Original Research Article

Research and implementation of information hiding technology based on text carrier

Xiaoshu Wang

Chongqing College of Mobile Communication, Hechuan District, Chongqing, 401520, China

Abstract: With the continuous development of the Internet and the expanding application of instant messaging systems, research on confidentiality has become increasingly vital. This paper focuses on text carrier-based information hiding technology applied in enterprise converged communication IM systems. It proposes five core algorithms for information hiding: encoding, synonym substitution, embedding, extraction, and decoding, designs an information hiding model, and implements the corresponding functionalities.

Keywords: Confidentiality; Converged Communication; Text carrier; Information hiding

1. Introduction

In today's rapidly evolving information age, traditional encryption faces significant limitations, prompting the emergence of information hiding technology. This technique embeds secret data into multimedia carriers, making it undetectable to attackers without decryption tools, ensuring secure network transmission. This paper proposes text-based information hiding for enterprise IM systems, enhancing privacy and communication security, offering a novel solution for information protection.

2. Research on key algorithms of information hiding

This paper employs the insertion method for information hiding, embedding mapped reserved characters at the end of the text to ensure that data volume and file size expansion do not affect text recognition and reading, while leaving the original content unaltered. The following sections will elaborate on five key algorithms: encoding, synonym substitution, embedding, extraction, and decoding.

2.1. Coding mode

Common encoding methods include ASCII, Unicode, and ANSI. ASCII, established by the American National Standards Institute, is an international standard; Unicode uses double-byte encoding and is incompatible with ANSI; ANSI, a double-byte encoding format compatible with ASCII, is primarily used for Chinese character extension and is the default text storage format in Windows. This paper adopts ANSI encoding, converting secret information into invisible reserved codes within ANSI, which are then embedded into carriers for information hiding.

Define a set of invisible codes,S select, $P \in S$, $|G| = L$, $G = \{g_1, g_2, g_3, ..., g_n\}$, any 2-byte code $g_i \in G$ is used to construct L-ary encoding. Let $2^r = L$, g_i represent a r bit digital encoding. Within $\{d_1, d_2, d_3, ..., d_n\}$, $d_j \in D$, D to G establish a bijective function $f(x) = x$ and inverse function f^{-1}.

This paper encodes the secret information of the "New Instant Messaging Hiding System" and embeds it into the text carrier using the embedding method, As shown in **Table 1**:

Table 1. Secret message coding.

ANSI Hexadecimal encoding	ANSI Binary encoding	r-bit (r=10) partitioned binary encoding		Decimal represents the binary code that has been redivided		Hexadecimal representation of serial numbers	
CED2	1100111011010010	1100111011	0011111101	[827]	[253]	[AEAD]	[AE3E]
D4DA	1101010011011010	0100101101	0101010100	[301]	[170]	[AE74]	[ADEB]
BDF1	10111110111110001	0100110110	0110110001	[310]	[433]	[AE7D]	[AE4E]
C4EA	1100010011101010	1010111101	1100111111	[701]	[831]	[AB5F]	[AEB1]
C2ED	1100001011101101	1111000111	0100101011	[967]	[299]	[ADBE]	[AE72]
C9CF	1100100111001111	0001001110	0101000000	[78]	[320]	[AD8F]	[AE87]
D2A9	1101001010101001	1010110000		[688]		[AB52]	
B1CF	1011000111001111	1011101101		[749]		[AB8F]	
D2B5	1101001010110101	1100100111		[807]		[ABC9]	

2.2. Synonym substitution

Synonym substitution achieves information hiding by replacing words in the carrier text with corresponding words from a synonym library, combined with encoding.

Carrier text *C*, secret information *M*, hidden information text *S*, synonym library *D*, embed- ding function $e(C,M,D)=S$, extraction function $d(S,D)=M$, ensure $e(C,M,D)=S$,Let *C* and *S* remain semantically unchanged.The synonym library is shown in **Table 2**:

Table 2. Thesaurus.

Original word	Fascinating	Fangxin	Longitudinal	Get to	Busy	Sprinkling
Synonym	Deliriously	Aromatic	Even if	Arrive	Rush	Overspread
Original word	Verdant	Clearly visible	Fluttering gently with the wind	Run in	Passing by	Rustle
Synonym	Fresh and green	Obviously revealed	Blowing in the Wind	Warm and moist	Drifting by	Rustling
Original word	Faintly	Seep	A touch of	Drifting with the wind	Warmth	Catch it
Synonym	Delicate	Pounce in	A strand	Swaying gently	Warm	Lift it up

The synonym substitution algorithm is as follows:

(1) Process the synonym library, remove extra spaces and store it in a one-dimensional array;

(2) Perform automatic word segmentation on the carrier text *C* and mark the synonyms to be replaced;

(3) Automatically segment the carrier text and mark the synonyms to be replaced.

