International Journal of Innovative Computing, Information and Control
Volume **8**, Number **1(A)**, January **2012**

ICIC International ©2012 ISSN 1349-4198
pp. **139–149**

A NEW DATA HIDING SCHEME BASED ON DNA SEQUENCE

Cheng Guo[1], Chin-Chen Chang[2] and Zhi-Hui Wang[3]

[1]Department of Computer Science
National Tsing-Hua University
No. 101, Section 2, Kuang-Fu Rd., Hsinchu 30013, Taiwan
guo8016@gmail.com

[2]Department of Information Engineering and Computer Science
Feng Chia University
No. 100, Wenhwa Rd., Seatwen, Taichung 40724, Taiwan
alan3c@gmail.com

[3]School of Software
Dalian University of Technology
No. 2, Linggong Rd., Ganjingzi District, Dalian 116023, P. R. China
wangzhihui1017@gmail.com

Received August 2010; revised February 2011

Abstract. *A new DNA sequence-based data-hiding scheme is presented in this paper. We establish an injective mapping between one complementary rule and two secret bits in a message. Based on this mapping mechanism, the proposed scheme can effectively hide two secret bits in a message by replacing one character. This approach can greatly improve the embedding capacity in data hiding. Robustness and security analyses show that the probability of an attacker's making a successful recovery of the hidden data is negligible. According to the experimental results, the proposed scheme has a stable and efficient embedding capacity with a low modification rate, and the fake DNA sequence does not need to expand the length of the reference DNA sequence.*
Keywords: Data hiding, DNA, Complementary rule, Embedding capacity, Security analysis

1. **Introduction.** Data-hiding techniques are becoming increasingly important in a variety of digital media applications, including annotation, ownership protection and authentication. Most previous work has focused on how to protect information from intruders using cryptology. However, cryptology is not sufficient when transmitting data in an unsecure, public channel. Although symmetric encryption and public key encryption can ensure the confidentiality of information, they also have some inherent shortcomings, including that the encrypted data are always in a chaotic form that can attract the adversary's notice. Data hiding is different from encryption in that encryption concerns protecting the content of messages, while data hiding concerns concealing the embedded data's very existence using a steganographic approach. An increasing number of applications drive the development of data-hiding techniques.

Data hiding is a series of processes used to embed data into various forms of media, such as text [11,12], image [2,4,5,7,10,13] and video [3,14,15], with minimal changes to the "host" information carrier. The stego-media embedded data can be delivered via the usual networks, while the embedded data are invisible and inaudible except to the designated receiver. Text, video and image, furnished with distinguishing but imperceptible marks, are often used to hide data, but people are trying to find new information-carriers since more kinds of information carriers will expand the applications of data hiding techniques.

Today, biotechnology [6,17] is applicable to many aspects of life, and how to use biological information as a carrier to hide data has become an interesting challenge. DNA sequences have some inherent properties that can be utilized to hide data because it is difficult to distinguish between a real DNA sequence and a fake one. Meanwhile, the 163 million DNA sequences that are publicly available ensure the robustness and security of the corresponding data-hiding schemes. The number and ease of DNA sequences make them good candidates for hiding data. A DNA sequence is conceptually equivalent to a digital signal, so many techniques from this area are directly applicable to DNA data-embedding.

The rest of this paper is organized as follows. The next section briefly introduces related work. Section 3 presents the proposed approach. Section 4 discusses robustness and security and presents a series of experiments and comparisons to verify the proposed scheme's performance. Finally, our conclusion is offered in Section 5.

2. **Related Work.** In this section, we briefly review some previous DNA sequence-based data-hiding schemes. In recent years, a number of methods [1,8,9] have been proposed for hiding data within DNA sequences. It is well known that DNA contains sequences of four bases: adenine (A), thymine (T), guanine (G) and cytosine (C). DNA is a long polymer made from repeating units called nucleotides. DNA polymers can be very large molecules containing millions of nucleotides. DNA does not usually exist as a single molecule, but as a pair of molecules that are held tightly together. These two long strands entwine like vines in the shape of a double helix.

