CovertSYS

A systematic covert communication approach for providing secure end-to-end conversation via social networks

Ahvanooey, Milad Taleby; Zhu, Mark Xuefang; Mazurczyk, Wojciech; Li, Qianmu; Kilger, Max; Choo, Kim Kwang Raymond; Conti, Mauro

DOI
10.1016/j.jisa.2022.103368

Publication date
2022

Document Version
Final published version

Published in
Journal of Information Security and Applications

Citation (APA)
Ahvanooey, M. T., Zhu, M. X., Mazurczyk, W., Li, Q., Kilger, M., Choo, K. K. R., & Conti, M. (2022). CovertSYS: A systematic covert communication approach for providing secure end-to-end conversation via social networks. *Journal of Information Security and Applications*, *71*, Article 103368. https://doi.org/10.1016/j.jisa.2022.103368

Journal of Information Security and Applications 71 (2022) 103368

Contents lists available at ScienceDirect

Journal of Information Security and Applications

journal homepage: www.elsevier.com/locate/jisa

CovertSYS: A systematic covert communication approach for providing secure end-to-end conversation via social networks

Milad Taleby Ahvanooey [a,b,c,*], Mark Xuefang Zhu [c], Wojciech Mazurczyk [d], Qianmu Li [e], Max Kilger [f], Kim-Kwang Raymond Choo [f], Mauro Conti [g,h]

[a] School of Computer Science and Engineering, Nanyang Technological University (NTU), P.O.Box. 639798, Singapore
[b] Cybercoding IT.Co., Ltd, Tsinghua University Science (TUS) Park, Jiangning, Nanjing, P.O. Box 211189, PR China
[c] School of Information Management, Nanjing University (NJU), Nanjing, P.O. Box 210023, PR China
[d] Institute of Computer Science, Faculty of Electronics and Information Technology, Warsaw University of Technology (WUT), Poland
[e] School of Cyber science and Engineering, Nanjing University of Science & Technology (NJUST), P.O. Box 210094, PR China
[f] Department of Information Systems and Cyber Security, University of Texas at San Antonio (UTSA), San Antonio, TX 78249-0631, USA
[g] Department of Mathematics, University of Padua (UNIPD), 63-35121, Padua, Italy
[h] Faculty of Electrical Engineering, Mathematics and Computer Science, Delft University of Technology (TU Delft), 2600 AA, Delft, Netherlands

ARTICLE INFO

Keywords:
Information hiding
Text steganography
Covert communication
Privacy preservation
Applied cryptography

ABSTRACT

While encryption can prevent unauthorized access to a secret message, it does not provide undetectability of covert communications over the public network. Implementing a highly latent data exchange, especially with low eavesdropping/discovery probability, is challenging for practical scenarios, such as social and political movements in authoritarian regimes, military operations, and privacy preservation. Moreover, the current literature suffers from a low embedding capacity and monolingual applicability, limiting the amount of hiding secret data within short text messages using state-of-the-art algorithms, e.g., linguistic-based, structural-based, or coverless-based solutions. In this paper, we present a systematic covert communication technique called CovertSYS that enables a multilingual secure end-to-end conversation via messaging or social network platforms. The CovertSYS functions by encrypting a confidential message using a multi-factor authentication scheme and converting the encoded binary data into hidden Unicode symbols to be transmitted under cover of short text messages. We then conduct extensive experiments to confirm the security and validity of the proposed technique against state-of-the-art approaches. Our experimental results show that the CovertSYS provides a superior mean performance of 91.53% by improving the criteria scores: embedding capacity rate of 100%, imperceptibility rate of 76.4%, and distortion robustness rate of 98.2%. Finally, we discuss the practical implications of the proposed technique compared to the existing text steganography methods.

1. Introduction

Text chat via online messaging or social network platforms such as WhatsApp, WeChat, and short message service (SMS) remains a common form of communication between users via Information and Communications Technology (ICT). However, many messaging applications (apps) are generally not designed for end-to-end secure conversation. In other words, they are vulnerable to being intercepted by various entities, such as data centers, service providers, law enforcement agencies (LEA) and government security services [1]. As a result, there are several challenges in developing efficient covert communication mechanisms in terms of provable security and performance while meeting stringent requirements, such as steganography criteria, and maintaining an acceptable level of real-world usability [2].

Existing secure communication mechanisms generally focus on preventing an adversary from reading the content of the secret information. There are, however, sensitive situations where communicating parties prefer to additionally use covert communication systems to conceal the very existence of a confidential conversation by hiding their conversations under cover of another media or host object [3]. Examples of such situations include whistle-blowers trying to conceal their conversations while communicating with journalists or government officials. Such system is essential in sensitive environments such as those found in authoritarian countries for security in organizations

* Corresponding author at: Cybercoding IT.Co., Ltd, Tsinghua University Science (TUS) Park, Jiangning, Nanjing, P.O. Box 211189, PR China.
E-mail addresses: M.taleby@ieee.org (M. Taleby Ahvanooey), xfzhu@nju.edu.cn (M.X. Zhu), wojciech.mazurczyk@pw.edu.pl (W. Mazurczyk), Qianmu@njust.edu.cn (Q. Li), Max.Kilger@utsa.edu (M. Kilger), raymond.choo@fulbrightmail.org (K.R. Choo), conti@math.unipd.it (M. Conti).

https://doi.org/10.1016/j.jisa.2022.103368

Available online 18 November 2022

where the discovery of covert chats poses a severe physical or legal threat to one or more parties to the communication. Note that in some cases, knowledge of the existence of exchanged covert conversations is as valuable or more important than the content of the confidential message itself. The suspicion or discovery of exchanged covert conversations can put significant resources in motion on the part of security service and focus those resources on previously unknown parties and activities that need to remain hidden to accomplish their mission.

Technically, designing a covert communication system is a highly complex task because some other external entities (e.g., Internet service providers, database servers, and telecommunications centers) are involved in successfully transmitting hidden messages. Such entities and messaging platforms may expose shared knowledge and sensitive personal information to surveillance agencies [4]. Consequently, there is a need for a secure end-to-end covert communication system that can meet the essential requirements of text steganography and the necessity to keep the existence of secret text chats covert to third-party eavesdroppers. These requirements include: *(i)* all parties require encryption elements or keys based on objects such as a unique password to protect their confidential conversations, and these elements are not disclosed over the transmission channels, and *(ii)* the adversary should not be able to discover the content of exchanged messages between two parties through any other means such as the user's device or external technologies, i.e., even if the adversarial party gains complete knowledge of the used scheme except for the communicating parties' secret key according to Kerckhoffs's principle [5].

In general, text steganography as a *method of covert communication* in the form of short text messages is a highly complex task compared to other digital media [6] and means of communication, as it requires a sufficient number of linguistic features (e.g., words/letters) [7–9] and formatting characteristics (e.g., font, character encoding) [10–14]. In the literature, most existing methods lose their efficiency when applied to short text messages due to several drawbacks, such as dependency on a large number of letters/words for concealing a short confidential message, or they cause perceptible modifications after embedding it inside the carrier text. The main contributions of this study are twofold.

- We propose a multi-factor authentication scheme using an evolutionary algorithm that generates a one-time valid encrypted conversation based on users' passwords (e.g., a shared combination), sending/receiving times, and dates. This algorithm ensures the protection of encrypted conversations using the users' password as the secret key according to Kerckhoffs's principle so that the ciphertext is only accessible to users with the hardware biometric key containing the hashed value of the password combination.
- We introduce a multilingual steganography approach that employs structural characteristics of the cover text to append a sequence of hidden Unicode symbols based on the encrypted secret bits at the end of its content. These additional symbols do not change the appearance of the carrier text and make the covert data visually imperceptible to the human eye.

The remainder of the paper is organized as follows. Section 2 presents a brief background review of covert communication systems and the criteria requirements. Section 3 describes related material, including various types of text steganography techniques. Section 4 explores three possible cyberattacks on text messages as they are exchanged on online messaging platforms. In Section 5, we describe the proposed technique in detail. In Section 6, we experiment with the CovertSYS technique and evaluate its efficiency concerning the criteria requirements by performing it on 20 messaging platforms and comparing our results with state-of-the-art algorithms. Section 7 provides the concluding remarks.

Table 1
The Abbreviations and/or Acronyms.

Abbreviations	Description
SM	Secret Message (e.g., any kind of confidential information such as factual evidence against a company or a government)
CM	Cover Message (e.g., an innocent message such as poetry, or greeting.)
H	Hidden Data String
$Emb()$	Embedding Function
CM_H	Carrier Message + H string (i.e., a text message consists of covert data generated by the $Emb()$)
$Ext()$	Extraction Function
$D_f(CM_H)$	Detection Function
$OTP()$	One-Time-Pad Encryption Algorithm
SD	Secret Data String
BC	Block Cipher
SK	Secret Key
EC	Embedding Capacity
BPL	Bit Per Embeddable Locations
L_p	Losing probability
DR	Distortion Robustness
ZWCs	Zero Width Characters
$E_{l\|\|w}$	Embeddable bits per letters or words
ES	Embedded Spots
RE	Required embeddable bits for hiding an SM inside a CM
MitM	Man-in-the-Middle Attack
APT	Advanced Persistent Threat
SPO	Service Provider Operators
Bio-key	Biometric Key or Universal 2nd Factor (U2F) Secret Key
JW_s	Jaro–Winkler Similarity
SDB	Secret Data Binary String
ESDB	Encrypted Secret Data Binary String
PC	Password Combination
SKB	Secret Key Binary String

2. Background study

In this section, we describe a practical scenario for a covert communication system and define criteria requirements for evaluating the efficiency of currently existing approaches. Table 1 contains the definition of abbreviations and acronyms used in this article.

2.1. Covert communication scenario

In the landmark article on information security [15], Shannon introduced three fundamental secrecy systems: cryptosystems, privacy systems, and concealment systems. A cryptosystem encrypts a secret message (SM) in a specific way so that only authorized users can decrypt the ciphertext while it remains unknown to unauthorized parties. It enhances the security of confidential information by scrambling the SM into an indistinguishable and unreadable form. A privacy system limits access to susceptible knowledge about a user/organization such that only authorized parties are allowed to view the content, and unauthorized parties are denied access to the information by ordinary means and circumstances. Although these systems enhance the security of encrypted information, unauthorized parties are still exposed to the existence and appearance of messages, making them vulnerable to interception and cryptanalysis attacks. A concealment system or *information hiding* is a technically different mechanism compared to the other two secrecy systems. This technology employs a cover text, host object, or cover message (CM), to embed an SM or a watermark into its content. It then generates a carrier CM_H by embedding hidden (H) data to be sent via available digital transmission channels. In practice, information hiding conceals the existence of the SM so that it is indistinguishable from the CM_H by casual inspection for the human eye [7,12,16].

