2015 IEEE CS Security and Privacy Workshops

Passing go with DNA sequencing: Delivering messages in a covert transgenic channel

Ji Young Chun
Graduate School of Information Security
Korea University
Email: jychun@korea.ac.kr

Hye Lim Lee
Graduate School of Information Security
Korea University
Email: dream8933@naver.com

Ji Won Yoon
Graduate School of Information Security
and Cyber Defense Department
Korea University
Email: jiwon_yoon@korea.ac.kr

***Abstract*—DNA which carries genetic information in living organisms has become a new steganographic carrier of secret information. Various researchers have used this technique to try to develop watermarks to be used to protect proprietary products; however, as recent advances in genetic engineering have made it possible to use DNA as a carrier of information, we have realized that DNA steganography in the living organism also facilitates a new, stealthy cyber-attack that could be used nefariously to bypass entrance control systems that monitor and screen for files and electronic devices. In this paper, we explain how "DNA-courier" attacks could easily be carried out to defeat existing monitoring and screening techniques. Using our proposed method, we found that DNA as a steganographic carrier of secret information poses a realistic cyber-attack threat by enabling secret messages to be sent to an intended recipient without being noticed by third parties.**

I. Introduction

Deoxyribonucleic acid (DNA) carries genetic information in living organisms. Recent advances in genetic engineering have made it possible to insert artificial DNA strands with non-genetic information into cells of living organisms. Several methods for doing so have been developed using DNA steganography for the purpose of communicating secret hidden messages [7], [13]. These methods have accomplished this goal while maintaining the regulatory functions of the underlying DNA.

DNA steganography could present a new threat if used to circumvent maximum-security screenings, which are capable of detecting hardware or electronic devices such as cell phones, laptops, and data storage devices. To date, there has been no way of sending secret messages through maximum security screenings, as electronic devices with access to the Internet are not permitted through. In this situation, DNA steganography could be used to carry secret information and make detection difficult if not impossible. A spy could hide messages in artificial DNA [1]. For example, after encoding messages into bacteria and growing it on agar plates, the messaged bacteria could be transferred to a thin film that could be attached to the spy's belongings such as clothes. Then, the spy could carry the secret messages into maximum-security places with little risk of detection and accomplish the mission. Such an attack would be relatively easy to pull off since synthesizing artificial DNA is not that expensive, cents per each base pair, and the messages last approximately 3 months at room temperature. Another example of this type of steganography is hiding messages into living organisms. First, a government or industry agent or spy could actually hide secret messages in genetically engineered bacteria. The agent would inject the messaged bacteria into his blood in order to ultimately share the secret information with an intended recipient, and only that recipient. The agent could move about freely without fear of the secret message being detected. Once the agent injects himself with the encoded bacteria, there has to be a way to retrieve it. Blood is an appropriate growth medium in which encoded bacteria can multiply; the agent could then give some blood to an intended recipient to read the message using a sequencing machine, having traveled to the recipient without fear of anyone detecting that he is carrying a secret message. This avoids the risk of detection inherent in entrance control systems that screen for electronic devices such as cell phones, laptops, and data storage devices, and eliminates the smaller risk of detection when the carrier is a thin plate.

In this work, we address the potential of using DNA steganography as a new cyber-attack model to bypass systems that screen for electronic devices. We also show that it is not difficult to perform such an attack: we introduce a new cryptographic attack method that is able to hide a message in DNA without being noticed by third parties, and even if detected, the content of the message remains confidential. No previous research has offered models with this degree of protection since they do not protect any message from being read once it is detected. None of the methods that are currently used to hide messages in DNA use sound cryptographic techniques. Instead, most existing papers use simple substitution algorithms to hide plaintext, but anyone who finds the hidden message there can read it. With a cryptographic algorithm, the message would be in ciphertext rather than in plaintext so that it could not be read even if it were found. Our method achieves the twin goals of making it as difficult as possible for third parties to detect the hidden message and rendering the message unreadable in case it is detected. Therefore, we first encrypt a secret message using a cryptographic encryption algorithm. Then, the ciphertext is encoded using the four different nucleotides in DNA: `A` (Adenine), `C` (Cytosine), `G` (Guanine), and `T` (Thiamine). Thus, the encoded ciphertext is an artificial DNA sequence inserted into a living organism's DNA. In order to hide an inserted message, the encoded ciphertext has to be indistinguishable from the living organism's DNA sequences. Otherwise, the ciphertext will merely be unreadable, but will not be undetectable. Therefore,

This research was supported by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Science, ICT and Future Planning (NRF-2013R1A1A1012797).

DOI 10.1109/SPW.2015.10

we propose an encoding scheme that outputs a message that looks like the living organism's DNA sequences based on statistical frequency analysis of those DNA sequences. Using our proposed technique, anyone could send secret messages to a targeted recipient without being noticed by third parties.

Note that in this paper we show that the organisms can be kept alive even after the artificial DNA sequence is inserted while other approaches to DNA-based watermarking or steganographic carriers are not *in vivo* [7], [13]–[17], [23], [24]. For example, Clelland et al. [7] insert data in a particular region of the human DNA strand which is identified by given forward and backward primers. The DNA strand is actually from a human being but after extracting it, they focus not on the human cell but only on the DNA sequence. That is, it is not a alive system anymore. Therefore, we present how to insert the artificial DNA sequence in a living organism in this paper.

Therefore, the main contributions of this paper are as follows:

- we address the potential of using DNA steganography as a new cyber-attack model to bypass systems that screen for electronic devices;
- we propose a steganographic method to synthesize an artificial DNA sequence that is indistinguishable from the living organism's DNA sequence; and
- we suggest several practically useful regions, including non-coding regions and bacterial plasmids, to embed such artificial DNA sequence.

The rest of this paper is organized as follows. In Section II, we provide background on steganography and coding and non-coding regions within living organism genomes. An overview of our attack is presented in Section III. Next, we present our DNA-courier attack in Section IV. We detail our mapping table in Section V, and fake data embedding in Section VI, which are used in our DNA-courier attack. We analyze our attack in Section VII and discuss related work in Section VIII. In the end, we conclude in Section IX.

II. Background

A. Steganography

Steganography is a method for concealing the existence of a message within another message, image, or file. While the focus of cryptography is to use encryption to render a message unreadable, the focus of steganography is to hide the message so that ideally only the intended recipient would be able to detect the hidden communication. The use of steganographic techniques dates back to ancient Greece, where historical methods relied on physical steganography, for example on the human skin. In modern times, the form of the carrier of secret information has changed. After the two World Wars, a new carrier technique using electromagnetic waves was introduced. Recently, the most popular carriers include digital audio and video files. Steganographic techniques in these media have enabled the sending of secret messages to a targeted recipient without being noticed by third parties. However, those secret message carriers can be detected in maximum-security places where sending or taking files into the places is restricted.

B. Coding and non-coding regions

Two distinct regions exist within living organism genomes: coding regions and non-coding regions (See Figure 1). The coding regions of an organism's DNA sequences encode protein sequences, whereas non-coding regions of an organism's DNA sequences do not encode protein sequences. In the past, most non-coding DNA was considered to have no known particular regulatory function and was referred to as "junk DNA"; however, recent works show that up to 80% of non-coding DNA may be responsible for biological regulatory functions [9]. In the remaining 20% of non-coding DNA, it is safe to assume that DNA can be freely changed into new artificial DNA sequences. In fact, several data embedding experiments have successfully performed in these regions [13], [24]. Coding regions can also be used to insert secret messages, but only a limited number of DNA sequences can be changed in a coding region before the integrity of the essential protein structures of the host organism is compromised [3]. Since more artificial DNA sequences can be inserted into non-coding DNA, longer secret messages will need to be hidden in non-coding regions.