Table 1. Amino acid to codon mapping

Ala/A	GCU, GCC, GCA, GCG	**Leu/L**	UUA, UUG, CUU, CUC, CUA, CUG
Arg/R	CGU, CGC, CGA, CGG, AGA, AGG	**Lys/K**	AAA, AAG
Asn/N	AAU, AAC	**Met/M**	AUG
Asp/D	GAU, GAC	**Phe/F**	UUU, UUC
Cys/C	UGU, UGC	**Pro/P**	CCU, CCC, CCA, CCG
Gln/Q	CAA, CAG	**Ser/S**	UCU, UCC, UCA, UCG, AGU, AGC
Glu/E	GAA, GAG	**Thr/T**	ACU, ACC, ACA, ACG
Gly/G	GGU, GGC, GGA, GGG	**Trp/W**	UGG
His/H	CAU, CAC	**Tyr/Y**	UAU, UAC
Ile/I	AUU, AUC, AUA	**Val/V**	GUU, GUC, GUA, GUG
START	AUG	**STOP**	UAA, UGA, UAG

In [8], Shimanovsky et al. used these nucleotides to encode binary information. There are $4^3 = 64$ different codon combinations possible with a triplet codon of three nucleotides. All 64 codons are assigned for either amino acids or stop signals during translation. Therefore, there is redundancy in codons to amino acid mapping (Table 1 shows the mapping of twenty standard amino acids to codons.). The redundancy can be used to hide data. For example, the codons GCU, GCC, GCA and GCG are mapped to the unique amino acid Ala, so there are several ways to generate the same amino acid sequence. The codons GCU, GCC, GCA and GCG can represent 1, 2, 3 and 4, respectively. If the embedded message is 4, the fourth codon, GCG, is used to replace the original codon.

In 2007, Chang et al. [1] proposed two DNA sequence-based data hiding schemes. The first was a lossless compression-based data hiding scheme in which the secret message

was appended to the end of the compressed decimal-formatted DNA sequence to form a bit stream. Then, the scheme added a 16-bit header before the bit stream to record the size of the compression result. The second proposed scheme adopted the difference expansion technique to conceal a secret bit in two neighboring words, but in order to solve the inherent underflow and overflow problems, the stego results had to conform to certain constraints. In Chang et al.'s two data hiding schemes, the host DNA sequence could be reconstructed using a reversing operation.

Shiu et al. [9] also proposed three data-hiding methods based on the DNA sequence: the insertion method, the complementary pair method and the substitution method. In these methods, a reference DNA sequence S is selected and the secret message M is incorporated into it to obtain S'. In the insertion method, bits from secret message M are inserted one at a time into the reference DNA sequence, but this scheme inevitably increases the redundancy and expands the length of the DNA sequence, and the notable expansion can easily attract the attention of intruders. In the substitution method, the fake reference DNA sequence S' maintains the original sequence length by replacing the designated characters. However, the reference DNA sequence S has to be sent to the receiver in order for the receiver to identify and extract the message hidden in S'. In addition, replacement for each character is related to one secret message bit embedded in the reference DNA sequence, so if the reference DNA sequence needs to be embedded with a long secret message, the sequence will suffer from a high modification rate. As to the complementary pair method, it expands the reference DNA sequence significantly in the process of embedding the secret message. So, it can easily attract the attention of intruders. Therefore, our method just compares with the insertion method and the substitution method.

It is well known that the characters A, C, G and T can be transformed into binary codes. That is, each character can be denoted as two secret bits in a message. For example, the following rule may be used as a binary coding: $((A{:}00)(C{:}01)(G{:}10)(T{:}11))$. Obviously, replacing one character to hide only one secret bit in a message is inefficient, but how to hide two bits in a message by replacing one character has remained a challenge.