Technically, information hiding technology in digital textual contents can be categorized into two major types, differentiated by their method and application: text watermarking and text steganography [17]. When malicious users fabricate and spread falsified documents,

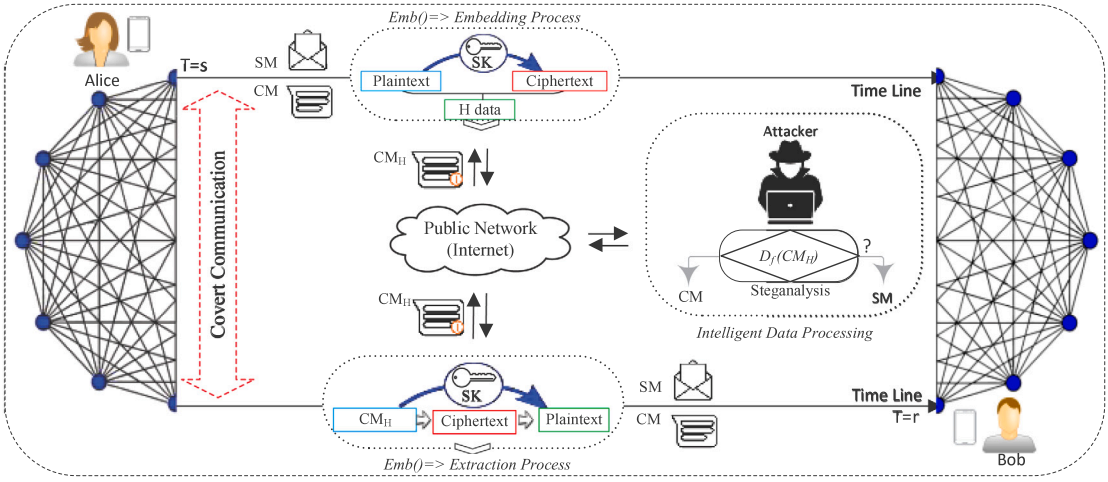


Fig. 1. A potential covert communication scenario in which Alice and Bob utilize a text steganography approach when sending (s) or receiving (r) a CM_H considering an access delay between two parties, denoted as the $T(s,r)$.

text watermarking can be utilized as an effective tool to verify the authenticity of the watermarked information. On the other hand, text steganography can efficiently protect confidential information from being monitored by third-party attacks over open transmission channels. In other words, steganography is one of the most well-known research areas of information hiding that shares some common characteristics with the related but fundamentally different method of the copyright marking known as *digital watermarking*. Nevertheless, steganography and watermarking methods embed an SM or watermark inside the CM to provide different security purposes. The main objective of steganography is to disguise the very existence of confidential information, while in watermarking, the main aim is to resist copyright violation or tampering. Moreover, the payload size embedded using a steganographic approach is significantly higher than watermarking. Such requirements directly affect their application scenarios and make their designs and efficiency evaluations quite different [7,18,19].

In theory, Simmons [20] has introduced the first covert communication scenario, which is known as the *Prisoners' model*. However, this scenario only comprises some limited assumptions for applying it in a real environment that arguably suffers from two drawbacks: (i) Prisoners' communications are under the watch of the Warden's eyes during the writing of CM_H; (ii) the Warden must provide a paper (hard copy) as the transmission medium to prisoners. This assumption means that communicators have to consider some limitations inherent in this technique, such as the text concept, the paper's length, and language. Therefore, we believe that in digital communication systems, several other undefined hypotheses exist that must be considered during the development of a covert communication mechanism.

As depicted in Fig. 1, we assume that Alice and Bob are placed in different geographical locations and intend to exchange covert chats using Embedding "$Emb()$" and Extraction "$Ext()$" functions via messaging platforms. However, it is reasonable to assume that the content of their exchanged messages typically can be monitored by unauthorized parties such as competitors, government agencies, or other adversaries. Once the attacker identifies the existence of H data, (s)he may try to interrupt, denigrate or block the transmission channel. Therefore, Alice attempts to utilize an $Emb()$ to disguise an SM into an innocent appearing CM_H under the encoding control using a secret key (SK) as well as transmits the CM_H to Bob via a messaging platform without interference by adversarial parties. After receiving a CM_H sent by Alice, Bob employs the corresponding $Ext()$ to decode the covert SM from the H data. In such a scenario, the primary goal of the covert communication service is to guarantee security and prevent the discovery of the users' hidden chats from a variety of detection techniques (e.g., Detection Function ("$D_f()$") [19].

Steganography can potentially be applied to facilitate covert communication service over the application layer of network channels [21], but this is a very challenging task in the form of a short text message due to the limited number of letters/words [11], the character encoding standard employed and other factors. Moreover, an attacker can perform well-known *steganalysis* mechanisms for monitoring or detecting the existence of covert data through the transferred contents over the transmission channels [12].

2.2. Performance criteria requirements

Given the covert communication scenario (see Fig. 1), there exist multiple experimental challenges in designing a provably secure and efficient system that relies on criteria requirements for transferring an SM undercover of a CM_H. In the literature, cybersecurity experts have described some general criteria [10,12,16], such as hidden capacity and imperceptibility, which partially cover the evaluation metrics of the covert communication systems to some extent. Considering these criteria, we define four features for assessing the performance of text steganographic algorithms, which are referred to as (1) embedding capacity (EC), (2) imperceptibility, (3) distortion robustness (DR), and (4) Security. Below, we describe these evaluation criteria.

- **Payload or Embedding Capacity (EC)** is the number of SM letters/bits that can be concealed in a CM using the $Emb()$ of a given method. This feature is computed considering the bit(s)-per-embeddable-locations (BPL) inside a CM, which is limited to the number of letters (l) or words (w) in it [7]. Let us suppose that the CM= $\{l_1, l_2, \ldots, l_{nl}\}$ or CM=$\{w_1, w_2, \ldots, w_{nw}\}$ in which words/characters can be considered as embeddable locations into the CM such as after each sentence, between words, or substitution of synonyms. The number of Embeddable (E) bits inside a CM using an algorithm, can be computed by the following formula.

$$E_{l\|w} = \left(\sum_{i=i|k}^{nw} BPL_i + \sum_{j=j|k}^{nl} BPL_j \right). \quad (1)$$

where, $1 < i$ or $j : \exists_{k \in ES} \leq nl$ or nw, and the RE is the required embeddable bits for hiding an SM inside a CM. Therefore, the EC_s score can be computed as follows.

$$EC_s = \left(\frac{E_{l\|w}}{RE}\right), \; 0 < EC_s \leq 1. \quad (2)$$

- **Imperceptibility or Invisibility** is a measure of assessing the embedding trace of an SM into a CM_H, which should be imperceptible to the attacker's $D_f(CM_H)$. In other words, this measure

analyzes any alterations that occurred through the CM_H after embedding an SM. According to [19,22], the best way of assessing the imperceptibility rate by the $D_f(CM_H)$ is to calculate the variation of CM and CM_H statistically using the $Jaro-Winkler$ distance. The Jaro distance (J_s) for two given textual strings $L_1 = Length(CM)$ and $L_2 = Length(CM_H)$ is:

$$J_s = \begin{cases} \frac{1}{3}\left(\frac{m}{|L_1|} + \frac{m}{|L_2|} + \frac{m-t}{|m|}\right) & : if(m > 0) \\ 0, & : else \end{cases} \quad (3)$$

Where m is the number of identical characters, and t is the number of transpositions that is equal to half the number of not match spots in the concatenated strings of all matching symbols in the order of their initial occurrence. Two letters from CM and CM_H, respectively, are assumed to be identical only if they match and are not larger than $TF = \lfloor \frac{Max(|L_1|,|L_2|)}{2} \rfloor - 1$. A letter of CM corresponds at most with one character of the CM_H, while chosen by picking the first entrant during a nested iteration over the two strings above. In other words, the identical characters are the equivalent letters with a maximum distance of the TF. Contrary to common assumptions, t is not equal to the number of permutations required to align the sequence of the identical symbols. The Jaro–Winkler similarity $JW_s(L_1, L_2)$ employs a prefix scale (p) to rank strings that match from the beginning more favorably for a given prefix length (υ). Moreover, the JW_s consists of a boost threshold (b_t) for identical prefixes to high J_s values:

$$JW_s = \begin{cases} J_s + [\upsilon.p.(1 - J_S)] & : if(J_s \geq b_t) \\ J_s(L_1, L_2) & : else \end{cases} \quad (4)$$

where υ is the length of the common prefix for two input strings up to a L_{bound} maximum value. Then, $L_{bound} \times p \leq \upsilon$ must holds true, and $0 < JW_s \leq 1$, i.e., the default values for $p = 0.1$, $L_{bound} = 4$, and $b_t = 0.7$.

- **Distortion Robustness (DR)** is the likelihood that the H data will be removed from a CM_H while it is being transmitted over messaging platforms and may be susceptible to attacks that could demolish the H data. Since attackers may monitor the transmission channel, they can impose intentional or unintentional modifications to the CM_H. Hence, an efficient steganography method must make the H data extremely hard to modify before transferring it to the receiver through the network channel. This measure can be computed numerically by determining loss probability (L_p), which is the rate of how many embedded spots (ES), including H data, are eliminated from the CM_H [19]. When an adversary manipulates the CM_H, the H data may not be discovered on the receiver side. The lower rate of L_p causes a higher rate of DR. Let us assume that there exists some letters (nl) or words (nw) in the CM_H, and then the DR_s can be obtained as follow.

$$L_p = \frac{ES}{nl} + \frac{ES}{nw}. \quad (5)$$

where, $1 < ES, nw < nl, \{ES, nw, nl\} \in \mathbb{N}$.

$$DR_s = [1 - L_p],\ 0 < DR_s \leq 1. \quad (6)$$

- **Security** involves analyzing the performance of a covert communication system given the above requirements. Therefore, we utilize the sample (S) standard deviation formula for assessing the loss of efficiency considering the above three criteria scores. According to [23], the usage of S deviation is more efficient than the standard one if the sample size (number of criteria) exceeds 2.

$$S = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(X_i - \bar{X})^2}. \quad (7)$$

where, $X_i = \{EC_s, JW_s, DR_s\}$, N is the number of criteria scores. Therefore, $\bar{X}$ is the sample mean of these three criteria that can be calculated as follows:

$$\bar{X} = \left(\frac{1}{3}\sum_{i=1}^{3} X_i\right). \quad (8)$$

Note that a low S implies that the values tend to be close to the sample mean of the set of numbers, while a high value indicates that the values are spread out over a broader range.