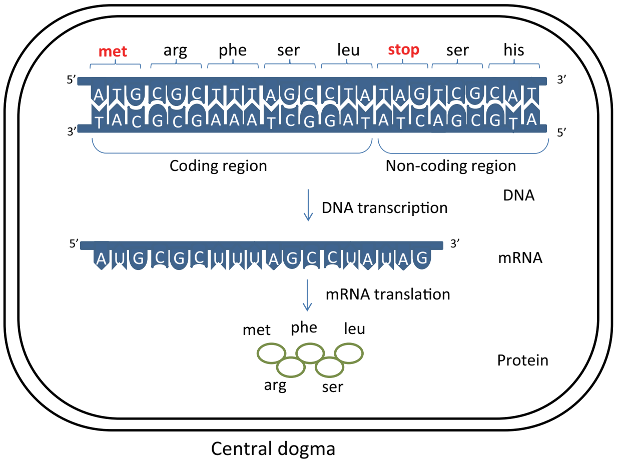


Fig. 1. Coding and Non-Coding Regions

C. Transgenesis

Transgenesis is the process of integrating exogenous genes into a living organism by genetic engineering technology so that these genes can be expressed in and inherited by its offspring [18]. The main goal of transgenesis is to add specific functions to living organisms in order to over-express particular genes or to mutate targeted genes. There are several gene transfer technologies: DNA microinjection, retrovirus-mediated gene transfer, and stem cell transgenesis. In addition, we can knockout particular genes via transgenesis by replacement. The most well-known approach is using plasmids which are physically separated from chromosomal DNA and can independently replicate. For example, we can create a transgenic plant or animal using a plasmid following these steps:

1) Isolate the DNA sequence that you want to encode. In this case, after cutting particular DNA from a genome

using restriction enzymes, a DNA ligase enzyme can be used to link and create a genetic code which are not normally found in nature.

2) Insert the isolated DNA into a plasmid.
3) Inject the artificial plasmid into bacteria.
4) Grow a large number of bacteria containing this artificial plasmid.
5) Inject a large amount of bacteria into the organism.

Figure 2 shows step 2) to step 4) of the transgenesis using a plasmid.

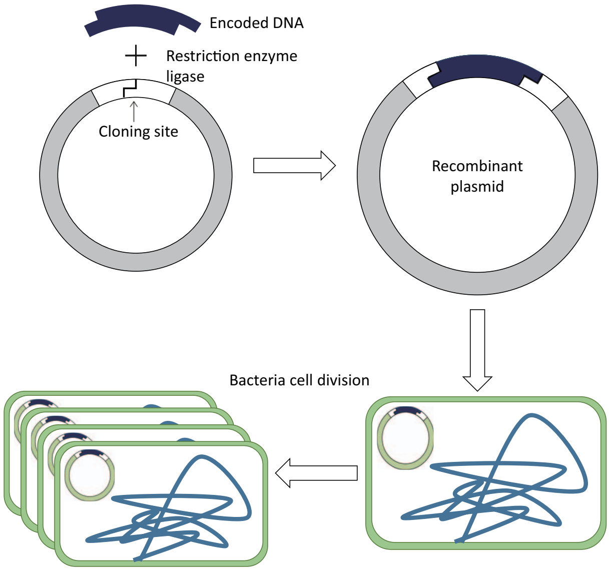


Fig. 2. The process to put encoded DNA into bacteria using a plasmid.

D. Ethical Issues

In this paper, we show the potential of a synthetic biology-based attack model which modifies natural genes into artificial genes. To date, such transgenic techniques have raised several ethical concerns [4]–[6], [19], [21]. Glenn classified three different concerns [12] by

- **social concerns**: What social and legal controls should be placed on such research?;
- **extrinsic concerns**: Are there long-term effects on the environments when such genetically modified organisms are released in the field?; and
- **intrinsic concerns**: Are there fundamental issues with creating new species?

In order to minimize the above concerns of the ethical issues, we basically design the attack model and simulate it using computational and mathematical analysis in a dry lab. In addition, we embed the artificial DNA sequence in non-coding regions and plasmid DNA sequences in order to avoid compromising the regulatory functions of the underlying DNA.

III. Overview

In this section, we will explain a new stegonagraphic attack that we designed to see how bypass entrance control systems that monitor and screen for electronic devices might be foiled. We found that this attack could realistically be carried out, which suggests the need for counterattack measures. Our attack used DNA data embedding, a technique that is currently in its infancy, but one which will undoubtedly grow as underlying techniques for DNA sequencing and synthesizing become cheaper and faster.

We designed our attack with the goal of sending secret messages that could not be detected by third parties. We did this by inserting synthesized DNA encoded with secret messages into living organism genomes. To make it hard to detect the existence of the hidden messages, any inserted synthesized DNA has to be indistinguishable from the living organism's DNA sequences and maintain the regulatory functions of the underlying DNA. To accomplish this, our proposed method ensures that the codon statistics on the synthesized DNA remain similar to the living organism's DNA, making it difficult to infer the existence of hidden messages.

In our attack model, we assume there to be three parties, Alice, Bob, and Carol as shown in Figure 3. Alice wants to send secret messages to Bob without being noticed by Carol. To reach Bob, Alice has to pass through Carol's entrance control system, which monitors and screens for electronic devices. Therefore, Alice cannot carry the secret messages in electronic devices. In this situation, Alice could carry the secret messages in DNA to successfully bypass the Carol's system.

Basically there are three ways to deliver DNA sequences with embedded secret messages. Firstly, artificially encoded DNA itself can be transferred to the destination by attaching it to Alice' cloths. Secondly, she can use not a raw DNA sequence but a living bacteria of which plasmids have encoded artificial sequence. The last way is using a host where the living bacteria can be injected and transferred. In this paper, we mainly focus on the last way which is based on the transgenesis. That is, she can be the host or can use another living organism as a host, such as a plant or a pet.

After designing how to insert transgenes into a living organism, we considered the scheme to deliver the transgenic organism. The next thing which we need to consider is how to encode the secret message to put into the DNA. That is, we need to construct an algorithm to encode and decode between DNA and secret messages.

As we know, a DNA sequence is represented as a succession of four letters, A, C, T, and G. We denote a DNA sequence as vector $\mathbf{v} = (v_1, ..., v_n)$, where $v_i \in \{\mathtt{A}, \mathtt{C}, \mathtt{T}, \mathtt{G}\}$. We assume that Alice and Bob share a) a secret key to the encryption algorithm and b) the mapping table that maps ciphertext bits into letters in a DNA sequence. Alice performs the following encoding procedure:

1) Messages are encrypted to get an ℓ-bit ciphertext, $\mathbf{c} = (c_1, ..., c_\ell)$, using an encryption algorithm.
2) The ciphertext is converted into an encoded DNA sequence using the mapping table.
3) To make the synthesized DNA sequence which is indistinguishable from the living organism's DNA se-

