Steganography in Thai Text

Natthawut Samphaiboon and Matthew N. Dailey
Computer Science and Information Management
Asian Institute of Technology
P.O. Box 4, Klong Luang, Pathumthani 12120 THAILAND
Email: nata@cs.ait.ac.th, mdailey@ait.ac.th

Abstract—**Steganography, or communication through covert channels, is desirable when the mere existence of an encrypted message might provide useful information to eavesdroppers. Text is ideal for steganography due to its ubiquity. However, text communication channels do not necessarily provide sufficient redundancy for covert communication. We propose a steganographic scheme for Thai plain text documents that exploits redundancies in the way particular vowel, diacritical, and tonal symbols are composed in TIS-620, the standard Thai character set. The scheme is blind in that the original carrier text is not required for decoding. In an experimental evaluation, we find that the message embedding scheme allows 2.2 bytes of embedded covert text per kilobyte of carrier text on average, and that the document modifications are unnoticeable by casual observers. The method is thus a practical and effective method for covert communication over Thai plain text channels.**

I. Introduction

Privacy is a major concern for users of public networks such as the Internet. Traditionally, privacy is among the central concerns of cryptography, which achieves private communication through encryption. One problem with encryption, however, is that although it may hide the *contents* of a message, the mere *transmission* of an encrypted message may reveal something about that message's contents.

Steganography is the art and science of communicating using covert channels. Steganographic schemes have been proposed to embed secret messages in several kinds of media such as images, video, and audio.

Text steganography, in which the goal is to embed secret messages in plain text files, is among most challenging types of steganography. One reason text steganography is difficult is that text contains little redundancy compared to other media. Another is that humans are sensitive to abnormal-looking text. Because the grammatical and orthographic characteristics of every language are different, text stegonographic schemes must be specifically designed to exploit the specific characteristics of the target language. There have been several successful attempts to design text steganographic schemes for English [1], [2], Japanese [3], Korean [4], Chinese [5], Persian [6], and Arabic [6].

In this paper, we introduce a new steganographic scheme for Thai plain text documents. The scheme exploits redundancies in the way certain Thai vowel, diacritical, and tonal symbols are combined to form compound characters in TIS-620. In two studies, we find that the scheme's effects on carrier text are unnoticeable to naive human observers and that the method achieves an embedding capacity of approximately 2.2 bytes per kilobyte of carrier text.

II. Background: Thai Orthography

Thai characters can be placed vertically at four levels, namely at the *top* level, *above* level, *baseline* level, or *below* level [7]. These four levels are shown in Figure 1. Table I classifies the symbols based on their vertical placement.

The standard Thai input/output method is WTT 2.0, proposed by the Thai API Consortium (TAPIC) in 1991. WTT 2.0 defines improper symbol sequences that lead to illegal symbol compositions [8]. There are three modes of input sequence checking: *passthrough* mode, *basic check* mode, and *strict* mode. In passthrough mode, all input sequences are allowed. In basic mode, some input sequence patterns are prohibited, and in strict mode, even more sequences are prohibited. A diagram describing the legal input sequences in strict checking mode is shown in Figure 2.

WTT 2.0 also specifies how illegal character sequences should be displayed. This is necessary because it is possible to have an improper character string input in passthrough mode that then needs to be rendered.

III. Steganography in Thai Text

We define a blind text steganographic scheme $\mathcal{TS} = (\mathcal{SE}, \mathcal{SD})$ over character set Σ by an encoding algorithm $\mathcal{SE} : \Sigma^* \times \{0,1\}^* \mapsto \Sigma^* \cup \{\perp\}$ and a decoding algorithm $\mathcal{SD} : \Sigma^* \mapsto \{0,1\}^*$. The encoding algorithm $\mathcal{SE}$ takes a carrier text string $c \in \Sigma^*$ and a secret message bit string $m \in \{0,1\}^*$, and returns a stego-text character string $s \in \Sigma^*$ if successful or $\perp$ if m is too long to be embedded in c. The decoding algorithm $\mathcal{SD}$ takes a stego-text character string $s \in \Sigma^*$ and returns a bit string $m \in \{0,1\}^*$. We require that for all $c \in \Sigma^*$ and $m \in \{0,1\}^*$, if $\mathcal{SE}(c,m) \neq \perp$, then $\mathcal{SD}(\mathcal{SE}(c,m)) = m$. That is, if $\mathcal{SE}$ embeds m in c successfully, then $\mathcal{SD}$ must decode m exactly.

The essense of text steganography is to vary the way a document is written without changing its meaning. The encoder should embed the covert text in such a way that it can be reliably decoded, without producing stego-text that is distracting to casual readers.