2.3. Unicode text processing

In general, digital systems process and display textual contents using an encoding standard such as Unicode, ANSI, and ASCII, as well as store/share such information in various formats such as SMS, social media post, Email, DOCX, HTML, and PDF [3]. According to the latest version of the Unicode v13.0.0 in March 2020, it comprises three types: Unicode Transformation Format (UTF)-8, UTF-16, and UTF-32 which support a set of 143,859 characters, including 154 modern/historical scripts and several new emojis. Since 2009, the WHATWG [24] established the UTF-8 variant as the primary character encoding for the World Wide Web and announced it as mandatory for all digital textual symbols in software systems. For example, according to a statistical report assessed by the W3Techs in June 2021 [25], an average of 96.9% of existing websites perform the UTF-8 for processing textual contents. Among the above Unicode symbols, some are invisible and reserved for altering visual glyphs of some letters in particular languages, called Zero-Width Characters (ZWCs) [19], i.e., they change the shape of letters in Arabic, Persian, and Urdu languages if they occur between two characters. Furthermore, there exist different forms of Unicode spaces, such as Thin, Hair, or Tab, which function by displaying transparent visual traces with various lengths between words through textual contents [13]. In addition to these symbols, the UTF-8 includes 16 homoglyphs for processing the Latin-alphabet languages. A homoglyph comprises two or more glyphs of a given alphabet that have slightly different shapes with different encoding numbers but appear nearly identical to the human eye (e.g., 'M'=[U+004D] vs. 'M'=[U+216F]) [11]. Several existing approaches have utilized these invisible Unicode features in literature to hide the existence of SM bits inside the CM_H [10–13,26].

3. Related works

Existing methods for covert communication using text steganography can be broadly categorized into three types: coverless, structural, and linguistic-based algorithms.

- **Coverless-based algorithms** involve composing an automatic CM_H using machine learning algorithms based on the SM bitstream to hide it. In practice, these approaches function based on linguistic features such as requiring a dictionary of words and grammar rules, which restrict their application to monolingual text messages [12]. For example, a new steganography scheme called RNN-Stega presented in [7] exploits the recurrent neural networks to automatically generate carrier sentences based on the SM bits. Later in [8], they extended their previous work named VAE-Stega using a variational auto-Encoder to improve the quality of automatic CM_H generation and hidden capacity. In addition, Li et al. [27] have suggested a neural-linguistic steganography model based on knowledge graphs called Topic-Aware, which automatically creates a paragraph of several sentences regarding a unique topic to conceal an SM. In this work, the authors improved the quality of generated stegotext compared to the RNN-Stega [7] concerning the perplexity rate. In practice, these techniques are limited to providing covert communication services under the guise of multi-sentence English text contents.

- Structural-based methods adjust the layout structures of CM, such as spaces between words/lines and the font type for disguising an SM. Such formatting features depend on character encoding (e.g., UTF-8), which does not change the content of the CM_H after embedding the H data into the CM. However, they may increase the length of the CM_H statistically, which may raise suspicions of adversaries during the steganalysis process. For example, Ahvanooey et al. [3] proposed an innovative steganographic technique called AITSteg, which generates pairs of numbers using the Gödel function based on ASCII codes of SM letters (only English alphabets) as well as encrypts the pairs of numbers employing an automatic key generation that considers the sending/receiving time and length of SM. Then, it embeds the encrypted SM bits by appending a group of Unicode ZWCs at the beginning of CM. Wu and Hsu [13] have presented another information hiding algorithm called LINE-Chat, which merges the normal space with other Unicode spaces (e.g., Ideographic and Tab), considering each 2-bit of SM bits. Then, this approach inserts the generated H string of two/three spaces between words to embed an SM into a text file of the text-chat history as a CM. Another space-based approach proposed by Khosravi et al. [26] inserts an extra standard space between words for hiding one bit of SM bits inside a PDF cover text. In practice, these space-based approaches generate lengthy gaps between words or lines that raise the suspicion level of adversaries concerning this unusual formatting. In [10], authors have introduced a novel watermarking algorithm called Homoglyphs that hides a watermark inside a Latin-based cover text file by replacing Unicode glyphs of letters and unique spaces between words for content authentication. Later in [11], they enhanced their previous work in Rizzo et al. [10] by including a Huffman compression algorithm to enhance the hidden capacity, which provides covert communication via social media. Since these methods employ the Unicode homoglyphs (1-bit per glyph) and unique spaces (3-bit per each one) for disguising the SM bits inside the CM, the embedded spots throughout the CM_H are slightly more perceptible, which can raise the attention of adversaries. In addition to generating unusual gaps between words, the visual imperceptibility of homoglyphs depends on the font typeset as formatting for the CM_H. Another font-based algorithm called FontCode, presented by Xiao et al. [14] alters the glyphs of letters inside the CM to conceal the SM letters by changing their font type (Times New Roman and Helvetica). In practice, such font-based approaches depend on the formatting settings of the transmission channel, which limits their application in particular platforms that already support these font types.
- **Linguistic-based approaches** alter the conceptual features of a CM for concealing an SM by substituting some composition elements such as abbreviations and synonyms inside the CM_H. These algorithms can be categorized into two different types: semantic and syntactic. Semantic techniques work based on amending the CM content according to particular features in a language, such as synonyms, spelling, or acronyms. Syntactic methods modify the CM content without manipulating its original meaning by replacing similar words with the same meaning in a particular language. The primary merit of linguistic-based mechanisms is to protect the embedded hidden message against retyping activities. For example, Chang and Clark [9] have proposed a graph-based steganography technique called Graph-Steg, which exploits a vertex coding strategy for substituting synonyms inside a CM and assigning a unique pair of SM bits (e.g., 00, 01, 10, or 11) to each replaced word. Another synonym-based steganography approach presented in [28] hides one bit of the SM bits per synonym of words (e.g., love='0' and Like='1'). Since these techniques substitute a limited number of synonyms inside the CM, in practice, they provide an insufficient payload for embedding the SM bits in short text messages.

Technically, short text messages consist of limited linguistic and structural features, restricting the practical applications of state-of-the-art concealment methods and limiting their efficiency. Such constraints include linguistic elements such as the length of the CM content (words or letters), language, and formatting settings (e.g., character encoding, font, and color). In addition, among the described approaches, some studies [7–10,14,27] have not discussed the details of SK exchange, and some other approaches utilize the random key generation [12,13]. Therefore, such techniques do not comply with Kerckhoffs's principle of cryptography since the SK could be exposed to adversaries if (s)he acquires access to the details of the algorithm. Because a steganographic approach employs an SK during the encoding/decoding phase, similar to cryptography, it must adhere to Kerckhoffs's principle for exchanging the SK between Alice and Bob. Hence, an efficient mechanism should be secure even if everything about the algorithm is publicly revealed, except for the SK [8,29], which means that it must guarantee a safe way of SK exchange, considering Kerckhoffs's principle and provide an optimum trade-off between the above criteria requirements.

4. Cyberattack scenarios

In this section, we describe how adversaries may design cyberattacks to detect the existence of H data through the CM_H. To encounter text steganography mechanisms, attackers design steganalysis models to identify whether a given CM_H contains covert information and, if possible, eliminate or decode the embedded SM [12]. For example, Luo et al. [30] have proposed a deep learning-based text steganalysis model for extracting linguistic syntactic and semantic features from business data by focusing on the word-based feature extraction. Another text steganalysis method introduced by Yang et al. [31] predicts the correlations between words from stegotext by mapping each word to a semantic space. In another research work, Yang et al. [32] have presented a linguistic-based steganalysis model using the recurrent neural network, which employs conditional probability distributions of each word for detecting the hidden information from carrier texts. In practice, these approaches analyze the CM_H utilizing word embedding analysis, which might only detect the existence of hidden information embedded using linguistic-based methods [7–9,28]. However, these algorithms lose their detection performance for carrier texts generated using structural-based methods. In other words, steganalysis methods function in two strategic forms: specific and universal. Whereas the steganalysis approach focuses on breaking a particular scheme, the universal method targets a set of mechanisms. Indeed, the steganalysis techniques could identify the existence of H data from the CM_H to some extent; however, decoding the SM is an impossible task without having the original SK according to Kerckhoffs's principle [12].

While a CM_H is shared on network channels, it is vulnerable to passive/active attacks that may perform the above intelligent steganalysis models to identify measurable features between standard text message and stegotext in terms of imperceptibility [33]. Note that, the primary goal of text steganography is to decrease the statistical distribution difference among these two text strings. According to [8], text imperceptibility can be classified into two types: perceptual-imperceptibility and statistical-imperceptibility. The first concept quantifies the visual quality of CM_H compared to CM using an image similarity analysis technique [11]. In contrast, statistical imperceptibility analyzes the inseparability rate between CM and CM_H using Eq. (4) [19]. As depicted in Fig. 1, let us assume that an attacker processes the communication channel between two parties to obtain secret knowledge through their conversations. Considering this assumption, we can classify the possible attack scenarios into three types: Man-in-the-Middle (MitM), Advanced Persistent Threat (APT), and Service Provider Operators (SPO), which are briefly summarized below.

- **MitM**: This attack intercepts the communication channel and eavesdrops on the transmitted data between two parties by processing the traffic based on the properties of TCP and HTTP protocols, which utilize the Unicode standard for text processing. In this attack, adversaries craft a text steganalysis method to quantify the statistical variation through a CM_H by considering any possible combinations or non-alphabetic characters [12,34]. For instance, (s)he can perform a $D_f(CM_H)$ to discover the existence of H data from the letters or words using Eq. (4) [35] for identifying the covert chats between the communicating parties. Note that the steganalysis only allows the detection of abnormal changes through the CM_H. However, in a case that the SK is neither generated nor saved/shared by the corresponding algorithm, then adversaries cannot extract/decode the original SM from the H data even if (s)he has complete knowledge about the details of the steganography approach [33]. Also, if an attacker discovers any abnormal H data through the transferred conversations between Alice and Bob, then (s)he may still take some practical actions, such as blocking the network channel or reporting their activities to the authorities.
- **APT**: In this specific type of cyberattack, an adversary (or a group) may attempt to gain confidential information from the presence of spyware on smartphones and maintain a long-term presence within susceptible organizational targets [36]. In other words, the APT attacker aims to collect sensitive knowledge by performing social engineering activities on victim devices, such as executing spyware to capture communicated conversations or data via email, SMS, and other messaging apps installed on them. The primary characteristics of such cyberattacks can be featured as: *(i)* Adapting information collection strategies when an event does not occur as planned; *(ii)* Persistent willingness to wait until the subject intelligence is achieved; *(iii)* Difficulty in identifying APT existence due to having extensive knowledge about the technical operation of the target device. APT smartphone attacks can be viewed as complex data leak threats that function covertly through because these devices are technically complex and involve sensors and embedded devices (e.g., microphone, GPS, or camera) for collecting, controlling, and managing information [37].
- **SPO**: In this attack, employees or vendors working for the data center of Internet service providers or messaging platforms (e.g., third-party companies such as Google, Facebook, or Tencent) have access to stored conversations and data through their database and communications servers. In practice, these platforms may provide various means of surveillance or interference by entities such as law enforcement agencies or national security services [38]. They can also track users' exact locations using the geolocation services of smartphones [4,39]. Let us assume that an SPO act as an active malicious user who manipulates the stored conversations (e.g., CM_H) through their database servers to mislead one of the communicating parties. In such a case, the DR rate of an approach can be calculated using Eq. (6).