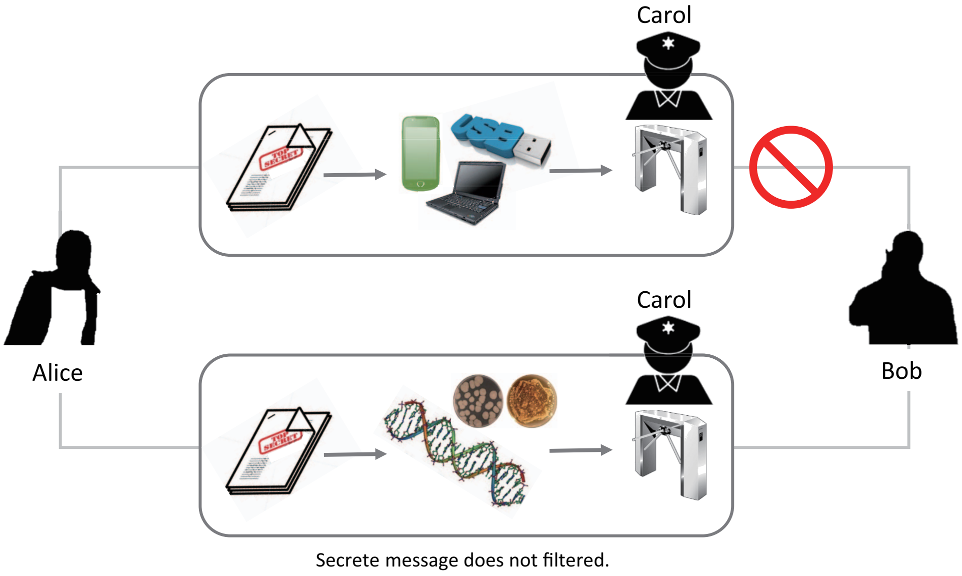


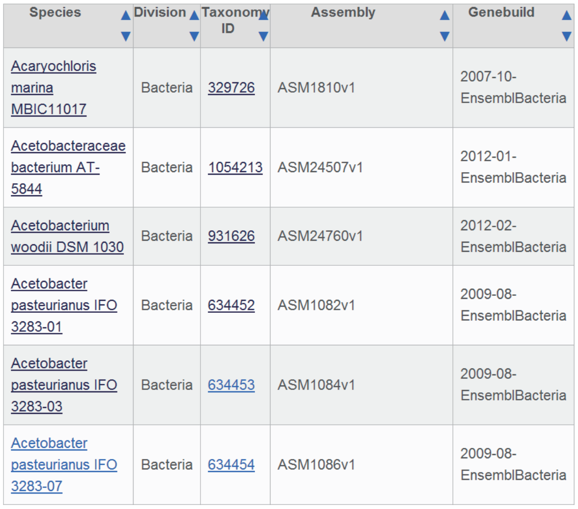
Fig. 3. Our attack scenario using biological covert channels to bypass traditional screening system: when Alice wants to transmit secret messages to Bob without being notice by Carol, there are three possible ways using our propsed approach. Firstly, artifically encoded DNA itself can be sent to the destination by attaching it to Alice's cloth. Secondly, Alice can use a living bacteria of which plasmids have encoded artificial sequence. Lastly, Alice can instead send a host where the living bacteria can be injected and transferred.

quence, fake data is embedded into the encoded DNA sequence based on a statistical frequency analysis of those DNA sequences.

After performing the encoding procedure, Alice gets the synthesized DNA sequence. Alice injects the synthesized DNA sequence into bacteria to be able to pass through Carol's entrance control system. After reaching Bob, Alice draws blood with the synthesized DNA sequence and gives it to Bob. Bob can use the mapping table to remove the embedded fake data from the synthesized DNA to get the encoded DNA sequence and convert the encoded DNA sequence back into ciphertext. Bob decrypts the ciphertext using the secret key and reads the secret messages. Instead of injecting herself with the encoded bacteria, Alice could pass through the Carol's system with the encoded bacteria as if it is an usual biological sample.

IV. DNA-Courier Attack

As mentioned in the background section, our attack model is based on synthetic biology so we need to minimize the ethical issues. Therefore, we embed encrypted data without compromising or with less dangering the regulatory functions of the underlying DNA. We could also use a plasmid, a small DNA molecule that replicates itself independently of the host cell chromosome, to safely insert messages. We can insert messages into a plasmid using restriction enzymes and a DNA ligase. Recombinant plasmid is easily inserted into bacteria, and then the bacteria with the plasmid can be cloned. Alternatively, we can use the 20% of non-coding regions that do not have any particular regulatory role to insert secret messages as mentioned in the previous subsection. In this paper, we use DNA sequences in non-coding regions of Acetobacteraceae bacteria AT-5844, identified by Ensembl Genomes in order to embed a secret message in DNA to carry out a DNA-courier attack [8]. (See Figure 4.)



Species	Division	Taxonomy ID	Assembly	Genebuild
Acaryochloris marina MBIC11017	Bacteria	329726	ASM1810v1	2007-10-EnsemblBacteria
Acetobacteraceae bacterium AT-5844	Bacteria	1054213	ASM24507v1	2012-01-EnsemblBacteria
Acetobacterium woodii DSM 1030	Bacteria	931626	ASM24760v1	2012-02-EnsemblBacteria
Acetobacter pasteurianus IFO 3283-01	Bacteria	634452	ASM1082v1	2009-08-EnsemblBacteria
Acetobacter pasteurianus IFO 3283-03	Bacteria	634453	ASM1084v1	2009-08-EnsemblBacteria
Acetobacter pasteurianus IFO 3283-07	Bacteria	634454	ASM1086v1	2009-08-EnsemblBacteria

Fig. 4. Genomes in Ensembl Genomes

A. General Procedure of DNA-courier attack in transgenesis

With the bacterial genome, plaintexts and a secret key, we can make synthesized DNA as shown in the encoding procedure of Figure 5.

In our simulation, we sought to hide the plaintext message, "`HelloMyNameIsLHL`" in non-coding regions of the

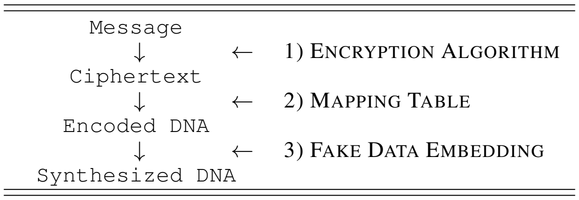


Fig. 5. Data Encoding Procedure

selected bacteria. We could have chosen a longer message, but kept it short for simplicity's sake. The message was encrypted using the encryption algorithm AES-128 [25] with a secret key, `aaaaaaaabbbbbbbbccccccccdddddddd`, to convert the message to ciphertext, as follows:

- A used encryption algorithm: `AES-128`
- A secret key: `aaaaaaaabbbbbbbbccccccccdddd dddd`
- A plaintext: `HelloMyNameIsLHL`
- A ciphertext: `01110110100010011011001100001011001001100010001110110101001111000111010011110111000011100110011011001010111000010100010000110001`

The ciphertext was then converted into encoded DNA using the mapping table found in Table I below and further described in Section V.

TABLE I. MAPPING TABLE

5-bit	Codon	5-bit	Codon	5-bit	Codon	5-bit	Codon
00000	CAT	01000	TCG	10000	CAC	11000	GCC
00001	CCG	01001	CGC	10001	CGA	11001	CCC
00010	AGC	01010	TCC	10010	ATT	11010	GTA
00011	CGT	01011	GTG	10011	CGG	11011	GTT
00100	GAT	01100	ACC	10100	CAG	11100	TGG
00101	GAG	01101	AAC	10101	AAG	11101	CAA
00110	CTG	01110	GAA	10110	ACA	11110	GGA
00111	GTC	01111	GCG	10111	GCA	11111	AGG

Each 5-bit ciphertext was converted into a codon consisting of a code sequence using 3 of the 4 nucleotides, `A`, `C`, `T`, and `G`. Two zero bits which are marked in bold in Table II below, were added to the last 5-bit ciphertext in order to complete a 5-bit sequence.

