^{30})^2 \cdot P_3^6]\}$. Thus, it is clear that the proposed method is more secure than ANiTW in terms of the probability of obtaining the map of the watermark character.

C. DISCUSSION

This section analyses and discusses research findings not included in the main experiment.

1) COMPUTATIONAL COMPLEXITY

In addition to the four watermarking criteria aspect, we can also compare the computation complexity of a watermarking scheme. For both ANiTW and the proposed method, the main embedding algorithm is divided into two processes: mapping the watermarking character into the ZWC and placing the ZWC into a suitable embeddable location. Since the length of the look-up table is static, there is no significant difference in the mapping process. Following that, the computational complexity between both methods is compared by the embeddable location searching process.

On ANiTW, the algorithm will search for a punctuation mark that ends a CT sentence, such as '.', '!', and '?' to place grouped watermark's ZWC representation. Suppose a watermark with a length of L is embedded into a CT with a length of N, the computational complexity for this process in ANiTW is $O(L)$. Meanwhile, the proposed method finds the suitable embedding location for each watermark character by searching for a suitable CT character following the rules mentioned in Section III-B1.d. The computational complexity for this process in the proposed method is $O(NL)$. Thus, the computational complexity between ANiTW and the proposed method depends on the message's length.

V. CONCLUSION

In order to improve the performance and overcome the problem of ANITW, this proposed method provides a novel approach to a text watermarking scheme that embeds a high-level security component to verify the message's content integrity and source authenticity. To secure the integrity aspect of social media messages, a part of the message hash is implemented as a watermark so the system can detect a malicious modification over a watermarked text. In addition, to improve the source authenticity, an ID and signature generated by IBS Scheme are embedded, which enable the reader to authenticate the received message's source easily. Moreover, to improve the performance of ANITW, a novel bigram-based character encoding and embedding rule are proposed to enable the scheme to achieve a high embedding capacity and invisibility. The bigram encoding can obtain a high embedding capacity, so a longer hash and signature function can be embedded into social media messages as a cover text with limited space. As an impact, the proposed watermarking scheme can achieve a higher security level. Based on the results of the analysis result, the proposed method is applicable as a tool to verify text integrity and source authenticity without interfering with the functionality of social media messages. The experiment results proved that the proposed method is able to overcome the vulnerabilities and improve the performance of ANITW in terms of embedding capacity, invisibility, and security. For future works, a modification and improvement of the embedding mode, position setting, and embedding location rule on the proposed method can be evaluated to implement the designed scheme in another aspect. Repeating the watermarking process with a modified rule allows the proposed method watermarking scheme to be used as a copy-control or modification history tracker for Latin-based text documents. Adding an escape character into the scheme can also prevent the proposed method's false negative extraction process. Moreover, a multi-level n-gram coding system can also be designed to improve the embedding capacity such that a higher security level of the watermark can be implemented. The extraction process can also be improved by analyzing a higher n-gram value from the watermarked CT to identify illegal embedding and to add an error correction feature to the system.

REFERENCES

[1] S. Dixon, "Number of social media users worldwide from 2018 to 2027," Tech. Rep., Jul. 2022.

[2] S. Rathore, P. K. Sharma, V. Loia, Y.-S. Jeong, and J. H. Park, "Social network security: Issues, challenges, threats, and solutions," *Inf. Sci.*, vol. 421, pp. 43–69, Dec. 2017.

[3] S. Deliri and M. Albanese, "Security and privacy issues in social networks," in *Data Management in Pervasive Systems*. 2015, pp. 195–209.

[4] A. Hutchinson, "Internal research from Facebook shows that re-shares can significantly amplify misinformation," Tech. Rep., May 2021.

[5] S. Lee and J. Kim, "WarningBird: A near real-time detection system for suspicious URLs in Twitter stream," *IEEE Trans. Depend. Sec. Comput.*, vol. 10, no. 3, pp. 183–195, May 2013.

[6] N. Hidayah, "Laporan pemetaan hoaks edisi desember 2021," Tech. Rep., 2021.

[7] M. H. Alkawaz, G. Sulong, T. Saba, A. S. Almazyad, and A. Rehman, "Concise analysis of current text automation and watermarking approaches," *Secur. Commun. Netw.*, vol. 9, no. 18, pp. 6365–6378, Dec. 2016.