5. The proposed covert communication scheme

In this section, we describe a motivating scenario and the proposed technique in details

5.1. Motivating scenario

Since most Internet users utilize social networks to exchange their daily conversations, which might contain sensitive knowledge about their business, attackers can run one of the above-described attacks to collect the users' exchanged messages. For instance, let us assume that Alice and Bob are brokers working for a brokerage firm whose customers can sell/buy stocks in an account through an online trading system. They intend to expose confidential information by exchanging text chats via messaging platforms such as SMS, WhatsApp, and WeChat. As these users share influential knowledge about the investment on a particular stock that may be worth millions of dollars in some cases, such conversations are targeted by attackers. Since the messaging platforms utilize standard protocols (e.g., HTTP(S) and TCP) for transferring text messages, the conversations between the communicating parties are susceptible to the above stated cyberattacks. As depicted in Fig. 2, to guarantee the security and privacy of such conversations, we propose a systematic covert communication approach called CovertSYS, which provides secure end-to-end conversations via messaging platforms under the setting controls of users' credentials (e.g., password and biometric key hereafter refereed to Bio-key). The CovertSYS processes a multilingual confidential message based on Unicode and creates and encrypts the SM bits using a multi-factor authentication scheme. Then, the encrypted SM bits are converted to a sequence of Unicode ZWCs using a text steganography technique for transferring it under the guise of a short CM. In the CovertSYS technique, Alice and Bob can utilize any language and define a password combination for encoding confidential conversations and keeping them safe from the attacks mentioned above. The proposed technique consists of three different functions, namely: Embedding "$Emb()$", One-time-Pad "$OTP()$", and Extraction "$Ext()$". Below, we explain the implementation steps of these functions in detail.

5.2. Embedding function

Where Alice intends to initiate a secure end-to-end conversation via third-party messaging platforms, she can utilize the CovertSYS scheme to disguise the confidential information to be exfiltrated under cover of a short CM. She must designate a language choice and set a password combination with the proviso that Bob is already aware of them. During the setting stage, the password combination is saved as a hashed value using the Secure Hash Algorithm 3 (SHA-3) inside the users' Bio-keys, which is not stored on users' devices or exchanged on the transmission channel. When Alice executes the CovertSYS for covertly transmitting an SM through an innocent CM, the $Emb()$ authenticates Alice's password using the U2F Bio-key, and if the access is granted, then it implements the following steps (see Algorithm 1 and Algorithm 2):

Step (1): It generates a Secret Data (SD) string by designating a sending time (ST) (e.g., 02:51="0251"), and a sending Date (D) (e.g., 16/10/2020= "16102020") as well as combining these variables with the SM string to be embedded into the CM. In addition to the Bio-key, these two factors will be used for authenticating the SD during the extraction process.

Step (2): It converts each letter of the SD to an 8-bit string for the English characters and a 16-bit string for each Chinese character according to UTF-8 encoding (note that it can be adapted to support any language that Unicode covers under different character encoding types). Finally, it generates a Secret Data Binary (SDB) string by combining all generated 8-bit or (16-bit) strings.

Step (3): Using an evolutionary algorithm, it performs the One-Time-Pad algorithm for composing a Block Cipher (BC) based on the SK Binary (SKB) string as the third factor. Then, it reproduces the BC by duplicating it $N = [\frac{Length(SDB)}{Length(SKB)}]+1$ times until the length of the BC is equal to the length of the SDB string. Next, it eliminates the additional bits from the other side (left/right side of the BC), considering the modulo operation result by 2 to identify the even or odd length of the SD string (see Algorithm 2). The OTP is a well-known unbreakable cryptosystem [40].

Step (4): It encrypts the SDB string using the exclusive OR (XOR) based on pairs of bits in the BC and generates the Encrypted SDB (ESDB). This operation guarantees the security of the CovertSYS according to Kerckhoffs's principle. It implies that the adversary cannot extract the SM even if (s)he knows the details of the $Emb()$ without

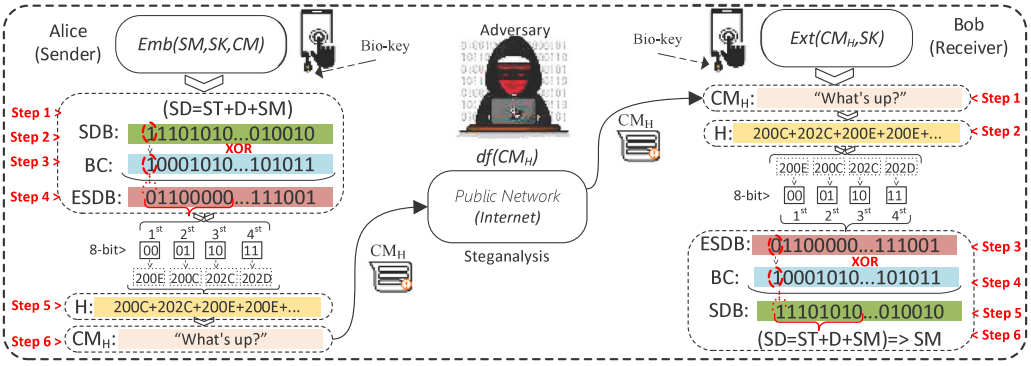


Fig. 2. The implementation steps of the CovertSYS scheme.

Algorithm 1 : Pseudo-code of $Emb()$

Input: Cover Message (CM), Secret Message (SM), Password Combination (PC);
Output: Carrier Message (CM_H) which is composed of CM and H string;

$CM \leftarrow$ *Cover Message;*
$SM \leftarrow$ *Secret Message;*
$PC \leftarrow$ *Password Combination;*
if ($PC.HashedValue() = Bio\text{-}key.HashedValue()$) **then** // *Verifying the PC based on the U2F Bio-Key*;
  $VPC \leftarrow PC$;
  $ST \leftarrow$ *Generates a 4-digit number based on the current time;*
  $D \leftarrow$ *Creates a 8-digit number based on the current date;*
  $SD \leftarrow$ *Composes a secret data string by combining the ST+D+SM strings;*
  $UTF8SD \leftarrow$ *Converts each letter of SD to Unicode character using StandardCharsets.UTF-8;*
    for each $C_i \in UTF8SD\{C_1, C_2, ..., C_{nl}\}$ **do**
    $SDB \leftarrow$ *Converts each* C_i *to a 8-bit or 16-bit string considering the language;*
  end for
  $ESDB \leftarrow$ *Encrypts the SDB by performing the OTP(SDB,VPC);* // (check Algorithm 2)
    for each $ZWC[i] \in BTA\{u_{00}, u_{01}, u_{10}, u_{11}\}$ **do**
    $H \leftarrow H + ZWC[i]$;
  end for
  $CM_H \leftarrow CM+H$;
  $SR \leftarrow$ *"The SM has been embedded at the end of* CM_H*"*;
  else
  $SR \leftarrow$ *"The password combination does not match the stored users' credentials through the Bio-key!"*
  end if
Return CM_H, SR;

having the Bio-key and the SK, i.e., the defined password combination by Alice and Bob.

Step (5): It divides the ESDB string into 2-bit sequences and replaces each with a ZWC based on the predetermined pattern binary tree (see Fig. 3) for constructing the H string [41].

Step (6): It adds the H string at the end of the CM (e.g., "What's Up?") and creates the CM_H, which can be sent by the messaging platforms, particularly those support the Unicode standard for text processing. Surprisingly, the H string is visually invisible for observers who may run the APT or SPO attacks on the CM_H. However, the existence of H strings (ZWCs) can be statistically processed if the adversary performs a MitM to monitor the transmission channel between two parties.

5.3. Extraction function

Once Bob receives the CM_H, he can utilize the CovertSYS to extract the confidential conversation from it. The $Ext()$ authenticates Bob's password using the U2F Bio-key, and if the access is granted, then it performs the following steps for decoding the original SM from the H string.

Step (1): It processes the CM_H and extracts the ZWCs to generate the ESDB string.

Step (2): It replaces each ZWC of the H string with a 2-bit in the ESDB according to the predetermined pattern in the binary tree (See Fig. 3).

Step (3): It performs the $OTP()$ algorithm for decrypting the ESDB based on the SK.

Step (4): The OTP generates the BC to decode the ESDB string by performing the exclusive-OR on pairs of bits.

Step (5): It converts each 8-bit of the decrypted SDB to its corresponding letter based on the UTF-8 encoding for English alphabets (or 16-bit for Chinese letters).

Step (6): It authenticates the validity of the SM by calculating the receiving time (RT) and date (RD) at the receiver and compares them with the sender's values extracted from the CM_H to discover the access delay $T(s, r)$ between two parties. If there is no more than 2 min delay (configurable), it allows extracting the H string and decoding the SM from it. Otherwise, the CM_H will be considered an invalid stegotext. In practice, this strategy generates various H strings for even the same SM in different sending/receiving periods, which makes the dynamic entropy permutation against any predictive steganalysis models.

Algorithm 2 : Pseudo-code of *OTP(String BS, String SK)*

Input: Binary String (BS), Secret Key (SK);
Output: Xored Binary String (XBS) which is the encrypted/decrypted version of BS using the SK;

BS ← *Binary String;*
SK ← *Secret Key; // Bob/Alice's password combination.*
SKB ← *Converts the SK letters to binary string based on the UTF-8 encoding numbers;*
$N \leftarrow [Length(BS)/Length(SKB)]$;
if $(Mod(Length(BS), Length(SKB)) == 0)$ **then**
  $N \leftarrow N + 0$;
  else
  $N \leftarrow N + 1$;
  end if
while $(N >= 1)$ **do**
$BC \leftarrow BC + SKB$;
$N \leftarrow N - 1$;
end while
if $(Length(BC) > Length(BS))$ **then**
  $T \leftarrow Length(BC) - Length(BS)$;
  else
  $T \leftarrow 0$;
  end if
if $(T! = 0)$ && $(Mod(Length(BC), 2) == 0))$ **then**
  $BC \leftarrow BC.Substring(0, T)$;
  else
  $BC \leftarrow BC.Substring([Length(BC) - T], T)$;
  end if
XBS ← XOR(Computes exclusive OR for each bit in BS based on its pair in BC);
Return XBS;

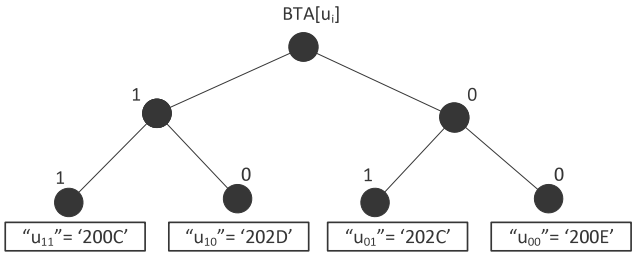


Fig. 3. Predetermined pattern based on binary tree for creating H string.

6. Experimental analysis

In this section, we evaluate the efficiency of the CovertSYS scheme by considering our criteria requirements. To verify the efficiency and covertness of the proposed technique, we designed and implemented a proof-of-concept app using Java in the form of Android and Windows software. Using benchmark examples, we experimented with the CovertSYS app and evaluated the performance criteria.

6.1. Covert communication criteria analysis

The following points discuss the performance features of the CovertSYS concerning criteria requirements according to the example listed in Table 2.