(4) Traverse the words in the text *C* , use two one-dimensional arrays to represent synonyms in *C* , check if they exist in the synonym library; if so, replace them with corresponding words from the library;

(5) Repeat step (3) until all replacements are complete.

The extraction algorithm is as follows:

(1) Perform word segmentation on the text *S* ;

(2) Traverse the synonym library *D* ; if *S* exists, restore the original text based on its position and decoding method;

(3) Repeat step (2) until the original carrier text is fully restored.

2.3. Embedding algorithm

Send the secret information in the carrier text to the receiver according to the following embedding algorithm:

(1) Retrieve the client chat carrier;

(2) Wait for an operation request; if there is no embedding request, proceed to step (7); otherwise, proceed to step (3);

(3) Input the secret information, convert its ANSI encoding to a hexadecimal string, then to a binary string, and finally to decimal after re-division;

(4) Map the decimal string of the secret information to the invisible character indices in the reserved encoding set;

(5) Retrieve the synonym library, analyze the carrier text, and replace synonyms to obtain the modified carrier text;

(6) Insert the mapped ciphertext at the end of the modified carrier text, separated by a special delimiter;

(1) Send the secret information via Socket.

Figures 1 and **2** show the original carrier text and the carrier text with embedded secret information, respectively:

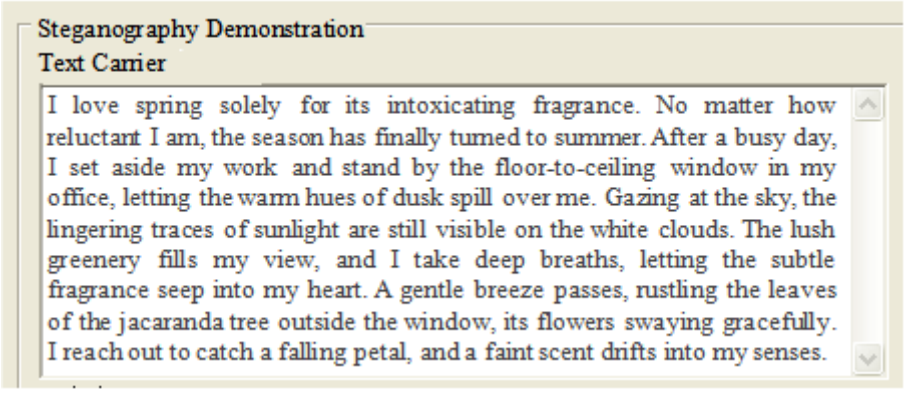

Figure 1. Original carrier text.

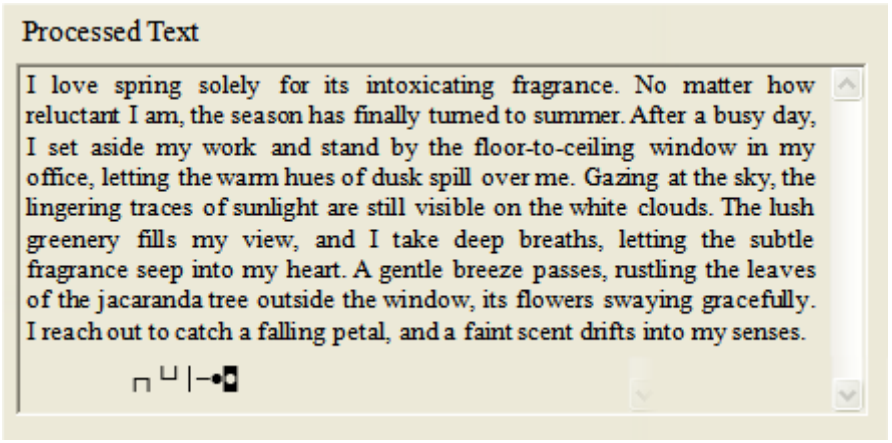

Figure 2. Carrier text after embedding secret information.

2.4. Extraction and decoding algorithms

The extraction algorithm and decoding are the inverse processes of the embedding algorithm. They analyze

the altered text carrier, map the invisible pre-encoded data back into binary code by $f^{-1}(x)=x$, and restore the secret information, which is then displayed on the client side.