[8] N. F. Johnson, "FRANK Y. SHIH* digital watermarking and steganography: Fundamentals and techniques. CRC/Taylor & Francis (2008).*ISBN-13: 9781420047578. 46.99. 180, pp. Hardcover," *Comput. J.*, vol. 53, no. 5, pp. 616–617, Jun. 2010.

[9] R. Petrovic, B. Tehranchi, and J. M. Winograd, "Security of copy-control watermarks," in *Proc. 8th Int. Conf. Telecommun. Modern Satell., Cable Broadcast. Services*, Sep. 2007, pp. 117–126.

[10] M. T. Ahvanooey, Q. Li, X. Zhu, M. Alazab, and J. Zhang, "ANiTW: A novel intelligent text watermarking technique for forensic identification of spurious information on social media," *Comput. Secur.*, vol. 90, Mar. 2020, Art. no. 101702.

[11] N. S. Kamaruddin, A. Kamsin, L. Por, and H. Rahman, "A review of text watermarking: Theory, methods, and applications," *IEEE Access*, vol. 6, pp. 8011–8028, 2018.

[12] M. T. Ahvanooey, Q. Li, J. Hou, A. R. Rajput, and C. Yini, "Modern text hiding, text steganalysis, and applications: A comparative analysis," *Entropy*, vol. 21, no. 4, p. 355, Apr. 2019.

[13] O. F. A. Adeeb and S. J. Kabudian, "Arabic text steganography based on deep learning methods," *IEEE Access*, vol. 10, pp. 94403–94416, 2022.

[14] H. M. Bashir, Q. Li, and J. Hou, "A high capacity text steganography utilizing unicode zero-width characters," in *Proc. Int. Conf. Internet Things (iThings), IEEE Green Comput. Commun. (GreenCom), IEEE Cyber, Phys. Social Comput. (CPSCom), IEEE Smart Data (SmartData), IEEE Congr. Cybermatics (Cybermatics)*, Nov. 2020, pp. 668–675.

[15] N. Naqvi, A. T. Abbasi, R. Hussain, M. A. Khan, and B. Ahmad, "Multilayer partially homomorphic encryption text steganography (MLPHE-TS): A zero steganography approach," *Wireless Pers. Commun.*, vol. 103, no. 2, pp. 1563–1585, Nov. 2018.

[16] M. T. Ahvanooey, Q. Li, J. Hou, H. D. Mazraeh, and J. Zhang, "AITSteg: An innovative text steganography technique for hidden transmission of text message via social media," *IEEE Access*, vol. 6, pp. 65981–65995, 2018.

[17] R. Kumar, S. Chand, and S. Singh, "An efficient text steganography sheme using unicode space characters," *Int. J. Forensic Comput. Sci.*, vol. 10, no. 1, pp. 8–14, Dec. 2015.

[18] M. Azeem, J. He, A. Ditta, and F. Akhtar, "A novel approach to secret data concealment with high cover text capacity and security," *Int. J. Electron. Secur. Digit. Forensics*, vol. 12, no. 1, p. 1, 2020.

[19] M. T. Ahvanooey, Q. Li, H. J. Shim, and Y. Huang, "A comparative analysis of information hiding techniques for copyright protection of text documents," *Secur. Commun. Netw.*, vol. 2018, pp. 1–22, Apr. 2018.

[20] C. Cachin, "An information-theoretic model for steganography," *Inf. Comput.*, vol. 192, no. 1, pp. 41–56, Jul. 2004.

[21] L. Y. Por, K. Wong, and K. O. Chee, "UniSpaCh: A text-based data hiding method using unicode space characters," *J. Syst. Softw.*, vol. 85, no. 5, pp. 1075–1082, 2012.

[22] P. Singh and R. Chadha, "A survey of digital watermarking techniques, applications and attacks," *Int. J. Eng. Innov. Technol. (IJEIT)*, vol. 2, no. 9, pp. 165–175, Feb. 2013.