In our scheme, we enlarge the utilization of the complementary rule proposed in the Shiu et al. scheme [9] in order to increase the embedding capacity and reduce the modification rate. The experimental results show that the proposed scheme is stable, efficient, robust and secure. In addition to these advantages, our proposed protocol has the following key characteristics:

1) To the best of our knowledge, replacing one character can hide only one secret bit in the existing data hiding schemes based on a DNA sequence. In our scheme, we can hide two secret bits by replacing one character.

2) The proposed scheme does not expand the length of the reference DNA sequence in the process of embedding the secret message.

3) In the proposed scheme, the secret message can be embedded in a DNA sequence with a low modification rate.

3. **The Proposed Scheme.**

In order to ensure robustness and security, existing data hiding schemes try to maintain the length of the faked DNA sequence embedded the secret message while enlarging the embedding capacity for data hiding. In the proposed scheme, we want to hide two secret bits per character by replacement. We can reasonably assume that the secret message M has an even number of bits, and we utilize a complementary rule, discussed in [9], to establish an injective mapping. That is, each character x is assigned a complement, denoted as $C(x)$. For example, we can apply the following

complementary rule: $(AT)(CA)(GC)(TG)$, where $C(A) = T$, $C(C) = A$, $C(G) = C$, $C(T) = G$.

For each character x of a DNA sequence, x, $C(x)$, $C(C(x))$ and $C(C(C(x)))$ are not equal, so we can establish a kind of injective mapping between two secret bits and one complementary rule as follows: For a character x,

$00 \to x$;
$01 \to C(x)$;
$10 \to C(C(x))$;
$11 \to C(C(C(x)))$.

We embed a secret message M into the reference DNA sequence according to this rule, hiding two secret bits per character. For example, with the secret message $M = 00111001$, and the reference sequence $S = ATTC$ using the above rule,

$00 \to x \Rightarrow A \to A$;
$11 \to C(C(C(T))) \Rightarrow T \to A$;
$10 \to C(C(T)) \Rightarrow T \to C$;
$01 \to C(C) \Rightarrow C \to A$.

Then, the faked DNA sequence $S' = AACA$.

The proposed scheme is comprised of two processes: the data hiding process and the extraction and recovery process.

Since a DNA sequence is composed of four nucleotides, A, C, G and T, there are many repetitious pairs of nucleotides, such as AA, CC. Our scheme utilizes the repetitious pairs and the complementary rule to hide data. To simplify the discussion, assume that the secret message $M = 0110$. In the sequence: "$AT\boldsymbol{CC}GCA\boldsymbol{TT}G$", there are two pairs of repeated characters, $\boldsymbol{CC}$ and $\boldsymbol{TT}$. We use the second C and the second T to hide data. According to the rule, rule $01 \to C(C) \Rightarrow C \to A$ and $10 \to C(C(T)) \Rightarrow T \to C$, the second C and T are replaced by A and C, respectively.

This phase details the proposed scheme. Suppose the secret message $M = 00101101$, and let the reference DNA sequence $S = GAATTCATGATCAGTTGTAA$. As a prerequisite, the reference DNA sequence, the complementary rule and the injective mapping are known to the sender and receiver. The scheme works as follows:

Step 1. Label the repeated characters in the reference DNA sequence. The second characters repeated are indicated by bolding and denoted as s_i. In the case of three consecutive repeated characters or four consecutive repeated characters, such as AAA or $TTTT$, they will be identified as $A\boldsymbol{A}A$ and $T\boldsymbol{TT}\boldsymbol{T}$. The sequence S may now become $S = GA\boldsymbol{AT}\boldsymbol{T}CATGATCAGT\boldsymbol{T}GTA\boldsymbol{A}$. We use the bold characters to hide data.

Step 2. Divide M into segments, wherein each segment contains 2 secret bits. Then we have the following segments: $M = m_1m_2m_3m_4 = 00, 10, 11, 01$.

The process of embedding is illustrated in Figure 1.