3. Design of information hiding model

The characteristics of transmitting secret information in IM communication[4] include ensuring the secure arrival of secret information, avoiding continuous embedding in the text carrier, and requiring close collaboration between the sender and receiver to address issues such as the initiation, termination, synchronization, and verification of the covert channel.

Figure 3 illustrates the model diagram of secret information embedding, communication, and extraction:

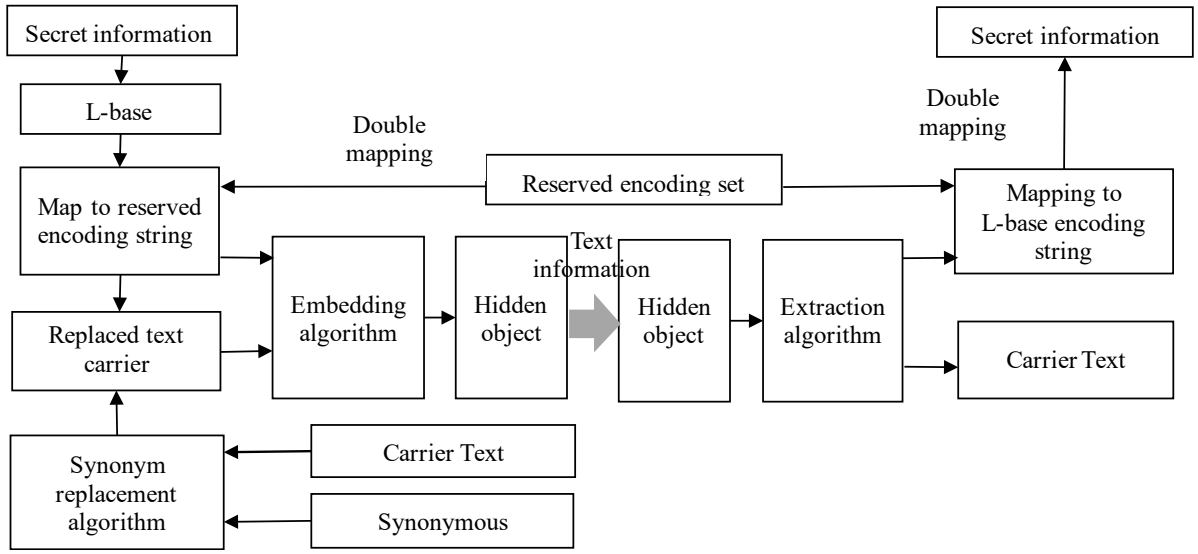

Figure 3. Model diagram of secret information embedding, communication, and extraction.

L-ary encoding: Uses bit strings for 1024-ary encoding, which can be represented as integers or in binary.

Reserved encoding characters: Invisible characters in ANSI encoding, such as space and ESC characters.

fals**Bijective function**: Establishes a one-to-one correspondence between secret information and invisible characters through mapping functions $f(x)=x$ and $f^{-1}(x)=x$.

Figure 4 illustrates the L-ary encoding reserved for ANSI code mapping:

L-base encoding		Double mapping function $f(x)=x$ $f^{-1}(x)=x$	Reserved Encoding Set (ANSI)	
Integer	Binary number		Number	Hexadecimal encoding
0	0xA140	⟷	0	00000 00000
1	0xA140	⟷	1	00000 00001
2	0xA140	⟷	2	00000 00010
. . . . .	. . . . .	⟷	. . . . .	. . . . .
1024	0xADE5	⟷	1024	11111 11111

Figure 4. Mapping between L encoding and reserved encoding in ANSI code synonym.

Synonym substitution: Establish a synonym code-book with one-to-one correspondence.

Embedding algorithm: Convert secret information into bit strings, divide them into units of (r=10) bits,

transform them into L-ary encoded strings, map them to reserved invisible characters, and embed them at the end of the carrier to form a stenographic object for transmission.

Extraction algorithm: The inverse process of the embedding algorithm, separating the steganographic object into reserved invisible secret information characters and carrier text.

Decoding algorithm: The inverse process of the encoding algorithm, i.e., the restoration of the original secret information.

4. Realization of information hiding technology

This paper implements information hiding technology through four components: encoding, synonym substitution, embedding, extraction, and decoding.

4.1. Implementation of coding algorithm

This paper employs ANSI encoding combined with a reserved character encoding library to achieve mutual mapping between secret information and invisible characters, providing input for the embedding algorithm and enhancing transmission security.