- **Payload or Embedding Capacity (EC):** As we already pragmatically explained in Algorithm 1, the *Emb*() adds a ZWC in the H string for embedding each pair of SM bits into the CM_H, if a messaging platform is employed as a transmission channel between two parties. The CovertSYS affords the EC by observing the max number of characters limit in the corresponding app. For example, SMS limits a message size to a maximum length of 1024 characters for the UCS-2 (16-bit) and 2048 letters for the GSM-7 (8-bit). We assessed all the maximum text limits and the number of embeddable characters considering the language type (e.g., four ZWCs for embedding each English letter and eight ZWCs for each Chinese character) that the CovertSYS can covertly transmit over seventeen out of twenty popular platforms. Note that the length of CM is subtracted from the number of embeddable letters in each platform to compute the exact EC_s rate using Eq. (2). As depicted in Table 3, we summarized the maximum character limit of each messaging platform according to their official websites and experimented with the EC and visual-imperceptibility of the CovertSYS by sending a CM_H via these apps. To compute the EC_s for the benchmark sample in Table 2, we utilize Eq. (2) considering the $RE = 25 \times 8 - bit = 200$ bits, and the CovertSYS embeds 100 ZWCs for hiding the SM at the end of the CM; then, the $EC_s = \frac{200}{200} = 1 = 100\%$. Note that the CovertSYS is the extended version of AITSteg [3], which provides a higher maximum EC through the transmission channel by embedding four ZWCs to hide each English character, i.e., AITSteg requires six ZWCs to disguise the same letter. This feature changes the statistical imperceptibility as well.
- **Imperceptibility:** As listed in Table 2, the CovertSYS does not modify the written symbols of the CM_H after appending the H string because the ZWCs are used to disguise the H string inside the CM. To evaluate visual imperceptibility, we have tested the CM_H by transferring it via messaging platforms (see Table 3) on operating systems such as Android, iOS, and Windows. Our empirical experiments showed that APT and SPO-based attacks could not view the H string and facilitate its discovery by the human eye. Based on our empirical results, seventeen platforms supported the Unicode ZWCs and allowed transmitting the CM_H with invisible visual imperceptibility (no trace), so an observer could only see the CM. Note that three platforms – namely Skype, Twitter, and Telegram – do not permit the transmission of the H string since they employ exclusive formatting settings or character encodings for text processing. We then calculated the statistical imperceptibility between the CM and the CM_H for the example

Algorithm 3 : Pseudo-code of $Ext()$

Input: Carrier Message (CM_H), Password Combination (PC);
Output: Secret Message (SM) which is authenticated based on the T(s,r);

$CM_H \leftarrow$ *Carrier Message;*
$PC \leftarrow$ *Password Combination;*
if (*PC.HashedValue() = Bio-key.HashedValue()*) **then** // *Verifying the PC based on the U2F Bio-Key*;
  $VPC \leftarrow PC$;
    for each $ZWC_i \in CM_H\{u_{200E}, u_{202C}, u_{202D}, u_{200C}\}$ **do**
    $ESDB \leftarrow ESDB+$ *Converts each* ZWC_i *to a* $2-$ *bit string*;
  end for
  $SDB \leftarrow$ *Decrypts the ESDB by performing the OTP(ESDB,VPC) (check Algorithm 2);*
  if ($Language =$*"English"*) **then**
    $T_{bit} \leftarrow 8$;
    else if ($Language =$*"Chinese"*)
    $T_{bit} \leftarrow 16$;
    end if
    While ($Length(SDB) >= T_{bit}$) **do**
    $BSTR \leftarrow SDB.Substring(0, T_{bit})$;
    $SDB \leftarrow SDB.Substring(T_{bit})$;
    $SD \leftarrow SD+$ *Converts each BSTR (8-bit or 16-bit) string to it's Unicode letter;*
  end while
  $ST \leftarrow SD.Substring(0,4)$; // Alice's sending time.
  $D \leftarrow SD.Substring(4,8)$; // Alice's sending date.
  $RT \leftarrow$ *Generates a 4-digit number based on the current time*;
  $RD \leftarrow$ *Constructs a 8-digit number based on the current date*;
  $SD \leftarrow$ *SD.Substring(12)*;
  if ($RD == D$) && ($RT <= (ST + 2)$) **then**
    $SM \leftarrow SD$;
    $SR \leftarrow$ *"The SM has been authenticated!"*;
    else
    $SR \leftarrow$ *"The SM is not valid!"*;
    end if
  else
  $SR \leftarrow$ *"The password combination does not match the stored users' credentials through the Bio-key!"*
  end if
Return SM, SR;

in Table 2 using Eq. (4) as a steganalysis methodology which the $JW_s = 0,764 = 76.4\%$.

- **Distortion Robustness (DR)**: Since the $Emb()$ increases the content of CM_H statistically, we considered the embedding spot for hiding the H string at the end of CM. This insertion point can reduce the DR_s, which may lead to a lower probability of the H string being destroyed by attackers. Here, the DR_s can be calculated using Eq. (6). For instance, depicted in Table 2, the $nl = Length(CM)$ and the $es = 1$. Then the L_p for this CM_H sample is: $L_p = \frac{1}{56} \cong 0,017$. Thereby, $DR_s = [1-0,017] \cong 0.982 = 98.2\%$. It indicates that if an adversary alters a part of the CM_H (except the end of CM), then the H string remains inside the CM_H and could be extracted using the $Ext()$. Nevertheless, the existence of ZWCs can be processed statistically using a steganalysis approach, and knowledgeable adversaries may attempt to decode the Unicode letters of the H string. Since CovertSYS functions based on a dynamic encoding pattern and OTP encryption algorithm, it is impossible to decode the SM from the ZWCs for attackers without having the users' password and Bio-key, which adheres to the Kerckhoffs's principle of cryptography.
- **Security:** We utilize this measure for assessing the performance of the proposed technique considering the scores of three criteria (e.g., $EC_s = 100$, $JW_s = 76.4$, and $DR_s = 98.2$) using Eq. (8). Thereby, the $Mean\ \bar{X} = 91.53\%$ and $S = 13.13$. To show the performance of the CovertSYS, we compare these obtained numbers with the results of the state-of-the-art approaches in the following subsection.

6.2. Comparative analysis

As a methodology for constructing a fair comparative analysis, we evaluate the state-of-the-art methods considering the same SM and CM_H sample in Table 2 as well as calculate the payload (or EC) rate according to the ES through the corresponding CM_H for each approach. As shown in Fig. 4, we suppose that the sample SM is embedded into the CM using the mentioned methods, where each English letter needs an 8-bit string RE to be embedded inside the CM according to the UTF-8 encoding. Table 4 lists the obtained criteria scores based on the CM_H examples in Fig. 4 using the Eqs. (2)–(8).

As shown in Table 4, because the S value of the CovertSYS is lower than the other algorithms, the standard deviation proves that S values tend to be close to the sample mean according to the definition. Moreover, we can see that the $Mean\bar{X}$ of the CovertSYS is higher than the other approaches. Therefore, the proposed technique outperforms the state-of-the-art methods when applied to short text messages. Fig. 5(a) shows that only CovertSYS, AITSteg, and FontCode can embed the $RE = 200$ bits for hiding the sample SM in the CM, and other methods have achieved a score lower than the red line threshold for affording the RE bits. Fig. 5(b) depicts the ternary plot by computing the trade-off between three features: statistical imperceptibility, EC, and DR scores for each method according to the values in Table 4, where the more the point is near to the center of the plot, the more optimum performance between three features achieved. As depicted in Table 5, we rated the performance features by assigning an empirical level (high, medium, and low) for each method considering the calculated scores from the CM_H examples in Fig. 4. Moreover, we summarized the advantages and

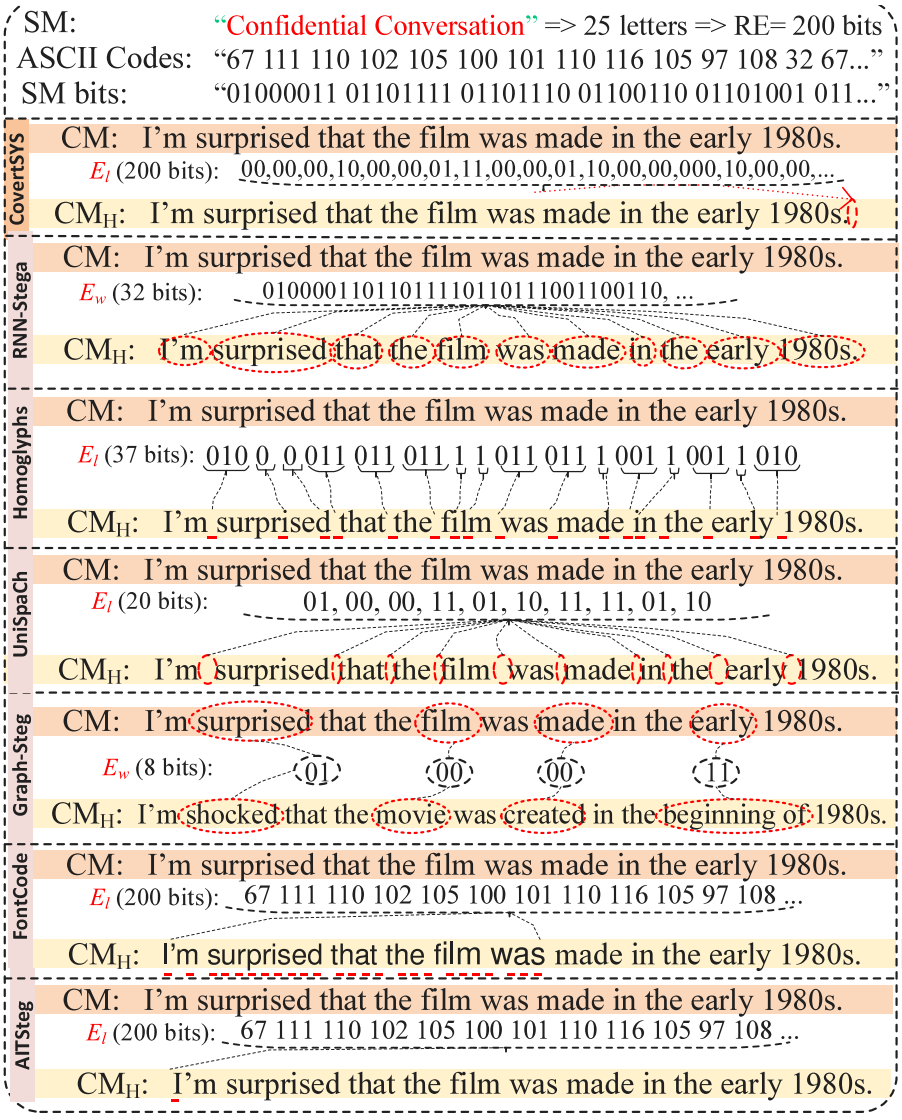
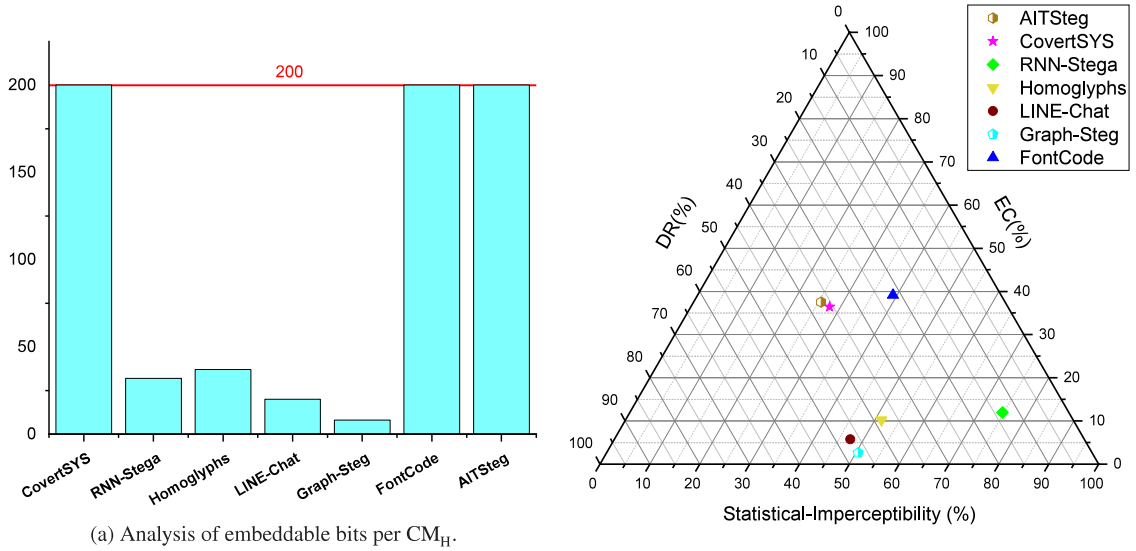