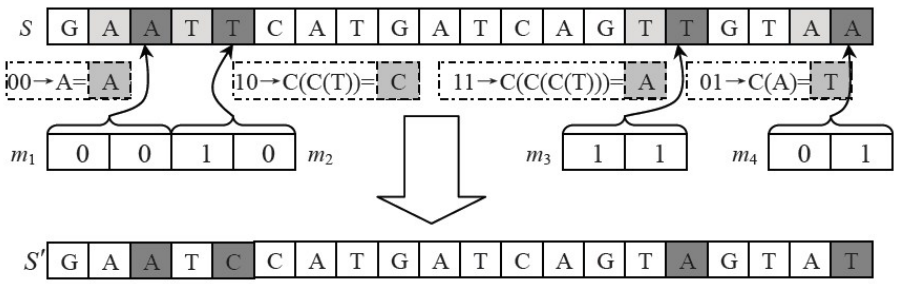

FIGURE 1. The embedding process of secret message segment

Step 3. Establish the injective mapping between the segments of M and the bold characters of S. Transform S into S' by the following rules:

If $m_i = 00$, the ith bold character is unchanged;
else if $m_i = 01$, the ith bold character is replaced by $C(s_i)$;
else if $m_i = 10$, the ith bold character is replaced by $C(C(s_i))$;
else if $m_i = 11$, the ith bold character is replaced by $C(C(C(s_i)))$.

Thus, $S' = GAATCCATGATCAGTAGTAT$.

Step 4. Send S' to the receiver.

In Algorithm 1, the formal hiding data procedure is presented.

Algorithm 1. Data hiding algorithm

Input: A reference sequence S, a complementary rule and a secret binary message M

Output: A faked DNA sequence S' with the secret message M hidden

Step 1. Read the reference DNA sequence and denote the second characters repeated as s_i

Step 2. Partition the secret message M into 2-bits segments $m_1, m_2, \cdots, m_{|M|/2}$.

Step 3. For i from 1 to $|M|/2$, do the following operations:
If $m_i = 00$, do not change s_i;
else if $m_i = 01$, change s_i to be $C(s_i)$;
else if $m_i = 10$, change s_i to be $C(C(s_i))$;
else if $m_i = 11$, change s_i to be $C(C(C(s_i)))$;

Step 4. Send the above sequence S' to the receiver.

The receiver processes the fake sequence S' and extracts the secret message embedded through the reference sequence S, the complementary rule, and the injective mapping. The extraction and recovery process is as follows:

Step 1. The receiver marks the repeated characters in the reference sequence S. The second characters repeated are indicated by bolding and denoted as s_i.

Step 2. Let the corresponding position characters of S' be denoted s'_i.

Step 3. For i from 1 to $\dfrac{|M|}{2}$, where $|M|$ is the length of the secret message,
If $s'_i = s_i$, $m_i = 00$;
else if $s'_i = C(s_i)$, $m_i = 01$;
else if $s'_i = C(C(s_i))$, $m_i = 10$;
else if $s'_i = C(C(C(s_i)))$, $m_i = 11$.

Step 4. Concatenate all m_i to be M. That is the secret message.

The receiver can recover the secret message embedded in the fake DNA sequence using the following Algorithm 2.

The example above is the basic version of the proposed method. In another version, in order to hide more information, we can make better use of consecutive repeated characters of the reference DNA sequence. Assume that there is a segment of a DNA sequence $AGCCAAAGTGTTTT$. Here, more repeated characters can be used to hide data. In the following, we compared the two schemes:

$AGC\boldsymbol{C}A\boldsymbol{A}AGTGT\boldsymbol{T}T\boldsymbol{T}$,
$AGC\boldsymbol{C}A\boldsymbol{AA}GTGT\boldsymbol{TTT}$.

The characters used to hide data are indicated by bolding. Obviously, the updated version has greater capacity for data-hiding.

4. **Security Analysis and Experimental Results.** In this section, we conduct simulations to demonstrate the security and practicability of the proposed scheme.