To satisfy these design constraints, the scheme must necessarily exploit redundancy, either in the language or in the character set. In the case of Thai text steganography, a careful

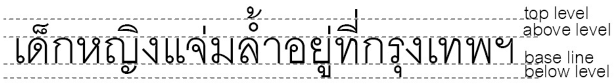


Fig. 1. Four levels of vertical placement of Thai symbols.

Class	Characters	Writing level
NON	THAI NUMBERS, and ฯ, ฿, ๆ, ๏, ๚, ๛	BASE
CONS	44 CONSONANTS	BASE
LV	เ (SARA-E) แ (SARA-AE) โ (SARA-O) ใ (SARA-AI-MAI-MUAN) ไ (SARA-AI-MAI-MA-LAI)	BASE
FV1	ะ (SARA-A) า (SARA-AA) ำ (SARA-AM)	BASE
FV2	ๅ (LAK-KHANG-YAO)	BASE
FV3	ฤ (RU) ฦ (LU)	BASE
BV1	ุ (SARA-U)	BELOW
BV2	ู (SARA-UU)	BELOW
BD	ฺ (PHIN-THU)	BELOW
TONE	่ (MAI-EK) ้ (MAI-THO) ๊ (MAI-TRI)	ABOVE
	๋ (MAI-CHAT-TA-WA)	TOP
AD1	ํ (NI-KHA-HIT) ์ (THAN-THA-KHAT)	ABOVE
AD2	็ (MAI-TAI-KHU)	ABOVE
AD3	๎ (YA-MAK-KAN)	ABOVE
AV1	ิ (SARA-I)	ABOVE
AV2	ั (MAI-HAN-A-KAT) ึ (SARA-UE)	ABOVE
AV3	ี (SARA-II) ื (SARA-UEE)	ABOVE

TABLE I
THAI ALPHABETICAL CHARACTER CLASSIFICATION. NON = NON-COMPOSABLE CHARACTERS; CONS = CONSONANTS; LV = LEADING VOWELS; FV = FOLLOWING VOWELS; BV = BELOW VOWELS; BD = BELOW DIACRITICS; TONE = TONAL SYMBOLS; AD = ABOVE DIACRITICS; AV = ABOVE VOWELS.

examination of the TIS-620 character set reveals two clear redundancies:

- The leading vowel SARA-AE can be written using either the compound character 0xE1 (แ) or separately using two SARA-E characters 0xE0 (เ).
- The trailing vowel SARA-AM, which can follow any consonant, can be written using either the compound character 0xD3 (ำ) or separately using the NI-KHA-HIT diacritical mark 0xED (ํ) followed by the SARA-AA character 0xD2 (า).

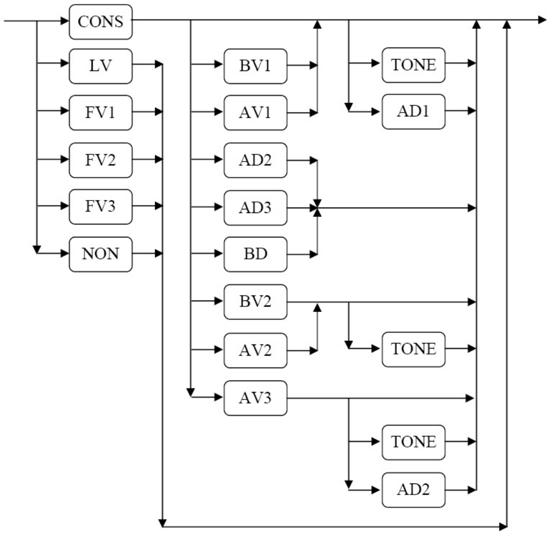
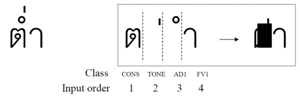


Fig. 2. Input sequence checking in strict mode.

We denote these two replacement sequences by SARA-E→SARA-E and (consonant)→NI-KHA-HIT→SARA-AA, respectively. Since the replacement sequences are LV→LV and CONS→AD1→FV1, which follow the standard input method, they are displayed properly by WTT 2.0-compliant renderers.

The difficulty with this scheme for replacement is that Thai words can contain SARA-AM composed with a tonal symbol. According to WTT 2.0, the tonal symbol must be input before SARA-AM. This means a naive replacement of SARA-AM with NI-KHA-HIT→SARA-AA creates an improper input sequence when the SARA-AM is preceded by a tonal symbol, since the final sequence would be (consonant)→(tonal symbol)→NI-KHA-HIT→SARA-AA, or CONS→TONE→AD1→FV1. An example of an improper input sequence created by naive replacement of SARA-AM is shown in Figure 3.