Fig. 4. An illustration of embedded spots of aforementioned techniques on the same SM and CM example.



(a) Analysis of embeddable bits per CM_H.

(b) Trade-off analysis between performance features.

Fig. 5. Performance analysis of the proposed technique and evaluated methods according to examples in Fig. 4 considering the three criteria scores.

Table 2
An example of the embedding process in detail.

Variables	Values
SM	“Confidential Conversation”
Length of (SM)	25 letters ⇒ $RE = 200$ SM bits
CM	“I’m surprised that the film was made in the early 1980s.”
Length of (CM)	56 letters
ST	22:12
D	31/07/2021
SD	“221231072021Confidential Conversation”
PC⇒SK	“512#&M”
SKB	“00110110001100100011000000110000000110101”
BC	“001100000011010100110111001100110011000000110000001101010011011100110 01100110000001100000011010100110111001100110011000000110000001101010011 01110011001100110000001100000011010100110111001100110011000000110000001 1010100110111001100110011000000110000001101010011011100110011001100000 011000000110101”
SDB	“001100100011001000110001001100100011001100110001001100000011011100110 01000110000001100100011000101000011011011110110111001100110011010010110 01000110010101101110011101000110100101100001011011000010000000100001101 10111101101110011101100110010101110010011100110110000101110100011010010 110111101101110”
ESDB	“000000100000011100000110000000001000000110000000100000101000000000000 00010000000000000001000000100011101000101110001011110010101100101110001 01001101010110010111100100010001011100010101100101111100010000001110011 01011010010110010100010101010101010000100100011001010110010001110101100 10101111101011011”
Steganalysis of H string	“200E+200E+200E+200C+200E+200E+202C+202D+200E+200E+202C+200C+200E+ 200E+200E+202C+200E+200E+200E+202D+200E+200E+200E+202C+200E+200E+ 202C+202C+200E+200E+200E+200E+200E+200E+200E+202C+200E+200E+200E+ 200E+200E+200E+200E+200C+200E+200E+202C+200E+202C+202D+202C+200E+ 202C+202C+202D+200E+202C+202C+202D+200C+202C+202C+202C+200C+202C +202C+202D+200E+202C+202C+200E+202D+202C+202C+202C+200C+202C+202C +202D+200C+202C+200E+202C+200E+202C+202C+202D+200E+202C+202C+202C +200C+202C+202C+202D+202D+200E+202C+200E+200E+202C+202D+200E+202D +202C+202C+200C+200C+202C+202C+200C+202C+202C+200E+202C+202C+202C +202C+202C+202C+202C+200E+200E+200C+202C+200E+202C+200C+202C+202C +202C+200C+202C+200E+202C+202D+202C+202C+200C+202C+202C+202C+202D +202D+202C+202C+200C+202D”
Visual-Imperceptibility Analysis of H string	“”
Output: CM_H	I’m surprised that the film was made in the early 1980s.

limitations of evaluated approaches in terms of their applications in short text messages. Note that we considered the benchmark sample consisting of an English SM and CM because most of the evaluated methods only apply to English texts.

6.3. Discussion

In this subsection, we discuss the practical implications of the CovertSYS scheme versus the state-of-the-art methods. These implications consist of language compatibility, safety, and prevention against cyberattacks.

6.3.1. Language compatibility

Is CovertSYS applicable to secure multilingual confidential information? Since it utilizes the UTF-8 encoding during the $Emb()/Ext()$ function, it can be used for providing secure end-to-end conversation via any language that the Unicode standard supports in its UTF-8 category. Fig. 5(a) and Table 5 show that three methods can afford sufficient RE bits and only CovertSYS and FontCode offer multilingual applicability. Since the FontCode utilizes two font types for hiding SM bits, it fails to transfer the covert data via messaging platforms that do not support the exploited font types. Moreover, this approach changes the appearance of some characters through the CM_H as shown in Fig. 4 because of the use of two font types for disguising the SM. Also, since the AITSteg utilizes the ASCII codes during the encoding process of the SM via the Gödel function, this method can only be applied to English texts.

6.3.2. Safety

Does CovertSYS provide a provably secure covert communication system? This technique encrypts the SM based on three authentication factors: users’ password, sending time, and sending date using the OTP strategy, which is a proven unbreakable cryptographic approach [40]. We designed an evolutionary algorithm to implement the OTP that generates a BC by reproducing the SK several times. Then, the BC is used during the encryption process with the $OTP()$. This feature produces a dynamic sequence of ZWCs that is entirely variant even for the same SM in different periods. It implies that the CovertSYS adheres to Kerckhoffs’s principle, i.e., even if adversaries discover the details of the CovertSYS, (s)he cannot decode the SM without knowing the users’ password combination. All the state-of-the-art methods did not take Kerckhoffs’s principle into account during their algorithm design, except for the VAE-Stega [8] technique, in which the authors assumed that the adversary has complete knowledge of this approach and did not clarify how to encrypt the SM using the SK, which may leave it vulnerable to knowledgeable and skilled attackers.

6.3.3. Prevention against cyberattacks

Below, we explain how the CovertSYS can prevent the H string of SM from being detected or destroyed through the CM_H against the stated cyberattacks. As depicted in Table 6, we evaluated the $D_f(CM_H)$ for each algorithm where the CM_H is compromised by certain cyberattacks in terms of the embedded spots on the corresponding stegotext.

- ***MitM attack*:** Let us suppose that an adversary intercepts the communication channel between two parties and monitors the

Table 3
Payload and visual imperceptibility analysis of the CovertSYS on the latest version of messaging platforms.

Direct messaging platform	UTF-8 text limit ⇒ characters per message	Maximum EC for English letters	Maximum EC for Chinese letters	Visual-imperceptibility analysis: invisible (✓) or No(×)
SMS	1,024	256	128	✓
WeChat	5,000	1,250	625	✓
QQ	5,000	1,250	625	✓
Instagram	1,000	250	125	✓
Tinder	1,000	250	125	✓
Line	5,000	1,250	625	✓
Imo	5,004	1251	625	✓
Zoom	500	125	62	✓
Weibo	1,000	250	125	✓
Reddit	10,000	2,500	1,250	✓
Viber	1,000	250	125	✓
WhatsApp	4,096	1,024	512	✓
Pinterest	2,000	500	250	✓
Microsoft Teams	2,400	600	300	✓
Tango	5,000	1,250	625	✓
Gmail	52,224	13,056	6,528	✓
LinkedIn	2,000	500	250	✓
Skype	160	–	–	×
Twitter	10,000	–	–	×
Telegram	4,096	–	–	×

Table 4
Performance criteria analysis using Eqs. (2)–(8) according to the highlighted examples in Fig. 4.

Algorithm	$EC_s(\%)$	Statistical-Imperceptibility Steganalysis ⇒ $JW_s(\%)$	$DR_s(\%)$	Mean $\bar{X}$	S
CovertSYS	100	76.4	98.2	91.53	13.13
RNN-Stega [7]	16	100	17.8	44.6	47.98
Homoglyphs [10]	18.5	93	69.6	60.36	38.09
LINE-Chat [13]	10	82.85	82.1	58.31	41.84
Graph-Steg [9]	4	78.3	72.7	51.66	41.37
FontCode [14]	100	100	55.3	84.33	27.13
AITSteg [3]	100	68.2	98.2	88.8	14.58

exchanged CM_H by performing the $D_f(CM_H)$. Note that the MitM attacker can statistically detect the existence of ZWCs only through the CM_H. Since the SK is neither generated nor stored/shared through the network channel, (s)he cannot extract or decode the SM from the H string without possessing the users' password combination and connecting the Bio-key, even if (s)he has sufficient knowledge regarding the details of the proposed technique. This feature guarantees the adherence of CovertSYS with Kerckhoffs's principle. To counter such covert communications, adversaries may block the network channel or report suspicious activities to the proper authorities. Because there are a variety of third-party messaging platforms (e.g., WhatsApp, WeChat, Imo) that Alice and Bob can choose as a transmission channel, the process of blocking such services can be a highly complex challenge. Where adversaries monitor or process the network channel by performing a MitM attack, the linguistic-based and coverless-based methods such as RNN-Stega [7], VAE-Stega [8], Topic-Aware [27], and Graph-Steg [9] could be efficient tools for concealing any traces of covert communication. However, since these approaches suffer from low EC and DR rates, they cannot be applied to short messages.

- ***APT attack***: In the case that an APT attacker installs spyware on Alice's and/or Bob's devices to gain access to confidential information exchanged between two parties, if they utilize the CovertSYS to exchange covert conversations, the adversary can only visually observe the CM content — the H string is entirely imperceptible to the human eye. Although other techniques also protect the covert data from being detected by the APT attacker, the FontCode and Homoglyphs depend on the used font type, which may display unknown embedded symbols if the platform does not support these features. These artifacts give away the fact that there is an anomaly within the short message.
- ***SPO attack***: When an SPO monitors the stored conversations between two parties through their communication servers (e.g., database), and since these servers employ the Unicode standard for text processing, (s)he can only observe the CM content and the H string stays visually invisible within the CM_H. In the case that the SPO manipulates the contents of an exchanged CM_H to mislead Alice/Bob, then the H string is likely to remain safely at the end of the stegotext with a higher DR rate as the CovertSYS's score is greater than the other methods (see Table 4).