Algorithm 2. Data recovery algorithm

Input: A fake sequence S', a complementary rule and the reference sequence S

Output: The hidden secret message M

Step 1. Read the reference DNA sequence and denote the second characters repeated as s_i.

Step 2. Denote the corresponding position characters of S' as s'_i.

Step 3. For i from 1 to $|M|/2$, do the following operations:

If $s'_i = s_i$, $m_i = 00$;
else if $s'_i = C(s_i)$, $m_i = 01$;
else if $s'_i = C(C(s_i))$, $m_i = 10$;
else if $s'_i = C(C(C(s_i)))$, $m_i = 11$.

Step 4. Concatenate all m_i $(1 \leq i \leq |M|/2)$ to be M.

4.1. **Security analysis.** In this section, we discuss a few robustness and security issues of the proposed scheme. There are two givens: only the sender and the receiver are aware of the reference sequence, and there are roughly 163 million DNA sequences available publicly. Thus, any attacker who wants to recover the secret message embedded in the DNA sequence must guess which DNA sequence is used to hide data, with the probability of success at $1/1.63 \times 10^8$. Our scheme also applies the complementary rule to hide data defined in the substitution method [9]. As discussed in the description of the substitution method [9], the number of complementary rules needs to be considered. There are six possible complementary rules:

$(AT)(TC)(CG)(GA)$, $(AT)(TG)(GC)(CA)$,
$(AC)(CT)(TG)(GA)$, $(AC)(CG)(GT)(TA)$,
$(AG)(GT)(TC)(CA)$, $(AG)(GC)(CT)(TA)$.

In the proposed scheme, we establish an injective mapping between the possible complementary rules and the possible binary coding rules of two secret bits. The number of the injective mappings is $4 \times 3 \times 2 \times 1 = 24$. Therefore, the probability of successfully guessing the hidden message is $\dfrac{1}{1.63 \times 10^8} \times \dfrac{1}{6} \times \dfrac{1}{24}$. The reference DNA sequence, the complementary rule, and the injective mapping are all necessary for an intruder who wants to discover the secret message, so the likelihood of recovering secret message without this information is negligible.

4.2. **Simulation results.** This section describes a series of experiments carried out to evaluate the performance of the proposed scheme. The proposed scheme was tested on a Pentium(R) Dual-Core CPU E5300 2.60GHz personal computer with 1.00GB RAM. As shown in Table 2, eight DNA sequences were used as the test sample. These DNA sequences are publicly available and can be obtained by accessing the NCBI (National Center for Biotechnology Information) database [16].

First, we clarified some parameters used for evaluating the performance – *capacity*, *payload* and *bpn* – the definitions of which are shown in Table 3.

The embedding capacity for data hiding is important in these schemes. A java program was written to calculate the embedding capacity for data hiding for eight DNA sequences used as the test samples. Table 4 shows the embedding capacity of the eight tested DNA sequences using the proposed scheme's basic version.

The sender first marks out the second consecutive repeated characters of the reference DNA sequence. Then, according to the complementary rule, the sender embeds the secret message into the DNA sequence. In the secret message retrieval phase, the receiver must

Table 2. The tested DNA sequences

Locus	Number of nucleotides	Definition
AC153526	200,117	Mus musculus 10 BAC RP23-383C2
AC166252	149,884	Mus musculus 6 BAC RP23-100G10
AC167221	204,841	Mus musculus 10 BAC RP23-3P24
AC168874	206,488	Bos taurus clone CH240-209N9
AC168897	200,203	Bos taurus clone CH240-190B15
AC168901	191,456	Bos taurus clone CH240-18511
AC168907	194,226	Bos taurus clone CH240-19517
AC168908	218,028	Bos taurus clone CH240-195K23

Table 3. The definition of the terminology

Terminology	Definition
capacity	The total length of the faked reference sequence after hiding the secret message
payload	The remaining length of new sequence after extracting out the reference DNA sequence
bpn	The number of bits hidden per characters

Table 4. The basic version's embedding capacity in data hiding

Locus	Number of nucleotides	The embedding capacity in data hiding (bit)
AC153526	200,117	868,44
AC166252	149,884	662,62
AC167221	204,841	868,38
AC168874	206,488	921,58
AC168897	200,203	903,54
AC168901	191,456	840,66
AC168907	194,226	862,92
AC168908	218,028	966,82

know the reference DNA sequence. *bpn* can be computed by $bpn = \frac{|M|}{C}$, where $|M|$ is the length of the secret message, and C is the *capacity*.