The solution to this problem is to swap the tonal symbol and the NI-KHA-HIT symbol, since, according to Figure 2, tonal symbols are always written after above-level symbols. Figure 4 shows an example of the SARA-AM replacement from Figure 3 with the input order fixed. Although this "fixed" input sequence is strictly improper, it can nevertheless be entered in passthrough mode or constructed by a computer program, and it is properly displayed by WTT 2.0-compliant rendering engines.



(a) (b)

Fig. 3. Invalid SARA-AM replacement. (a) Original text. (b) Improper input sequence created by replacement of SARA-AM with NI-KHA-HIT→SARA-AA.

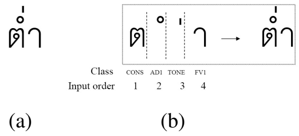


Fig. 4. Valid SARA-AM replacement. (a) Original text. (b) Properly rendered input sequence with SARA-AM replaced by NI-KHA-HIT and SARA-AA.

A. Encoding

The redundancies just described are exploited by our Thai text steganography scheme $\mathcal{TS}$. Here we describe the encoding algorithm $\mathcal{SE}$. We denote the i^{th} bit of bit string m by m_i, the j^{th} character of character string c by c_j, and the k^{th} character of character string s by s_k. A concatenation of character string s and a character c_j is denoted by $s \| c_j$. $\mathcal{SE}$ adds secret message information bits to the Thai carrier text stream by passing SARA-AE and SARA-AM characters through to the output unmodified to represent secret message bits with a value of 0 and by replacing them as previously described to represent secret message bits with a value of 1.

It is possible that the carrier text string c passed to $\mathcal{SE}$ contains instances of our steganographic modifications, e.g., SARA-E→SARA-E. To handle these cases, we introduce a preprocessing algorithm $\mathcal{P} : \Sigma^* \mapsto \Sigma^*$ that replaces any such combinations before the secret message is embedded.

Blind decoding, in which the original carrier text is not used, requires that we embed information about the length of the secret message m in the stego-text. To accomplish this, rather than embedding m directly, we construct and embed a bit string m', which is the concatenation of the n-bit binary encoding of the length of m (we use n = 32 unsigned bits) with m itself, i.e., $m' \leftarrow (|m|)_2 \| m$, where $|(|m|)_2| = n$.

Finally, to determine the embedding capacity of text string c, we introduce algorithm $\mathcal{C} : \Sigma^* \mapsto \mathbb{N}$, which counts the number of embedding symbols in c.

$\mathcal{SE}$ is described in detail in Algorithm 1.

B. Decoding

Our Thai text steganography decoding algorithm $\mathcal{SD}$ simply checks whether s_k is SARA-AE or SARA-AM, or one of the replacement patterns embedded as previously described, and outputs the corresponding secret message bit. Note from $\mathcal{SE}$ that the first n bits embedded in s encode the length of the original secret message m.

$\mathcal{SD}$ is described in detail in Algorithm 2.

IV. Experimental Evaluation

For purposes of evaluation, we implemented a prototype of the Thai text steganography scheme $\mathcal{TS} = (\mathcal{SE}, \mathcal{SD})$. A sample carrier text and the corresponding stego-text are shown in Figure 5 and Figure 6, respectively.

We performed two experiments: one to determine the visibility of the text modifications introduced by the encoding algorithm, and one to determine the embedding capacity of typical Thai text documents.

```
Input: carrier text character string $c$, secret message bit
       string $m$
Output: stego-text character string $s$
$c' \leftarrow \mathcal{P}(c)$; $m' \leftarrow (|m|)_2 \| m$; $max \leftarrow \mathcal{C}(c')$; $j \leftarrow 1$;
$s \leftarrow \varepsilon$;    // $\varepsilon$ represents the empty string
if $|m'| > max$ then
    return $\perp$;
else
    for $i = 1$ to $|m'|$ do
        while $c'_j \neq EOF$ do
            if $c'_j$ is SARA-AE or SARA-AM then
                if $m'_i$ is 0 then
                    $s \leftarrow (s \| c'_j)$; $j \leftarrow j + 1$; break;
                else if $c'_j$ is SARA-AE then
                    $s \leftarrow$ ($s$||SARA-E||SARA-E);
                    $j \leftarrow j + 1$; break;
                else if $c'_j$ is SARA-AM and $c'_{j-1}$ is TONE
                then
                    $s_{|s|} \leftarrow$ NI-KHA-HIT;
                    $s \leftarrow$ ($s \| c'_{j-1}$||SARA-AA);
                    $j \leftarrow j + 1$; break;
                else
                    $s \leftarrow$ ($s$||NI-KHA-HIT||SARA-AA);
                    $j \leftarrow j + 1$; break;
                end
            else
                $s \leftarrow (s \| c'_j)$; $j \leftarrow j + 1$;
            end
        end
    end
    while $c'_j \neq EOF$ do
        $s \leftarrow (s \| c'_j)$; $j \leftarrow j + 1$;
    end
end
return $s$;
```

Algorithm 1: Encoding algorithm $\mathcal{SE}$.