6.4. Computational complexity analysis

Since CovertSYS functions based on the $Emb()$ and $Ext()$ algorithms on the communicating parties' devices (e.g., smartphones or computers), these methods have the same computational complexity to be executed as they are implemented recursively on each side (sender or receiver). As depicted in Algorithms 1 and 3, these two functions need a linear search $O(n)$ for converting SM letters to their binary strings (or reverse) and a binary search $O(logn)$ for replacing the 2-bit string by a ZWC according to the predetermined pattern in the binary tree (See Fig. 3). Additionally, to generate a BC, the CovertSYS must reproduce it $O(n)$ times using the evolutionary algorithm as the shared key as well as there is a call to execute Algorithm 2 for encrypting/decrypting the SM bits using the OTP(), which requires $O(n)$. Therefore, the time complexity of the CovertSYS is equal to $O(nlogn)$ in the best-case scenario. As listed in Table 7, we evaluated the computational complexity of the state-of-the-art methods compared to the CovertSYS considering the n, which is the number of SM letters required to be embedded inside the CM.

Table 5
Performance characteristics of the CovertSYS vs the state-of-the-art approaches considering their applications in short text messages.

Algorithm	Mechanism Type	Language Compatibility	Imperceptibility Yes:(✓)/No:(×)	EC	DR	Advantages (+)/Limitations (–)
CovertSYS	Structural	Multilingual	Visual: ✓ Statistical: ×	High	High	+ High EC and high DR + High visual-imperceptibility + Multilingual support + Adherence to Kerckhoffs's principle – Medium statistical-imperceptibility
AITSteg [3]	Structural	English	Visual: ✓ Statistical: ×	High	High	+ High EC and high DR + High visual-imperceptibility – No adherence to Kerckhoffs's principle – Medium statistical-imperceptibility – Single language support – Depending on the Gödel function – Limited attacks considered
Topic-Aware [27] RNN-Stega [7] VAE-Stega [8]	Coverless and Linguistic	English	Visual: ✓ Statistical: ✓	Low	Low	+ High imperceptibility – Requiring an extra annotated dataset – Limited attacks considered – Suffering from low EC and low DR – No adherence to Kerckhoffs's principle – Depending on English linguistic rules – Single language support
Homoglyphs [10]	Structural	Latin	Visual: ✓ Statistical: ×	Medium	Medium	+ Latin/Italic languages support – Low statistical-imperceptibility – Limited attacks are considered – Depending on font style of CM – No adherence to Kerckhoffs's principle
LINE-Chat [13]	Structural	Multilingual	Visual: × Statistical: ×	Low	High	+ Multilingual support + High DR – Limited attacks considered – No adherence to Kerckhoffs's principle – Increasing gaps between words – Suffering from low EC and Low visual-imperceptibility
Graph-Steg [9]	Linguistic	English	Visual: ✓ Statistical: ✓	Low	Medium	+ High imperceptibility – Depending on synonyms of words – Suffering from low EC – Limited attacks considered – No adherence to Kerckhoffs's principle – Single language support
FontCode [14]	Structural	Multilingual	Visual: × Statistical: ✓	High	Low	+ High EC and high statistical-imperceptibility + Multilingual support – Depending on font style of CM – Low DR and low visual-imperceptibility – No adherence to Kerckhoffs's principle – Limited attacks considered

Table 6
Detection robustness analysis against stated cyberattacks.

Algorithm	Detection robustness against cyberattacks Yes:(✓) or No:(×)		
	MitM	APT	SPO
CovertSYS	×	✓	✓
Topic-Aware [27] RNN-Stega [7] VAE-Stega [8]	✓	✓	×
Homoglyphs [10]	×	✓	×
LINE-Chat [13]	×	✓	✓
Graph-Steg [9]	✓	✓	×
FontCode [14]	✓	✓	×
AITSteg [3]	×	×	✓

7. Conclusion

In this paper, we introduced a novel covert communication system (CovertSYS) that enables secure end-to-end conversation over messaging or social network platforms using multi-factor authentication and information hiding technologies. The CovertSYS scheme addresses the limitations of the state-of-the-art approaches when they are applied to short text messages. This approach focuses on the structural characteristics of text for transferring confidential information under the guise of a short carrier message, such that the embedded covert data is visually imperceptible and statistically irreversible without having the users' password. To investigate the validity of the introduced technique, we implemented the proposed scheme using Java and experimented with a proof-of-concept app by sending some benchmark examples via twenty messaging or social network platforms. Our results revealed that CovertSYS provides superior efficiency than the other state-of-the-art methods, considering three performance features: embedding capacity, imperceptibility, and distortion robustness. Moreover, this scheme could prevent various cyberattacks from detecting the presence of hidden information within the stegotexts and the decoding of the SM, even if everything about the CovertSYS is revealed except for the users' password and Bio-key. Finally, we compared and discussed the implications of the proposed technique versus current existing approaches.

Table 7
Comparative analysis of computational cost required by the state-of-the-art approaches.

Method	Summary of embedding algorithm	Minimum computational cost
CovertSYS	Binary search, hash function, and dynamic SK generation	$O(nlogn)$
AITSteg [3]	Gödel function and random SK generation	$O(2^n)$
Topic-Aware [27] RNN-Stega [7] VAE-Stega [8]	Automatic sentence generation using the recurrent neural networks	$O(2^n)$
LINE-Chat [13]	Binary search and construction of code tables	$O(n^3logn)$
Homoglyphs [10]	Binary search, hash function, and duplicate symbol substitution	$O(n^2logn)$
Graph-Steg [9]	Synonym substitution, vertex coding, and binary search	$O(n^2logn)$
FontCode [14]	Block generation based on the series of integers and font change	$O(n^2)$

CRediT authorship contribution statement

Milad Taleby Ahvanooey: Principal Investigator, Conceptualization, Methodology, Software, Data curation, Writing – Original draft. **Mark Xuefang Zhu:** Project supervision, Editing and review. **Wojciech Mazurczyk:** Consulting, Editing, and review. **Qianmu Li:** Editing, and review. **Max Kilger:** Editing, and review. **Kim-Kwang Raymond Choo:** Editing, and review. **Mauro Conti:** Editing, and review.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

No data was used for the research described in the article.

Acknowledgment

This work was supported in part by the National Planning Office of Philosophy and Social Sciences research fund, PR China (Ref No. 22BTQ017), and the National Key R&D Program, Ministry of Science and Technology of P.R. China (Ref No. 2020YFB1804604).

References

[1] Elżbieta Z, Wojciech M, Krzysztof S. Trends in steganography. Comm ACM 2014;57(3):86–95.
[2] Wojciech M, Steffen W. Information hiding: challenges for forensic experts. Comm ACM 2017;61(1):86–94.
[3] Ahvanooey MT, Qianmu L, Hou J, Mazraeh HD, Zhang J. AITSteg: An innovative text steganography technique for hidden transmission of text message via social media. IEEE Access 2018;6:65981–95.
[4] Tikkinen-Piri C, Rohunen A, Markkula J. EU general data protection regulation: Changes and implications for personal data collecting companies. Comput Law Secur Rev 2018;34(1):134–53.
[5] Keller J, Wendzel S. Covert channels in one-time passwords based on hash chains. In: Proceedings of the european interdisciplinary cybersecurity conference. 2020, p. 1–2.
[6] Rahman MS, Khalil I, Yi X. A lossless DNA data hiding approach for data authenticity in mobile cloud based healthcare systems. Int J Inf Manage 2019;45:276–88.
[7] Yang Z-L, Guo X-Q, Chen Z-M, Huang Y-F, Zhang Y-J. RNN-stega: Linguistic steganography based on recurrent neural networks. IEEE Trans Inf Forensics Secur 2018;14(5):1280–95.
[8] Yang Z-L, Zhang S-Y, Hu Y-T, Hu Z-W, Huang Y-F. VAE-Stega: linguistic steganography based on variational auto-encoder. IEEE Trans Inf Forensics Secur 2020;16:880–95.
[9] Chang C-Y, Clark S. Graph-steg: Practical linguistic steganography using contextual synonym substitution and a novel vertex coding method. Comput Linguist 2014;40(2):403–48.
[10] Rizzo SG, Bertini F, Montesi D. Fine-grain watermarking for intellectual property protection. EURASIP J Inf Secur 2019;2019(1):1–20.
[11] Flavio B, Giovanni RS, Danilo M. Can information hiding in social media posts represent a threat? Computer 2019;52(10):52–60.
[12] Ahvanooey MT, Qianmu L, Jun H, Rajput AR, Yini C. Modern text hiding, text steganalysis, and applications: a comparative analysis. Entropy 2019;21(4):355.
[13] Wu D-C, Hsu Y-T. Authentication of LINE chat history files by information hiding. ACM Trans Multimed Comput Commun Appl (TOMM) 2022;18(1):1–23.
[14] Xiao C, Zhang C, Zheng C. Fontcode: Embedding information in text documents using glyph perturbation. ACM Trans Graph 2018;37(2):1–16.
[15] Shannon CE. Communication theory of secrecy systems. Bell Syst Tech J 1949;28(4):656–715.
[16] Yang Z, Hu Y, Huang Y, Zhang Y, Wang H, Zhao X, et al. Behavioral security in covert communication systems. In: International workshop on digital watermarking. 2020, p. 377–92.
[17] Ahvanooey MT, Zhu MX, Mazurczyk W, Bendechache M. Information hiding in digital textual contents: Techniques and current challenges. Computer 2022;56(6):52–60.
[18] Hakak S, Kamsin A, Tayan O, Idris MYI, Gilkar GA. Approaches for preserving content integrity of sensitive online Arabic content: A survey and research challenges. Inf Process Manage 2019;56(2):367–80.
[19] Ahvanooey MT, Li Q, Zhu X, Alazab M, Zhang J. ANiTW: A novel intelligent text watermarking technique for forensic identification of spurious information on social media. Comput Secur 2020;90:101702.
[20] Simmons GJ. The prisoners' problem and the subliminal channel. In: Advances in cryptology. Springer; 1984, p. 51–67.
[21] Riccardo S, Abudahi L, Moonsamy V, Conti M, Poovendran R. No free charge theorem: A covert channel via usb charging cable on mobile devices. In: International conference on applied cryptography and network security. Springer; 2017, p. 83–102.
[22] Keil JM. Efficient bounded jaro-Winkler similarity based search. In: BTW 2019. Gesellschaft für Informatik, Bonn; 2019.
[23] Mahmoud M, Henderson R, Epprecht E, Woodall W. Estimating the standard deviation in quality-control applications. J Qual Technol 2010;42(4):348–57.
[24] Encoding Standard. Encoding living standard (latest update in march 2021). 2021, 2021 URL https://encoding.spec.whatwg.org/#security-background.
[25] Usage of Encodings. Usage of character encodings, statistical analysis. 2021, 2021 URL https://w3techs.com/technologies/cross/character_encoding/ranking.
[26] Khosravi B, Khosravi B, Khosravi B, Nazarkardeh K. A new method for pdf steganography in justified texts. J Inf Secur Appl 2019;45:61–70.
[27] Li Y, Zhang J, Yang Z, Zhang R. Topic-aware neural linguistic steganography based on knowledge graphs. ACM/IMS Trans Data Sci 2021;2(2):1–13.
[28] Li F, Tang H, Zou Y, Huang Y, Feng Y, Peng L. Research on information security in text emotional steganography based on machine learning. Enterp Inf Syst 2020;1–18.
[29] Kahn D. The codebreakers: the comprehensive history of secret communication from ancient times to the internet. Simon and Schuster; 1996.
[30] Luo Y, Yao C, Mo Y, Xie B, Yang G, Gui H. A creative approach to understanding the hidden information within the business data using deep learning. Inf Process Manage 2021;58(5):102615.
[31] Yang Z, Huang Y, Zhang Y-J. A fast and efficient text steganalysis method. IEEE Signal Process Lett 2019;26(4):627–31.
[32] Yang Z, Wang K, Li J, Huang Y, Zhang Y-J. TS-RNN: text steganalysis based on recurrent neural networks. IEEE Signal Process Lett 2019;26(12):1743–7.
[33] Sloan T, Hernandez-Castro J. Dismantling openpuff pdf steganography. Digit Investig 2018;25:90–6.
[34] Conti M, Dragoni N, Lesyk V. A survey of man in the middle attacks. IEEE Comm Surv Tutori 2016;18(3):2027–51.
[35] Taha A, Hammad AS, Selim MM. A high capacity algorithm for information hiding in Arabic text. J King Saud Univ-Comput Inf Sci 2020;32(6):658–65.
[36] Zulkefli Z, Singh MM. Sentient-based access control model: a mitigation technique for advanced persistent threats in smartphones. J Inf Secur Appl 2020;51:102431.
[37] Friedberg I, Skopik F, Settanni G, Fiedler R. Combating advanced persistent threats: From network event correlation to incident detection. Comput Secur 2015;48:35–57.
[38] Poletti C, Michieli M. Smart cities, social media platforms and security: online content regulation as a site of controversy and conflict. City Territ Archit 2018;5(1):1–14.