Table 5 displays the experimental results in terms of the parameters used to evaluate the performance (*capacity*, *payload* and *bpn*). *Capacity* and *payload* show that the length of the fake reference DNA sequence is not expanded. Furthermore, as *bpn* is within [0.42, 0.45], the proposed scheme has an acceptable embedding capacity, and the embedding capacity is stable with different reference DNA sequences.

In the updated version of the proposed scheme, the sender first marks all consecutive repeated characters and then uses the same method as the basic version to hide the secret message. This process is identical to the basic version in that the receiver must know the reference DNA sequence. Table 6 shows the embedding capacity of the eight tested DNA sequences using the updated version of the proposed scheme, and Table 7 shows the experimental results of using the proposed scheme's updated version to hide data in the eight tested DNA sequences. The length of the fake reference DNA sequence is not still expanded, and *bpn* is within [0.56, 0.62].

TABLE 5. The basic version's experimental results of the proposed scheme

Locus	**Number of nucleotides**	***capacity C***	***payload P***	$bpn = \|M\|/C$
AC153526	200,117	200,117	0	0.434
AC166252	149,884	149,884	0	0.442
AC167221	204,841	204,841	0	0.424
AC168874	206,488	206,488	0	0.446
AC168897	200,203	200,203	0	0.451
AC168901	191,456	191,456	0	0.439
AC168907	194,226	194,226	0	0.444
AC168908	218,028	218,028	0	0.443

TABLE 6. The updated version's embedding capacity in data hiding

Locus	**Number of nucleotides**	**The embedding capacity in data hiding (bit)**
AC153526	200,117	114,898
AC166252	149,884	865,62
AC167221	204,841	115,214
AC168874	206,488	123,846
AC168897	200,203	122,708
AC168901	191,456	111,642
AC168907	194,226	115,786
AC168908	218,028	129,204

TABLE 7. The updated version's experimental results of the proposed scheme

Locus	**Number of nucleotides**	***capacity C***	***payload P***	$bpn = \|M\|/C$
AC153526	200,117	200,117	0	0.574
AC166252	149,884	149,884	0	0.578
AC167221	204,841	204,841	0	0.562
AC168874	206,488	206,488	0	0.600
AC168897	200,203	200,203	0	0.613
AC168901	191,456	191,456	0	0.583
AC168907	194,226	194,226	0	0.596
AC168908	218,028	218,028	0	0.593

TABLE 8. The average capacity of hiding data and the average *bpn* of two different versions

Version	**The average embedding capacity in data hiding (bit)**	**Average bpn**
The basic version	861,87	0.440
The updated version	114,983	0.587

Table 8 compares the two versions in terms of the average embedding capacity and the average *bpn*. Compared with the basic version's experimental results, the updated version's embedding capacity for data hiding and average *bpn* are significantly increased. Since the reference DNA sequence has a large number of repeated characters, the updated version can utilize more consecutive repeated characters to hide data, while the basic version can utilize only the second consecutive repeated characters to hide data. Therefore,

the updated version has greater embedding capacity for data hiding than the basic version has, and the number of bits hidden per character in the updated version is higher than the number in the basic version.

4.3. **Discussion.** In most data hiding schemes, the cover media will experience some distortion. The data hiding technique is concerned with concealing the very existence of the hidden data so it can be unobtrusively communicated. So, minimizing the distortion of the cover media is important. As to the DNA sequence, the expansion rate and the modification rate of the referenced DNA sequence are used to measure the quality of the fake DNA sequence. The formula for the expansion rate and the modification rate are described as follows:

The expansion rate (ER): $\mathrm{ER} = \dfrac{H' - H}{H}$, where H' is the length of the fake DNA sequence, and H is the length of the reference DNA sequence.