A. Experiment 1: Visibility of stego-text modifications

To determine the extent to which the text modifications introduced by the Thai text steganography encoding algorithm are visible to casual observers, we performed an informal study in which observers were asked if they could find any differences between two text documents.

As carrier text, we prepared approximately six A4 pages of Thai plain text in the AngsanaNew font at 16 points. We used our implementation of algorithm $\mathcal{SE}$ to embed a secret message of the maximum possible length into this carrier text.

We presented participants with either printed or electronic versions of the plain text and stego-text, and gave them approximately 10 minutes to find any differences between the two versions of the document, without providing any background information. After that we told participants that one of the versions of the document contained a secret message, and again asked them to find any differences between the two versions of the document. Ten Thai natives and five non-

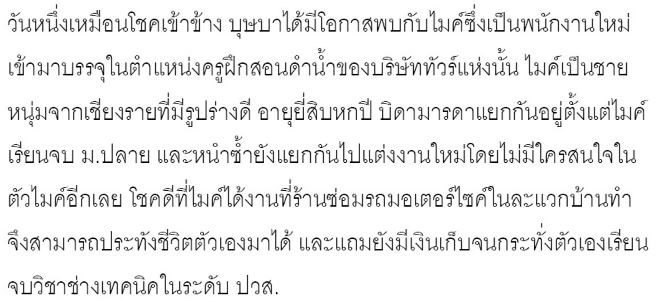
```
**Input**: stego-text character string $s$
**Output**: secret message bit string $m$
$m \leftarrow \varepsilon$; $size \leftarrow \varepsilon$;    // $\varepsilon$ represents the empty string
GLOBAL $k \leftarrow 1$;
**for** $i = 1$ *to* $n$ **do**
    $bit \leftarrow Checkbit(s_k)$; $size \leftarrow (size||bit)$;
**end**
$l \leftarrow (size)_{10}$;
**for** $j = 1$ *to* $l$ **do**
    $bit \leftarrow Checkbit(s_k)$; $m \leftarrow (m||bit)$;
**end**
return $m$;
**Subroutine** $Checkbit(s_k)$
**while** $s_k \neq EOF$ **do**
    **if** $s_k$ *is SARA-AE or SARA-AM* **then**
        $bit \leftarrow 0$; $k \leftarrow k + 1$; break;
    **else if** $s_k$ *is a replacement pattern* **then**
        $bit \leftarrow 1$; $k \leftarrow$ ($k$ + No. of symbols in the pattern);
        break;
    **else**
        $k \leftarrow k + 1$;
    **end**
**end**
return $bit$;
```

Algorithm 2: Decoding algorithm $\mathcal{SD}$.

Thai natives participated. None of the participants were able to identify any differences between the two versions of the document. In a separate test, one non-Thai native familiar with steganography and the goals of the study was informed that the stego-text carried approximately 2 message bits per line and that the embedding algorithm was not based on document formatting. This participant was able to identify the SARA-AE modification but could not identify the SARA-AM modification.

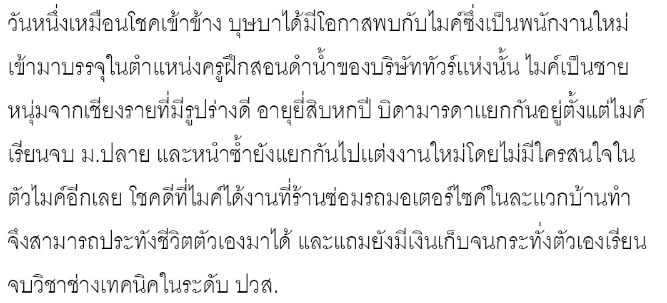
B. Experiment 2: Embedding capacity of Thai text documents

To determine the embedding capacity achievable with our algorithm, we collected ten TIS-620-encoded Thai-language documents in each of three categories, namely "news," "technical articles," and "fiction," from many sources on the Internet. We also collected 10 government documents from the Royal Thai Air Force. On average, the number of embeddable secret message bits was 0.22% of the original carrier text size, and the per-line embedding capacity was 1.93 bits.