To create a synthesized DNA sequence that is indistinguishable from the living organism's DNA sequences, fake data is embedded into an encoded DNA sequence based on statistical frequency analysis. The method described in Section VI is used to determine what fake data will be embedded. The fake data is inserted into the encoded DNA sequence in randomly selected positions. Fake data is marked in bold in Table III.

TABLE II. ENCODED DNA

01110 GAA	11010 GTA	00100 GAT	11011 GTT	00110 CTG	00010 AGC
11001 CCC	00110 CTG	00100 GAT	01110 GAA	11010 GTA	10011 CGG
11000 GCC	11101 CAA	00111 GTC	10111 GCA	00001 CCG	11001 CCC
10011 CGG	01100 ACC	10101 AAG	11000 GCC	01010 TCC	00100 GAT
00110 CTG	001**00** GAT				

TABLE III. SYNTHESIZED DNA

GAA	GTA	**TTC**	GAT	**AAA**	GTT	**GGG**	CTG
AGC	CCC	**GCT**	CTG	GAT	GAA	**TGT**	**GGT**
GTA	CGG	**CTC**	GCC	CAA	**GGG**	**CCA**	**TTG**
AGA	GTC	GCA	CCG	CCC	**TGC**	**GGT**	CGG
GGG	ACC	**GAC**	AAG	GCC	**GGC**	**CTT**	**AGT**
TCC	GAT	CTG	GAT				

Finally, we have a plaintext and its corresponding ciphertext, encoded DNA, and Synthesized DNA as shown in table IV.

TABLE IV. MESSAGE TO SYNTHESIZED DNA

Plaintext	HelloMyNameIsLHL
Ciphertext	0111011010001001101100110000101100100110001000111011010100111100011101001111011100001110011001101100101011100001010001000 0110001
Encoded DNA	GAAGTAGATGTTCTGAGCCCCCTGGATGAAGTACGGGCCCAAGTCGCACCGCCCCGGACCAAGGCCTCCGATCTGGAT
Synthesized DNA	GAAGTATTCGATAAAGTTGGGCTGAGCCCCGCTCTGGATGAATGTGGTGTACGGCTCGCCCAAGGGCCATTGAGAGTCGCACCGCCCTGCGGTCGGGGGACCGACAAGGCCGGCCTTAGTTCCGATCTGGAT

V. AN ALGORITHM TO CREATE A MAPPING TABLE

The core module of our attack model is to create the mapping table. There are three steps to making the mapping table described in this section:

1) **Ciphertext Distribution**: After dividing the ciphertext into 5-bit values, analyze the probability distribution of 5-bit values.
2) **Codon Distribution**: Analyze the probability distribution of codons in non-coding regions of the targeted living organism.
3) **Mapping Table Creation**: Create the mapping table based on the probability distributions of the 5-bit values and codons.

A. Ciphertext Distribution

We assume that a ciphertext $\mathbf{c} = (c_1, ..., c_{128}, c_{129}, c_{130})$, where $(c_1, ..., c_{128})$ is a 128-bit ciphertext from the AES-128 encryption, and $c_{129} = c_{130} = 0$. For example, the ciphertext, $\mathbf{c} = (c_1, c_2, c_3, ..., c_{128}, c_{129}, c_{130}) = (0, 1, 1, ..., 1, 0, 0)$. There are m ciphertexts, $\mathbf{c}^i = (c^i_1, ..., c^i_{130})$, where $1 \le i \le m$. Let $\mathbf{d}^i = (d^i_{00000}, ..., d^i_{11111})$, where d^i_j is a probability of each 5-bit values j in the i ciphertexts from $\mathbf{c}^1$ to $\mathbf{c}^i$. For example, the probability distribution of 5-bit values in the ciphertext in Table IV is as follows:

TABLE V. Probability Distribution of Ciphertext in Table IV

j	d_j	j	d_j	j	d_j	j	d_j
00000	-	01000	-	10000	-	11000	0.0769
00001	0.0385	01001	-	10001	-	11001	0.0769
00010	0.0385	01010	0.0385	10010	-	11010	0.0769
00011	-	01011	-	10011	0.0769	11011	0.0385
00100	0.1538	01100	0.0385	10100	-	11100	-
00101	-	01101	-	10101	0.0385	11101	0.0385
00110	0.1154	01110	0.0769	10110	-	11110	-
00111	0.0385	01111	-	10111	0.0385	11111	-

We first determine the number of ciphertexts that would be used to make a probability distribution of the ciphertexts. We analyzed the distance between $\mathbf{d}^{i-1}$ and $\mathbf{d}^i$ where $1 \le i \le m$ using the following equation:

$$\begin{aligned} &dist(\mathbf{d}^{i-1} - \mathbf{d}^i) \\ &= \sqrt{(d^{i-1}_{00000} - d^i_{00000})^2 + ... + (d^{i-1}_{11111} - d^i_{11111})^2} \end{aligned} \quad (1)$$

The values of $dist(\mathbf{d}^{i-1} - \mathbf{d}^i)$ where $1 \le i \le 150$ are shown in Figure 6. We used random strings as plaintexts to get ciphertexts. We found that $dist(\mathbf{d}^{i-1} - \mathbf{d}^i) < 0.003$ where i is greater than 70. This means that the distance does not vary much where i is greater than 70. We selected $i = 87$ as the number of ciphertexts that are used to determine the probability distribution of ciphertexts, since 87 is greater than 70 and the variance of the probability distribution where $i = 87$ is 3.1909 which is the smallest, where $1 \le i \le 150$. The probability distribution of 5-bit values in 87 ciphertexts, from $\mathbf{c}^1$ to $\mathbf{c}^{87}$, is in Table VI.

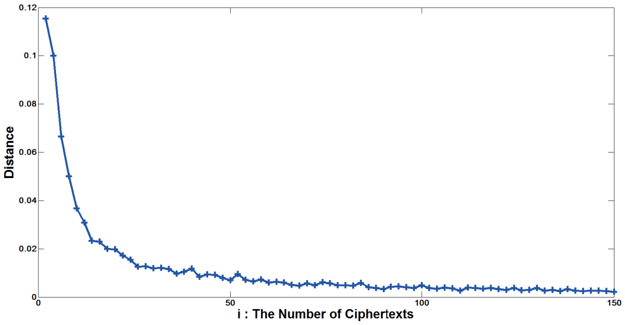


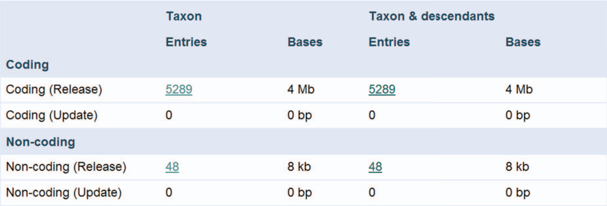
Fig. 6. $dist(\mathbf{d}^{i-1} - \mathbf{d}^i)$

TABLE VI. Probability Distribution of 5-bit values in 87 Ciphertexts

j	d^{87}_j	j	d^{87}_j	j	d^{87}_j	j	d^{87}_j
00000	0.0220	01000	0.0292	10000	0.0269	11000	0.0359
00001	0.0399	01001	0.0332	10001	0.0278	11001	0.0314
00010	0.0305	01010	0.0355	10010	0.0215	11010	0.0319
00011	0.0269	01011	0.0355	10011	0.0323	11011	0.0287
00100	0.0301	01100	0.0292	10100	0.0301	11100	0.0431
00101	0.0350	01101	0.0287	10101	0.0323	11101	0.0265
00110	0.0332	01110	0.0332	10110	0.0256	11110	0.0328
00111	0.0292	01111	0.0373	10111	0.0310	11111	0.0337

B. Codon Distribution

We used sequences of non-coding regions from Acetobacteraceae bacteria AT-5844 in Ensembl Genomes. There are 48 DNA fragments in the non-coding regions as shown in Figure 7. We analyze the probability distribution of codons in non-coding regions, and the analyzed distribution is depicted in Figure 8.