The modification rate (MR): $\mathrm{MR} = \dfrac{s_i \oplus s'_j}{H} \times 100\%$, where s_i is the i-th binary bit of the binary sequence of the reference DNA sequence, and s'_j is the j-th binary bit of the binary sequence of the fake DNA sequence.

Table 9 compares the expansion rate of the fake DNA sequence for the proposed scheme and for Shiu et al.'s two schemes, and Table 10 compares the modification rate of the fake DNA sequence for the proposed scheme and those for Shiu et al.'s two schemes.

As shown in Tables 9 and 10, the proposed scheme allows the receiver to obtain a better quality fake DNA sequence with a lower expansion rate and lower modification rate than Shiu et al.'s two schemes do.

TABLE 9. Comparison of the expansion rate (ER) of DNA sequence among Shiu et al.'s two schemes and our scheme updated version (embedding 20000 bits)

Locus	**Number of nucleotides**	**Insertion scheme (ER)**	**Substitution scheme (ER)**	**Our scheme (ER)**
AC153526	200,117	4.76%	0	0
AC166252	149,884	6.25%	0	0
AC167221	204,841	4.65%	0	0
AC168874	206,488	4.62%	0	0
AC168897	200,203	4.76%	0	0
AC168901	191,456	4.96%	0	0
AC168907	194,226	4.90%	0	0
AC168908	218,028	4.39%	0	0

We also compare related work in terms of the additional requirements of embedding the secret data into the reference DNA sequence.

As presented in Table 11, compared with the two other methods, our scheme does not require additional information in the process of embedding and extracting the secret message, and the proposed scheme does not expand the length of the DNA sequence. However, the receiver must know the reference DNA sequence to extract the secret message from the fake DNA sequence, which requirement limits the utilization of the proposed scheme. How to extract the secret message from the fake DNA sequence without the reference DNA sequence is the subject of further work.

Based on these experimental results, we can conclude that the proposed scheme has a high, stable embedding rate without expanding the length of the reference DNA sequence, and it achieves an appropriate tradeoff between embedding capacity and robustness.

TABLE 10. Comparison of the modification rate (MR) of DNA sequence among Shiu et al.'s two schemes and our scheme updated version (embedding 20000 bits)

Locus	Number of nucleotides	Insertion scheme (MR)	Substitution scheme (MR)	Our scheme (MR)
AC153526	200,117	14.28%	95.00%	5.00%
AC166252	149,884	18.76%	93.33%	6.67%
AC167221	204,841	13.96%	95.12%	4.88%
AC168874	206,488	13.86%	95.16%	4.84%
AC168897	200,203	14.27%	95.01%	4.99%
AC168901	191,456	14.89%	94.78%	5.22%
AC168907	194,226	14.69%	94.85%	5.15%
AC168908	218,028	13.16%	95.41%	4.59%
Average	195655	14.73%	94.83%	5.17%

TABLE 11. Comparisons among Shiu et al.'s two schemes and our scheme

	Insertion scheme [9]	Substitution scheme [9]	Our scheme
Expansion(1)	Yes	No	No
Original DNA sequence(2)	No	Yes	Yes
Additional information(3)	random number seeds k and r	No	No

(1): Whether the method expands the original DNA sequence.
(2): Whether the receiver requires the original DNA sequence aiming at retrieve the secret data
(3): Whether the method requires additional information

5. **Conclusions.** Recently, Shiu et al. proposed three data-hiding methods based on DNA sequences. However, in their scheme, only one secret bit can be hidden by replacing one character. The current paper proposes a simple and improved data hiding scheme based on a DNA sequence scheme. In our scheme, we establish an injective mapping between one complementary rule and two secret bits. Based on this mapping mechanism, the proposed scheme can hide two secret bits by replacing one character. Furthermore, the security analysis shows that it is, for all practical purposes, impossible for an intruder to recover the secret message. Finally, the experiments indicate that the proposed scheme has a stable and efficient embedding capacity for data hiding with a low modification rate and without expanding the length of the reference DNA sequence.