วันหนึ่งเหมือนโชคเข้าข้าง บุษบาได้มีโอกาสพบกับไมค์ซึ่งเป็นพนักงานใหม่
เข้ามาบรรจุในตำแหน่งครูฝึกสอนดำน้ำของบริษัททัวร์แห่งนั้น ไมค์เป็นชาย
หนุ่มจากเชียงรายที่มีรูปร่างดี อายุยี่สิบหกปี บิดามารดาแยกกันอยู่ตั้งแต่ไมค์
เรียนจบ ม.ปลาย และหนำซ้ำยังแยกกันไปแต่งงานใหม่โดยไม่มีใครสนใจใน
ตัวไมค์อีกเลย โชคดีที่ไมค์ได้งานที่ร้านซ่อมรถมอเตอร์ไซค์ในละแวกบ้านทำ
จึงสามารถประทังชีวิตตัวเองมาได้ และแถมยังมีเงินเก็บจนกระทั่งตัวเองเรียน
จบวิชาช่างเทคนิคในระดับ ปวส.

Fig. 5. Sample original carrier text.



วันหนึ่งเหมือนโชคเข้าข้าง บุษบาได้มีโอกาสพบกับไมค์ซึ่งเป็นพนักงานใหม่
เข้ามาบรรจุในตำแหน่งครูฝึกสอนดำน้ำของบริษัททัวร์แห่งนั้น ไมค์เป็นชาย
หนุ่มจากเชียงรายที่มีรูปร่างดี อายุยี่สิบหกปี บิดามารดาแยกกันอยู่ตั้งแต่ไมค์
เรียนจบ ม.ปลาย และหนำซ้ำยังแยกกันไปแต่งงานใหม่โดยไม่มีใครสนใจใน
ตัวไมค์อีกเลย โชคดีที่ไมค์ได้งานที่ร้านซ่อมรถมอเตอร์ไซค์ในละแวกบ้านทำ
จึงสามารถประทังชีวิตตัวเองมาได้ และแถมยังมีเงินเก็บจนกระทั่งตัวเองเรียน
จบวิชาช่างเทคนิคในระดับ ปวส.

Fig. 6. Stego-text corresponding to the carrier text of Figure 5 with 16 message bits embedded.

V. Conclusion

We propose a new blind steganographic scheme for Thai text that exploits redundancies in the way TIS-620 represents compound characters combining vowel and diacritical symbols. We find that the modifications made when information bits are embedded in the carrier text are unnoticeable to casual observers, and that the embedding capacity of 0.22% makes the scheme practical and effective for covert communication. Simultaneously ensuring covertness, privacy, and authenticity will be the focus of our future work.

Acknowledgments

We are grateful to Chanathip Namprempre, Kiyoshi Honda, and Poompat Saengudomlert for helpful comments on this research.

References

[1] J. T. Brassil, S. Low, and N. F. Maxemchuk, "Copyright protection for the electronic distribution of text documents," *Proceedings of the IEEE*, vol. 87, no. 7, pp. 1181–1196, July 1999.

[2] D. Huang and H. Yan, "Interword distance changes represented by sine waves for watermarking text images," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 11, no. 12, pp. 1237–1245, December 2001.

[3] T. Amano and D. Misaki, "A feature calibration method for watermarking of document images," in *Proceedings of the Fifth International Conference on Document Analysis and Recognition ICDAR'99*, September 1999, pp. 91–94.

[4] Y.-W. Kim and I.-S. Oh, "Watermarking text document images using edge direction histograms," *Pattern Recognition Letters*, vol. 25, no. 11, pp. 1243–1251, August 2004.

[5] W. Zhang, Z. Zeng, G. Pu, and H. Zhu, "Chinese text watermarking based on occlusive components," *The 2nd Information and communication Texhnology ICTTA'06*, vol. 1, pp. 1850–1854, April 2006.

[6] M. H. Shirali-Shahreza and M. Shirali-Shahreza, "A new approach to Persian/Arabic text steganography," in *Proceedings of the 5th IEEE/ACIS International Conference on Computer and Information Science and 1st IEEE/ACIS International Workshop on Component-Based Software Engineering, Software Architecture and Reuse ICIS-COMSAR'06*, July 2006, pp. 310–315.

[7] T. Karoonboonyanan, "Standardization and implementation of Thai language," in *National Electronics and Computer Technology Center*, Bangkok, 1999.

[8] T. Koanantakool, "The keyboard layouts and input method of the Thai language," in *Information Processing Institute for Education and Development Thammasat University*, Bangkok, 1991.