	Taxon		Taxon & descendants	
	Entries	Bases	Entries	Bases
Coding				
Coding (Release)	5289	4 Mb	5289	4 Mb
Coding (Update)	0	0 bp	0	0 bp
Non-coding				
Non-coding (Release)	48	8 kb	48	8 kb
Non-coding (Update)	0	0 bp	0	0 bp

Fig. 7. Coding and Non-Coding Regions in Acetobacteraceae Bacteria

C. Mapping Table Creation

Figure 9 shows how to create a mapping table based on the probability distributions of 5-bit values in 87 ciphertexts and codons in non-coding regions. We first sorted the data in ascending order based on d^{87}_j. Let b_i be the i-th probability of the sorted table in Figure 9, where $(1 \le i \le 32)$. If there are start and stop codons, `ATG`, `TAA`, `TAG`, and `TGA`, in synthesized DNA sequences, the synthesized DNA sequence that is inserted into non-coding regions might be mistaken as a coding-region. It could lead serious problems to the host organism's functionality. Therefore, start and stop codons should not be created in synthesized DNA sequences in our method. To do so, we randomly selected 32 codons from 60 codons which exclude start and stop codons from 64 codons. We then sorted the probability distribution in ascending order on probabilities as in the table in Figure 9. Let c_i be the i-th value that is scaled from each probability to make $c_1 + c_2 + ... + c_{32} = 1$.

Let $\mathbf{B} = (b_1, ..., b_{32})$, and $\mathbf{C} = (c_1, ..., c_{32})$. We define $dist(\mathbf{B} - \mathbf{C})$ as follows:

$$dist(\mathbf{B} - \mathbf{C}) = \sqrt{\sum_{i=1}^{32} (b_i - c_i)^2}. \quad (2)$$

We calculated the value, $dist(\mathbf{B} - \mathbf{C})$, using two sorted and scaled tables in Figure 9. In the same way, we performed 10,000 times to get the smallest value of $dist(\mathbf{B} - \mathbf{C})$ using randomly 32 selected codons. 32 codons that have the smallest

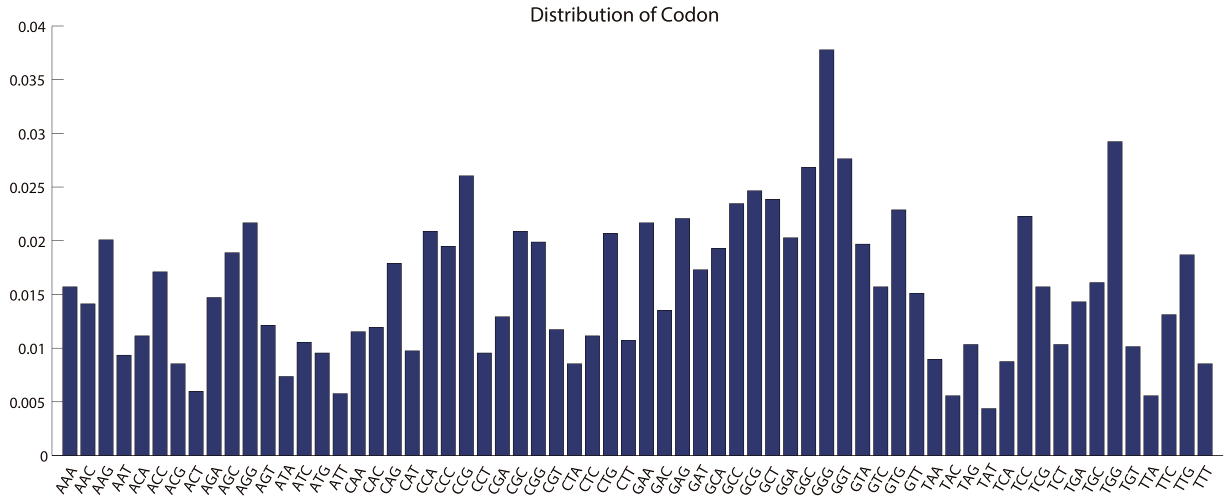
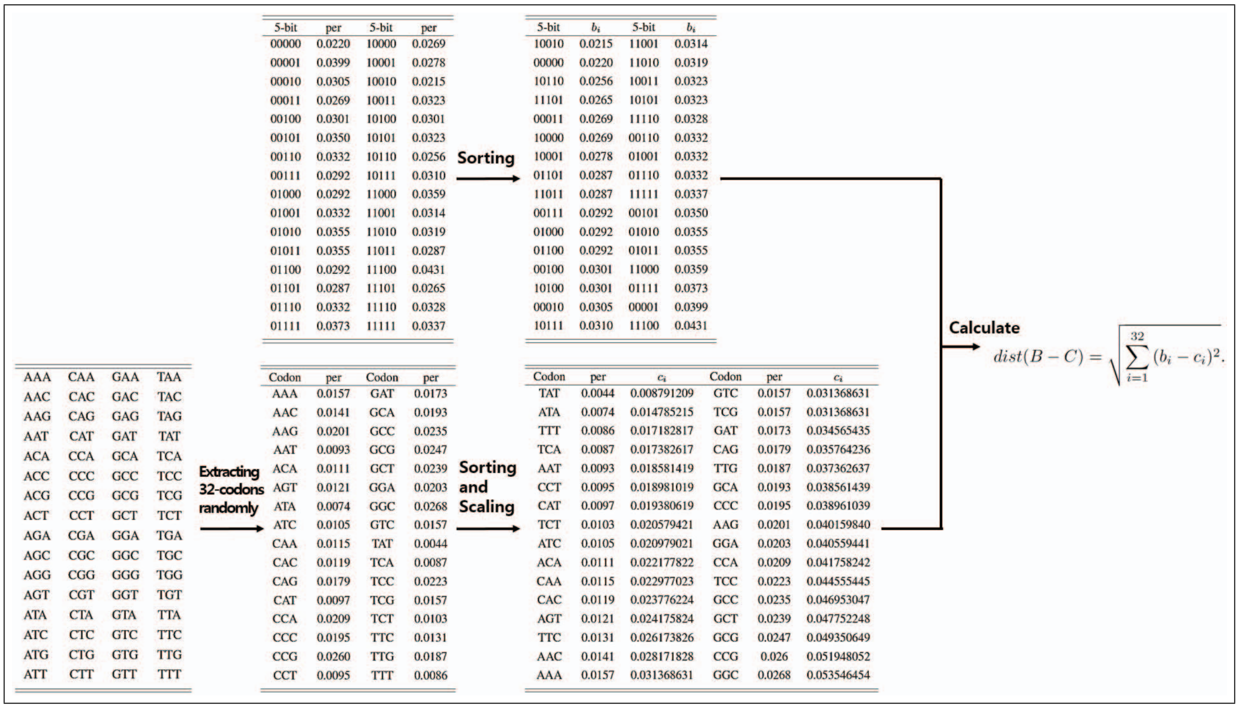