REFERENCES

[1] C.-C. Chang, T.-C. Lu, Y.-F. Chang and C.-T. Lee, Reversible data hiding schemes for deoxyribonucleic acid (DNA) medium, *International Journal of Innovative Computing, Information and Control*, vol.3, no.5, pp.1145-1160, 2007.

[2] D. Chou, C.-Y. Jhou and S.-C. Chu, Reversible watermark for 3D vertices based on data hiding in mesh formation, *International Journal of Innovative Computing, Information and Control*, vol.5, no.7, pp.1893-1901, 2009.

[3] M. C. Q. Farias, M. Carli and S. K. Mitra, Objective video quality metric based on data hiding, *IEEE Transactions on Consumer Electronics*, vol.51, no.3, pp.983-992, 2005.

[4] Y. P. Hsieh, C. C. Chang and L. J. Liu, A two-codebook combination and three-phase block matching based image-hiding scheme with high embedding capacity, *Pattern Recognition*, vol.41, no.10, pp.3104-3113, 2008.
[5] A. Kingston and F. Autrusseau, Lossless image compression via predictive coding of discrete radon projections, *Image Communication*, vol.23, no.4, pp.313-324, 2008.
[6] J. Li, Q. Zhang, R. Li and S. Zhou, Optimization of DNA encoding based on combinatorial constraints, *ICIC Express Letters*, vol.2, no.1, pp.81-88, 2008.
[7] C. C. Lin, W. L. Tai and C. C. Chang, Multilevel reversible data hiding based on histogram modification of difference images, *Pattern Recognition*, vol.41, no.12, pp.3582-3591, 2008.
[8] B. Shimanovsky, J. Feng and M. Potkonjak, Hiding data in DNA, *Proc. of the 5th International Workshop on Information Hiding*, *LNCS*, Netherlands, vol.2578, pp.373-386, 2002.
[9] H. J. Shiu, K. L. Ng, J. F. Fang, R. C. T. Lee and C. H. Huang, Data hiding methods based upon DNA sequences, *Information Sciences*, vol.180, no.11, pp.2196-2208, 2010.
[10] W.-L. Tai and C.-C. Chang, Data hiding based on VQ compressed images using hamming codes and declustering, *International Journal of Innovative Computing, Information and Control*, vol.5, no.7, pp.2043-2052, 2009.
[11] O. Vybornova and B. Macq, Natural language watermarking and robust hashing based on presuppositional analysis, *Proc. of the IEEE International Conference on Information Reuse and Integration*, Las Vegas, pp.177-182, 2007.
[12] Z. H. Wang, C. C. Chang, C. C. Lin and M. C. Li, A reversible information hiding scheme using left-right and up-down Chinese character representation, *Journal of Systems and Software*, vol.82, no.8, pp.1362-1369, 2009.
[13] M. Wu and B. D. Liu, Data hiding in binary image for authentication and annotation, *IEEE Transactions on Multimedia*, vol.6, no.4, pp.528-538, 2004.
[14] M. Wu and B. D. Liu, Data hiding in image and video: Part I – Fundamental issues and solutions, *IEEE Transactions on Image Processing*, vol.12, no.6, pp.685-695, 2003.
[15] M. Wu, H. Yu and B. D. Liu, Data hiding in image and video: Part II – Designs and applications, *IEEE Transactions on Image Processing*, vol.12, no.6, pp.696-705, 2003.
[16] Website, *NCBI*, http://www.ncbi.nlm.nih.gov/.
[17] S. Zhou, Q. Zhang, J. Zhao and J. Li, Optimization of DNA encodings based on free energy, *ICIC Express Letters*, vol.1, no.1, pp.33-37, 2007.