[23] Y. Wang and P. Moulin, "Perfectly secure steganography: Capacity, error exponents, and code constructions," *IEEE Trans. Inf. Theory*, vol. 54, no. 6, pp. 2706–2722, Jun. 2008.
[24] M. T. Ahvanooey, H. D. Mazraeh, and S. H. Tabasi, "An innovative technique for web text watermarking (AITW)," *Inf. Secur. J., Global Perspective*, vol. 25, no. 6, pp. 191–196, 2016.
[25] S. Kingslin, "Evaluative approach towards text steganographic techniques," *Indian J. Sci. Technol.*, vol. 8, no. 1, pp. 1–8, Jan. 2015.
[26] X. Zhou, Z. Wang, W. Zhao, and J. Yu, "Attack model of text watermarking based on communications," in *Proc. Int. Conf. Inf. Manag., Innov. Manag. Ind. Eng.*, 2009, pp. 283–286.
[27] F. Hess, "Efficient identity based signature schemes based on pairings," in *Proc. Selected Areas Cryptogr., 9th Annu. Int. Workshop*, 2003, pp. 310–324.
[28] E. Kiltz and G. Neven, "Identity-based signatures," *Identity-Based Cryptogr.*, vol. 2, no. 31, p. 75, 2009.
[29] K. G. Paterson and J. C. N. Schuldt, "Efficient identity-based signatures secure in the standard model," in *Proc. Inf. Secur. Privacy, 11th Australas. Conf.*, 2006, pp. 207–222.
[30] A. Shamir, "Identity-based cryptosystems and signature schemes," in *Proc. Workshop Theory Appl. Cryptograph. Techn.*, 1985, pp. 47–53.
[31] J. C. Choon and J. H. Cheon, "An identity-based signature from gap Diffie-Hellman groups," in *Proc. Int. Workshop Public Key Cryptogr.*, 2003, pp. 18–30.
[32] R. Sakai. (2000). *Cryptosystems Based on Pairing*. [Online]. Available: https://cir.nii.ac.jp/crid/1570291225722108672.bib?lang=en
[33] X. Yi, "An identity-based signature scheme from the Weil pairing," *IEEE Commun. Lett.*, vol. 7, no. 2, pp. 76–78, Feb. 2003.
[34] R. Yaduvanshi and S. Mishra, "An efficient and secure pairing free short ID-based signature scheme over elliptic curve," *SSRN Electron. J.*, 2019.
[35] S. Mishra, R. Yaduvanshi, K. Dubey, and P. Rajpoot, "ESS-IBAA: Efficient, short, and secure ID-based authentication algorithm for wireless sensor network," *Int. J. Commun. Syst.*, vol. 34, no. 8, p. e4764, May 2021.
[36] D. M. Kar and I. Ray, "Systematization of knowledge and implementation: Short identity-based signatures," 2019, *arXiv:1908.05366*.
[37] R. L. Solso and C. L. Juel, "Positional frequency and versatility of bigrams for two—Through nine-letter English words," *Behav. Res. Methods Instrum.*, vol. 12, no. 3, pp. 297–343, May 1980.
[38] H. Aldabbas, A. Bajahzar, M. Alruily, A. A. Qureshi, R. M. A. Latif, and M. Farhan, "Google play content scraping and knowledge engineering using natural language processing techniques with the analysis of user reviews," *J. Intell. Syst.*, vol. 30, no. 1, pp. 192–208, Jul. 2020.
[39] P. S. Dixon. (Jul. 2022). *Biggest Social Media Platforms 2022*. [Online]. Available: https://www.statista.com/statistics/272014/global-social-networks-ranked-by-number-of-users/

NABILA B. ROFIATUNNAJAH received the B.Sc. degree from the Department of Electrical Engineering, Bandung Institute of Technology, Indonesia, in 2020, and the M.Sc. degree from the School of Computing, Telkom University, Bandung, Indonesia, in 2022. Her current research interests include data security, steganography, and cryptography.

ARI M. BARMAWI (Member, IEEE) received the B.Sc. degree from the Department of Electrical Engineering, Bandung Institute of Technology, Indonesia, in 1985, and the M.Sc. and Ph.D. degrees from the Department of Computer Science, Keio University, Japan, in 1997 and 2001, respectively. She is currently an Associate Professor in cyber security and digital forensic study program with Telkom University. Her research interests include cryptography, steganography, and artificial intelligence. She is a member of the International Association of Cryptography Researcher (IACR) and ACM.