Fig. 8. Probability Distribution of Codons in Non-Coding Regions



5-bit	per	5-bit	per
00000	0.0220	10000	0.0269
00001	0.0399	10001	0.0278
00010	0.0305	10010	0.0215
00011	0.0269	10011	0.0323
00100	0.0301	10100	0.0301
00101	0.0350	10101	0.0323
00110	0.0332	10110	0.0256
00111	0.0292	10111	0.0310
01000	0.0292	11000	0.0359
01001	0.0332	11001	0.0314
01010	0.0355	11010	0.0319
01011	0.0355	11011	0.0287
01100	0.0292	11100	0.0431
01101	0.0287	11101	0.0265
01110	0.0332	11110	0.0328
01111	0.0373	11111	0.0337

5-bit	b_i	5-bit	b_i
10010	0.0215	11001	0.0314
00000	0.0220	11010	0.0319
10110	0.0256	10011	0.0323
11101	0.0265	10101	0.0323
00011	0.0269	11110	0.0328
10000	0.0269	00110	0.0332
10001	0.0278	01001	0.0332
01101	0.0287	01110	0.0332
11011	0.0287	11111	0.0337
00111	0.0292	00101	0.0350
01000	0.0292	01010	0.0355
01100	0.0292	01011	0.0355
00100	0.0301	11000	0.0359
10100	0.0301	01111	0.0373
00010	0.0305	00001	0.0399
10111	0.0310	11100	0.0431

$$dist(B-C) = \sqrt{\sum_{i=1}^{32} (b_i - c_i)^2}.$$

AAA	CAA	GAA	TAA
AAC	CAC	GAC	TAC
AAG	CAG	GAG	TAG
AAT	CAT	GAT	TAT
ACA	CCA	GCA	TCA
ACC	CCC	GCC	TCC
ACG	CCG	GCG	TCG
ACT	CCT	GCT	TCT
AGA	CGA	GGA	TGA
AGC	CGC	GGC	TGC
AGG	CGG	GGG	TGG
AGT	CGT	GGT	TGT
ATA	CTA	GTA	TTA
ATC	CTC	GTC	TTC
ATG	CTG	GTG	TTG
ATT	CTT	GTT	TTT

Codon	per	Codon	per
AAA	0.0157	GAT	0.0173
AAC	0.0141	GCA	0.0193
AAG	0.0201	GCC	0.0235
AAT	0.0093	GCG	0.0247
ACA	0.0111	GCT	0.0239
AGT	0.0121	GGA	0.0203
ATA	0.0074	GGC	0.0268
ATC	0.0105	GTC	0.0157
CAA	0.0115	TAT	0.0044
CAC	0.0119	TCA	0.0087
CAG	0.0179	TCC	0.0223
CAT	0.0097	TCG	0.0157
CCA	0.0209	TCT	0.0103
CCC	0.0195	TTC	0.0131
CCG	0.0260	TTG	0.0187
CCT	0.0095	TTT	0.0086

Codon	per	c_i	Codon	per	c_i
TAT	0.0044	0.008791209	GTC	0.0157	0.031368631
ATA	0.0074	0.014785215	TCG	0.0157	0.031368631
TTT	0.0086	0.017182817	GAT	0.0173	0.034565435
TCA	0.0087	0.017382617	CAG	0.0179	0.035764236
AAT	0.0093	0.018581419	TTG	0.0187	0.037362637
CCT	0.0095	0.018981019	GCA	0.0193	0.038561439
CAT	0.0097	0.019380619	CCC	0.0195	0.038961039
TCT	0.0103	0.020579421	AAG	0.0201	0.040159840
ATC	0.0105	0.020979021	GGA	0.0203	0.040559441
ACA	0.0111	0.022177822	CCA	0.0209	0.041758242
CAA	0.0115	0.022977023	TCC	0.0223	0.044555445
CAC	0.0119	0.023776224	GCC	0.0235	0.046953047
AGT	0.0121	0.024175824	GCT	0.0239	0.047752248
TTC	0.0131	0.026173826	GCG	0.0247	0.049350649
AAC	0.0141	0.028171828	CCG	0.026	0.051948052
AAA	0.0157	0.031368631	GGC	0.0268	0.053546454

Fig. 9. Mapping Table Creation

distance value among $10,000$ trials are used to make the mapping table, since it means that these two probability distributions are very similar. Therefore, we now obtain a mapping table in Table I.

VI. Fake Data Embedding

Our goal is to make synthesized DNA sequences that is indistinguishable from the living organism's DNA sequences. Since the mapping table uses 32 codons to encode 5-bit values instead of using whole codons, the result encoded DNA sequence is not very similar to the living organism's DNA sequences. Therefore, we need to insert more codons that are not used in the mapping table into the encoded DNA sequence to make the result synthesized sequence is indistinguishable from the living organism's DNA sequences. We first decided the number of fake codons that would be embedded. To get the number, we use the ratio between codons that are used and codons that are not used in the mapping table in Table VII. 26 codons are used to encrypt a 128-bit plaintext, therefore approximately 18 fake codons are inserted for indistinguishability.

TABLE VII. SUMS OF EACH CODONS

Start/Stop	per	Used	per	Not Used	per
ATG	0.0095	AAC	0.0141	AAA	0.0157
TAA	0.0089	AAG	0.0201	AAT	0.0093
TAG	0.0103	ACA	0.0111	ACG	0.0086
TGA	0.0143	ACC	0.0171	ACT	0.0060
		AGC	0.0189	AGA	0.0147
		AGG	0.0217	AGT	0.0121
		ATT	0.0058	ATA	0.0074
		CAA	0.0115	ATC	0.0105
		CAC	0.0119	CCA	0.0209
		CAG	0.0179	CCT	0.0095
		CAT	0.0097	CTA	0.0086
		CCC	0.0195	CTC	0.0111
		CCG	0.026	CTT	0.0107
		CGA	0.0129	GAC	0.0135
		CGC	0.0209	GCT	0.0239
		CGG	0.0199	GGC	0.0268
		CGT	0.0117	GGG	0.0378
		CTG	0.0207	GGT	0.0276
		GAA	0.0217	TAC	0.0056
		GAG	0.0221	TAT	0.0044
		GAT	0.0173	TCA	0.0087
		GCA	0.0193	TCT	0.0103
		GCC	0.0235	TGC	0.0161
		GCG	0.0247	TGT	0.0101
		GGA	0.0203	TTA	0.0056
		GTA	0.0197	TTC	0.0131
		GTC	0.0157	TTG	0.0187
		GTG	0.0229	TTT	0.0086
		GTT	0.0151		
		TCC	0.0223		
		TCG	0.0157		
		TGG	0.0292		
sum	**0.0430**	**sum**	**0.5809**	**sum**	**0.3759**