Encoding implementation steps:

(1)The function **CharToHex(char* iChBuf, char *iChTemp)** converts secret information into ANSI hexadecimal;

(2)The function **HextoBin(int hex, int len)** transforms hexadecimal into binary;

(3)The binary data stream is divided into units of =10, padding with zeros if necessary, and then recombined;

(4)The function **BinToDec(const char *str)** converts the binary data stream into decimal numbers;

(5)The function maps decimal numbers to indices in the reserved character encoding table, and **Replacesmsg(String Instr)** maps them to invisible characters;

(6)Encoding is completed.

4.2. Implementation of synonym replacement algorithm

This paper employs a synonym substitution method for information hiding, preserving the original meaning of the carrier text. It features low natural language processing requirements, strong practicality, and robust anti-detection capabilities, thereby enhancing communication security.

Synonym substitution algorithm implementation steps:

(1)The function **getFileString()** imports the carrier text document, performs word segmentation, and marks words for replacement;

(2)The function **Splitstringbysym(String instr, String sym, ArrayList<String> RtnArr ay)** processes the synonym library and determines if a word is a synonym based on the carrier text. If it is a synonym, proceed to (3) for replacement; otherwise, skip replacement and proceed to (4);

(3)The function **Replacemsg(String Instr)** performs synonym replacement, ensuring the original meaning of the carrier text remains unchanged;

(4)The replacement process concludes.

4.3. Implementation of embedding algorithm

The positions for embedding secret information include one character before and after single or double

quotation marks, two characters at the beginning of a paragraph, the end of a paragraph, and several characters after the end of the article. If character positions are insufficient, the remaining invisible characters are embedded at paragraph positions; if there are excess positions, they are filled with spaces. The embedding positions are flexibly chosen based on actual conditions. This algorithm successfully achieves information hiding, ensuring communication security.

Embedding algorithm implementation steps:

(1) Import the invisible characters and the carrier text after synonym replacement into the function **Replacesmsg(String Instr, String str)**;

(2) Use the function **Replacesmsg(String Instr, String str)** with a loop to search for punctuation marks at the end of text paragraphs;

(3) If a punctuation mark at the end of a paragraph is found, the function **Replacesmsg(Str ing Instr, String str)** inserts invisible characters; otherwise, continue searching and proceed to (4);

(4) Complete the embedding operation.

4.4. Implementation of extraction and decoding algorithm

Extraction algorithm: The function **ParseMsg(String Instr)** separates the carrier text, obtaining the synonym-replaced text and the invisible character secret information, and uses the function **Replacemsg_p()** to restore the original carrier text.

Decoding algorithm: The function maps invisible characters back to encoded strings, and the function **Replacesmsg_p()** converts the binary encoded strings into the original secret information.

4.5. Implementation of sending and receiving functions

4.5.1. Implementation of sending function

Implementation steps:

(1) The function **Baseendecode(text1, text2)** reads and displays the carrier text and code-book file;

(2) The function **Replacemsg(String Instr)** processes the carrier text for synonym substitution;

(3) Within the function **Replacesmsg(String Instr)**, **String.format("% c", Integer.parse Int(Locals Array. get(str1)))** reads the secret information in ANSI code format and maps it to invisible characters in the codebook;

(4) The function **Replacesmsg(String Instr)** inserts the invisible characters at the end of the synonym-substituted carrier text;

(5) The socket sends and displays the text carrier embedded with the secret file.

4.5.2. Implementation of the receiving function

Implementation steps:

(1) The client listener receives a message request from another client;

(2) Determine the type; if it is secret information, the function **Replacesmsg_p()** extracts invisible characters and the steganographic object; if it is a general message, proceed to (5);

(3) Reverse-map the secret information into invisible characters;

(4) The function **Replacesmsg_p()** decodes and displays the secret information;

(5) End the receiving operation.

5. Concluding remarks

This paper delves into text carrier-based information hiding technology, designed and implemented with a modular approach to facilitate functional expansion and secondary development. It has been successfully applied in enterprise unified communications. The solution demonstrates strong maintainability and scalability, and can be further optimized and refined in the future to meet new enterprise requirements.

6. References

[1] Cao Weibing, Dai Guanzhong, Xia Yu, et al. Text-based Information Hiding Technology[J]. Computer Application Research, 2003, 20(10): 39-41.

[2] Chen Gouxi, Chen Junjie. Research on the Security of Multi-carrier Information Hiding[J]. Mini-Micro Computer Systems, 2011, 32(04): 644-646.

[3] X.G.Sui, H.Luo, A steganalysis method based on the distribution of space characters, Proc of 2006 International Conference on Communications Circuits and Systems, Guilin, China, 2006,54-56.

[4] R. Asleson, T. S. Nathaniel, Foundations of Ajax[M]. Apress, 2005.