[39] Fuchs C. Societal and ideological impacts of deep packet inspection internet surveillance. Inf Commun Soc 2013;16(8):1328–59.

[40] Nagaraj N. One-time pad as a nonlinear dynamical system. Commun Nonlinear Sci Numer Simul 2012;17(11):4029–36.

[41] Gjonbalaj QD, Hamiti VR. Graph presentation of binary strings. J Appl Math Comput Mech 2018;17(2).

Milad Taleby Ahvanooey (Senior Member, IEEE) currently works as a Research Fellow at the School of Computer Science and Engineering, Nanyang Technological University (NTU), Singapore. Also, he is the co-founder and CISO of Cybercoding IT.Co., Ltd., Tsinghua University Science (TUS) Park, Jiangning District, Nanjing, China. Prior to this, he worked as a faculty member at Institute of Multimedia Information Processing, IM School, Nanjing University (NJU), P.R. China, where he completed his Postdoctoral fellowship, from Dec. 2019 to June. 2022. Also, he received his Ph.D. in Computer Engineering (Cybersecurity) from Nanjing University of Science & Technology (NJUST), Nanjing, China, (2019, with honors). His research interests include the application of AI in Cybersecurity & Digital Forensics areas as well as holds two patent applications. In addition, he achieved one of CSC Elite Outstanding awards for Ph.D. students at NJUST, in October 2019. He serves as an editorial member of Cyber Security Insights Magazine, Cryptography Journal (MDPI) from Dec. 2021–January 2023, as a reviewer board of ACM Transactions on TOMM, Computers in Human Behavior (Elsevier), Journal of Computer Security, as well as a Technical Program Committee (TPC) Member of several International conferences, such as SecureComm EAI 2022, ARES 2021 & 2022, IEEE ICME 2021 & 2022, MIC-Security 2021, IEEE MIC-Computing 2021, and IEEE ICDIS 2022.

Mark Xuefang Zhu is a Full Professor and Head of the Institute of Multimedia Information Processing with the School of Information Management at Nanjing University, China. He received his Ph.D. in Mathematics from the Department of Mathematics, Peking University in 1994, and his M.S.E, and B.E. in Telecommunications Engineering from Nanjing University of Posts and Telecommunications in 1990, and Xidian University in 1982 respectively. He has published more than 200 research papers on pattern recognition, speaker identification, multimedia information processing, digital watermarking, information security and retrieval, natural language processing, data mining, information visualization, computer graphics, haptic interaction and VR in public cultural service, information service and management, and digital humanities. He is a Member of IEEE, and the ASIS&T AAIS.

Wojciech Mazurczyk is a Professor with Institute of Computer Science, Faculty of Electronics and Information Technology, Warsaw University of Technology (WUT), Poland. He received his B.Sc. (2003), M.Sc. (2004), Ph.D. (2009, with honors) and D.Sc. (habilitation, 2014) all in Telecommunications from WUT. He also is an author or co-author of 2 books, over 150 papers, 2 patent applications and over 35 invited talks. He has been involved in many international and domestic research projects as a principal investigator or as a senior researcher. He served as a guest editor of many special issues devoted to network security (among others: IEEE TDSC, IEEE S&P, IEEE Commag). He is serving as Technical Program Committee Member of (among others): RAID, IEEE GLOBECOM, IEEE ICC, IEEE LCN, IEEE CNS, ACSAC, ARES and ACM IH&MMSec. From 2016 Editor-in-Chief of an open access Journal of Cyber Security and Mobility. From 2018 Associate Editor of the IEEE Transactions on Information Forensics and Security and Mobile Communications and Networks Series Editor for the IEEE Communications Magazine. He is also an IEEE Senior Member.

Qianmu Li is a Full Professor with the School of Cyber Science and Engineering, Nanjing University of Science and Technology (NJUST), China. He received the B.Sc. and Ph.D. degrees from Nanjing University of Science and Technology, China, in 2001 and 2005, respectively. He also was a Post-Doctorate researcher at Nanjing university, P.R. China, from September 2005 to September 2007. His main research interest is in the area of BigData Mining, Pattern Recognition, Edge Computing, Security and Privacy. In these areas, he contributed to more than 242 papers in topmost international peer-reviewed journals and conferences, and eight books. He received the China Network and Information Security Outstanding Talent Award in 2016, and multiple Education Ministry Science and Technology Awards in 2012. He has been a Visiting Researcher at FUI (2012), and WU (2017). He is also a Member of IEEE as well as the author/co-author of 8 text books.

Max Kilger is an Associate Professor in Practice in the Department of Information Systems & Cyber Security at the University of Texas at San Antonio and Director of the Master in Data Analytics Program. His mission is to help build a better, more comprehensive understanding of the relationship between people and digital technology from a national security perspective. He has over 20 years of experience in the area of information security concentrating on the social and psychological factors motivating malicious online actors, hacking groups and cyberterrorists. Max has written and co-authored a number of journal articles, book chapters and books in the national security space on profiling, motivations of malicious online actors, the social structure of the hacking community, cyberviolence, factors related to civilian attacks on critical infrastructure and the emergence of cyberterrorism. He co-authored the popular book Reverse Deception: Organized Cyberthreat Counter-Exploitation and has co-authored his second book Deception in the Digital Age. He is a founding and board member of The Honeynet Project — a not-for-profit international information security organization with 50 teams of experts in 42 countries working for the public good. Max was a member of a National Academy of Engineering committee dedicated to make recommendations for combating terrorism. He is a member of a multinational instructional team for a NATO counterterrorism course and a faculty member of the Cyber Center for Security and Analytics. He is also on the Scientific Board of the graduate interdisciplinary cyber security program at LUISS Guido Carli University in Rome, Italy.

Kim-Kwang Raymond Choo received the Ph.D. in Information Security in 2006 from Queensland University of Technology, Australia. He currently holds the Cloud Technology Endowed Professorship at The University of Texas at San Antonio. He is the founding co-editor-in-chief of ACM *Distributed Ledger Technologies: Research & Practice*, and founding chair of IEEE Technology and Engineering Management Society's Technical Committee on Blockchain and Distributed Ledger Technologies. He is an ACM Distinguished Speaker and IEEE Computer Society Distinguished Visitor (2021–2023), and included in Web of Science's Highly Cited Researcher in the Computer Science - 2021, and Cross-Field - 2020. In 2015, he and his team won the Digital Forensics Research Challenge organized by Germany's University of Erlangen–Nuremberg. He is the recipient of the the IEEE Systems, Man, and Cybernetics Technical Committee on Homeland Security (TCHS) Research and Innovation Award in 2022, and 2019 IEEE Technical Committee on Scalable Computing Award for Excellence in Scalable Computing (Middle Career Researcher), the British Computer Society's 2019 Wilkes Award Runner-up, the Fulbright Scholarship in 2009, the 2008 Australia Day Achievement Medallion, and the British Computer Society's Wilkes Award in 2008. He has also received IEEE Computer Society's Bio-Inspired Computing Special Technical Committee (STC) Outstanding Paper Award for 2021, and best paper awards from the IEEE Systems Journal in 2021, IEEE Consumer Electronics Magazine for

2020, EURASIP Journal on Wireless Communications and Networking in 2019, IEEE TrustCom 2018, and ESORICS 2015.

Mauro Conti is a Full Professor at the University of Padua, Italy. He is also affiliated with TU Delft and University of Washington, Seattle. He obtained his Ph.D. from Sapienza University of Rome, Italy, in 2009. After his Ph.D., he was a Post-Doc Researcher at Vrije Universiteit Amsterdam, The Netherlands. In 2011 he joined as Assistant Professor at the University of Padua, where he became Associate Professor in 2015, and Full Professor in 2018. He has been Visiting Researcher at GMU, UCLA, UCI, TU Darmstadt, UF, and FIU. He has been awarded with a Marie Curie Fellowship (2012) by the European Commission, and with a Fellowship by the German DAAD (2013). His research is also funded by companies, including Cisco, Intel, and Huawei. His main research interest is in the area of Security and Privacy. In this area, he published more than 450 papers in topmost international peer-reviewed journals and conferences. He is the Editor-in-Chief for IEEE Transactions on Information Forensics and Security, an Area Editor-in-Chief for IEEE Communications Surveys and Tutorials, and has been an Associate Editor for several journals, including IEEE Communications Surveys and Tutorials, IEEE Transactions on Dependable and Secure Computing, IEEE Transactions on Information Forensics and Security, and IEEE TRANSACTIONS ON NETWORK AND SERVICE MANAGEMENT. He was Program Chair for TRUST 2015, ICISS 2016, WiSec 2017, ACNS 2020, CANS 2021, and General Chair for SecureComm 2012, SACMAT 2013, NSS 2021 and ACNS 2022. He is Fellow of the IEEE, Senior Member of the ACM, and Fellow of the Young Academy of Europe.