TABLE VIII. | B-C |

5-bit	count	prob.(B)	codon	prob.(C)	scaling(C)	| B-C |
00000	0	0	CAT	0.0097	0.01670	0.01670
00001	1	0.03846	CCG	0.0260	0.04476	0.00630
00010	1	0.03846	AGC	0.0189	0.03254	0.00592
00011	0	0	CGT	0.0117	0.02014	0.02014
00100	4	0.15384	GAT	0.0173	0.02978	0.12406
00101	0	0	GAG	0.0221	0.03804	0.03804
00110	3	0.11538	CTG	0.0207	0.03563	0.07975
00111	1	0.03846	GTC	0.0157	0.02703	0.01143
01000	0	0	TCG	0.0157	0.02703	0.02703
01001	0	0	CGC	0.0209	0.03598	0.03598
01010	**1**	**0.03846**	TCC	**0.0223**	**0.03839**	**0.00007**
01011	0	0	GTG	0.0229	0.03942	0.03942
01100	1	0.03846	ACC	0.0171	0.02944	0.00902
01101	0	0	AAC	0.0141	0.02427	0.02427
01110	2	0.07692	GAA	0.0217	0.03736	0.03756
01111	0	0	GCG	0.0247	0.04252	0.04252
10000	0	0	CAC	0.0119	0.02049	0.02049
10001	0	0	CGA	0.0129	0.02221	0.02221
10010	0	0	ATT	0.0058	0.00998	0.00998
10011	2	0.07692	CGG	0.0199	0.03426	0.04266
10100	0	0	CAG	0.0179	0.03081	0.03081
10101	1	0.03846	AAG	0.0201	0.03460	0.00386
10110	0	0	ACA	0.0111	0.01911	0.01911
10111	1	0.03846	GCA	0.0193	0.03322	0.00524
11000	2	0.07692	GCC	0.0235	0.04045	0.03647
11001	2	0.07692	CCC	0.0195	0.03357	0.04335
11010	2	0.07692	GTA	0.0197	0.03391	0.04301
11011	1	0.03846	GTT	0.0151	0.02599	0.01247
11100	0	0	TGG	0.0292	0.05027	0.05027
11101	1	0.03846	CAA	0.0115	0.01980	0.01866
11110	0	0	GGA	0.0203	0.03495	0.03495
11111	0	0	AGG	0.0217	0.03736	0.03736

TABLE IX. POSSIBLE NUMBERS OF EACH CODONS

codon	per(i)	x_i	codon	per(i)	x_i
AAA	0.0157	0.704035874	AAT	0.0093	0.417040359
ACG	0.0086	0.385650224	ACT	0.0060	0.269058296
AGA	0.0147	0.659192825	AGT	0.0121	0.542600897
ATA	0.0074	0.331838565	ATC	0.0105	0.470852018
CCA	0.0209	0.937219731	CCT	0.0095	0.426008969
CTA	0.0086	0.385650224	CTC	0.0111	0.497757848
CTT	0.0107	0.479820628	GAC	0.0135	0.605381166
GCT	0.0239	1.071748879	GGC	0.0268	1.201793722
GGG	0.0378	1.695067265	GGT	0.0276	1.237668161
TAC	0.0056	0.251121076	TAT	0.0044	0.197309417
TCA	0.0087	0.390134529	TCT	0.0103	0.461883408
TGC	0.0161	0.721973094	TGT	0.0101	0.452914798
TTA	0.0056	0.251121076	TTC	0.0131	0.587443946
TTG	0.0187	0.838565022	TTT	0.0086	0.385650224

We then needed to decide a possible number of each non-used codons that will appear in the synthesized DNA sequence. We picked TCC to get a probability of each non-used codons, since it has the smallest difference between probabilities of a 5-bit value and a codon as shown in Table VIII. The possible number x_i of each codons that are not used in the mapping table was calculated using the following equation:

$$1 : 0.0223 = x_i : per(i), \quad i \in \{1, 2, ..., 28\}$$

The numbers are shown in Table IX. For example, since the number of GCT is 1.071748879, the codon GCT is likely to appear once in a synthesized DNA sequence. 18 codons that are marked in bold in Table III were selected based on the numbers in Table IX.

VII. ANALYSIS

We analyze our method in this section. We show that synthesized DNA sequences that are created using our method are indistinguishable from DNA sequences in the host bacteria. We first created synthesized DNA sequences from 300 128-bit ciphertexts of AES-128, and then analyzed the probability distribution of codons in synthesized DNA sequences. Table X shows the distances between the densities of synthesized DNA sequences and host bacteria sequences. As a reference experiment, we also calculated the distances between the probability densities of DNA sequences in the host bacteria and in random sequences. We found that the former is always smaller than the latter. The p-value of this test is $p << 0.001$, confirming that the synthesized DNA sequences from our method are indistinguishable from the DNA sequences in the host bacteria. We also analyzed our method using longer ciphertexts: 100 1280-bit ciphertexts. This result also showed that synthesized DNA sequences from our method are similar to DNA sequences in the host bacteria.

There is a possibility that the data could be changed by random mutations. We can reduce the risk that we cannot retrieve the same message because of the mutations by inserting error correction algorithms such as Reed-Solomon codes [22] into synthesized DNA sequences. We can also reduce the risk by inserting the message redundantly into non-coding regions. We can use our method for a short-term storage, since it doesn't take much time to deliver the secret message using our method. Therefore, mutations might be rare during the communication. There is also a possibility that the inserting data might kill the host organism, therefore, we need to carefully select appropriate positions in non-coding regions.

VIII. Related Work

Recently, DNA watermarking [13]–[15], [17] and DNA steganography [7], [16], [23], [24] have been studied. DNA watermarking could be used to authenticate ownership and protect copyright of DNA sequences. J Craig Venter Institute embedded a watermarked DNA sequence which represents the initials of researchers in a synthesized bacterial genome and created the first cell with a synthesized genome [10], [11]. Hiding messages in DNA was first introduced by T. Clelland et al. in 1999 [7]. A brief message was successfully inserted in a sample human DNA. DNA-Crypt software that encodes a message in DNA code developed by Heider and Barnekow [13]. It employed error correction codes to detect errors.

DNA computing that is developed by Leonard Adleman in 1994 is a new research field that uses DNA molecule instead of using microchips in the silicon-based computer. DNA computers are faster and smaller than the traditional computers. Adleman used DNA computing to solve the directed Hamilton path problem which also known as the traveling salesman problem. Ogihara and Ray showed that DNA computers can simulate Boolean gates [20]. DNA computers can also be used for parallel processing. DNA data storage [2], [26] that is long-term and high density can store approximately 200 novels in a DNA microchip [2].

IX. Conclusion

We have addressed the potential for using DNA steganography as a new, stealthy cyber-attack to bypass systems that screen for electronic devices. We have shown the ease with which DNA-courier attacks could be carried out, whereby anyone could send secret messages to an intended recipient without being noticed by third parties. Accordingly, future work is needed to design counterattack measures.

References

[1] J. Aron. Spies could hide messages in gene-modified microbes, 26 September 2011. http://www.newscientist.com/article/dn20965-spies-could-hide-messages-in-genemodified-microbes.html#.U9Rer_kemck.

[2] C. Bancroft, T. Bowler, B. Bloom, and C. T. Clelland. Long-term storage of information in DNA. *Science (New York, NY)*, 293(5536):1763–1765, 2001.

[3] M. B. Beck, E. C. Rouchka, and R. V. Yampolskiy. Finding Data in DNA: Computer Forensic Investigations of Living Organisms. In *Digital Forensics and Cyber Crime*, pages 204–219. Springer, 2013.

[4] J. Boldt and O. Muller. Newtons of the leaves of grass. *Nature Biotechnology*, 26:387–389, 2008.

[5] E. Check. Synthetic biologists face up to security issues. *Nature*, 436:894–895, 2005.

[6] E. Check. Synthetic biologists try to calm fears. *Nature*, 441:388–389, 2006.

[7] C. T. Clelland, V. Risca, and C. Bancroft. Hiding messages in DNA microdots. *Nature*, 399(6736):533–534, 1999.

[8] T. W. T. S. I. EBI. Ensemblgenomes. http://ensemblgenomes.org/.

[9] ENCODE Project Consortium and others. An integrated encyclopedia of DNA elements in the human genome. *Nature*, 489(7414):57–74, 2012.

[10] D. G. Gibson, G. A. Benders, C. Andrews-Pfannkoch, E. A. Denisova, H. Baden-Tillson, J. Zaveri, T. B. Stockwell, A. Brownley, D. W. Thomas, M. A. Algire, et al. Complete chemical synthesis, assembly, and cloning of a mycoplasma genitalium genome. *science*, 319(5867):1215–1220, 2008.

[11] D. G. Gibson, J. I. Glass, C. Lartigue, V. N. Noskov, R.-Y. Chuang, M. A. Algire, G. A. Benders, M. G. Montague, L. Ma, M. M. Moodie, et al. Creation of a bacterial cell controlled by a chemically synthesized genome. *science*, 329(5987):52–56, 2010.

[12] L. M. Glenn. Ethical issues in genetic engineering and transgenics. http://www.actionbioscience.org/biotechnology/glenn.html.

[13] D. Heider and A. Barnekow. DNA-based watermarks using the DNA-crypt algorithm. *BMC bioinformatics*, 8(1):176, 2007.

[14] D. Heider and A. Barnekow. DNA Watermarking: Challenging perspectives for biotechnological applications. *Current Bioinformatics*, 6(3):375–382, 2011.

[15] D. Heider, M. Pyka, and A. Barnekow. DNA watermarks in non-coding regulatory sequences. *BMC research notes*, 2(1):125, 2009.

[16] A. Khalifa. LSBase: A key encapsulation scheme to improve hybrid crypto-systems using DNA steganography. In *Computer Engineering & Systems (ICCES), 2013 8th International Conference on*, pages 105–110. IEEE, 2013.

[17] M. Liss, D. Daubert, K. Brunner, K. Kliche, U. Hammes, A. Leiherer, and R. Wagner. Embedding permanent watermarks in synthetic genes. *PloS one*, 7(8):e42465, 2012.

[18] X. Miao. Recent advances in the development of new transgenic animal technology. *Cellular and Molecular Life Sciences*, 70(5):815–828, 2013.

[19] A. J. Newson. Current ethical issues in synthetic biology: Where should we go from here? *Accountability in Research*, 18(3):181–193, 2011.

[20] M. Ogihara and A. Ray. Simulating boolean circuits on a DNA computer. *Algorithmica*, 25(2-3):239–250, 1999.

[21] E. Parens, J. Johnston, and J. Moses. Do we need "synthetic bioethics"? *Science*, 321(5895):1449, 2008.

[22] I. S. Reed and G. Solomon. Polynomial codes over certain finite fields. *Journal of the Society for Industrial & Applied Mathematics*, 8(2):300–304, 1960.

[23] V. I. Risca. DNA-based steganography. *Cryptologia*, 25(1):37–49, 2001.

[24] B. Shimanovsky, J. Feng, and M. Potkonjak. Hiding data in DNA. In *Information Hiding*, pages 373–386. Springer, 2003.

[25] N.-F. Standard. Announcing the advanced encryption standard (AES). *Federal Information Processing Standards Publication*, 197, 2001.

[26] P. C. Wong, K.-k. Wong, and H. Foote. Organic data memory using the DNA approach. *Communications of the ACM*, 46(1):95–98, 2003.

Appendix

TABLE X. Distances between Distributions of 300 128-bit ciphertexts

index	distance	index	distance	index	distance	index	distance	index	distance	index	distance
1	0.147512146	51	0.041903218	101	0.038570921	151	0.039972056	201	0.038615788	251	0.038089161
2	0.103495475	52	0.042204974	102	0.038474803	152	0.039865555	202	0.038545310	252	0.038135240
3	0.083489638	53	0.041838386	103	0.038413539	153	0.039825821	203	0.038534052	253	0.038134848
4	0.076118582	54	0.041575270	104	0.038342586	154	0.039695433	204	0.038498995	254	0.038189659
5	0.066798393	55	0.041234766	105	0.038413883	155	0.039588052	205	0.038372885	255	0.038128359
6	0.069972110	56	0.041427463	106	0.038490170	156	0.039471098	206	0.038404579	256	0.038033437
7	0.064892752	57	0.040629617	107	0.038487142	157	0.039479582	207	0.038374844	257	0.038119120
8	0.064063200	58	0.040657218	108	0.038385750	158	0.039407945	208	0.038351857	258	0.038156715
9	0.063488816	59	0.040131339	109	0.038180429	159	0.039385959	209	0.038289238	259	0.038088547
10	0.058829017	60	0.039847343	110	0.038219193	160	0.039246814	210	0.038189676	260	0.037987194
11	0.056792348	61	0.040094551	111	0.038494083	161	0.039170117	211	0.038300130	261	0.037906019
12	0.052773501	62	0.040191244	112	0.038557446	162	0.038959807	212	0.038230055	262	0.037942159
13	0.054774254	63	0.040211873	113	0.038606747	163	0.038854255	213	0.038175889	263	0.038057706
14	0.052489220	64	0.040106977	114	0.038654951	164	0.038964422	214	0.038123121	264	0.038131382
15	0.051083249	65	0.039795804	115	0.038519866	165	0.038821471	215	0.038265282	265	0.038063664
16	0.049777335	66	0.039718408	116	0.038574641	166	0.038718751	216	0.038249037	266	0.037979657
17	0.049815946	67	0.039586451	117	0.038779953	167	0.038679502	217	0.038326313	267	0.037982335
18	0.049149850	68	0.039732932	118	0.038609644	168	0.038716610	218	0.038415099	268	0.038041978
19	0.047408260	69	0.039588800	119	0.038635711	169	0.038666960	219	0.038500088	269	0.038018959
20	0.047669146	70	0.040001024	120	0.038622010	170	0.038732105	220	0.038523137	270	0.037918725
21	0.046216477	71	0.039873525	121	0.038614738	171	0.038658320	221	0.038525784	271	0.037938758
22	0.043680909	72	0.039843316	122	0.038676113	172	0.038519708	222	0.038607481	272	0.037791467
23	0.044777654	73	0.040272782	123	0.038862975	173	0.038606363	223	0.038531322	273	0.037700072
24	0.043720564	74	0.040181571	124	0.038770740	174	0.038587770	224	0.038576758	274	0.037682076
25	0.042982417	75	0.040088972	125	0.038970408	175	0.038582622	225	0.038522133	275	0.037618073
26	0.042840320	76	0.040065162	126	0.038920653	176	0.038670976	226	0.038543058	276	0.037589763
27	0.043330056	77	0.039909681	127	0.038752860	177	0.038532634	227	0.038537544	277	0.037652174
28	0.043471272	78	0.040065871	128	0.038861574	178	0.038591730	228	0.038513551	278	0.037531353
29	0.043588112	79	0.039562417	129	0.038887731	179	0.038603943	229	0.038550168	279	0.037474572
30	0.043902968	80	0.039684587	130	0.039049517	180	0.038749163	230	0.038511791	280	0.037387368
31	0.043691989	81	0.039635192	131	0.039119179	181	0.038800480	231	0.038417135	281	0.037343038
32	0.044866708	82	0.039511106	132	0.039073945	182	0.038857691	232	0.038464868	282	0.037292969
33	0.044554084	83	0.039747095	133	0.039062753	183	0.038838124	233	0.038361360	283	0.037384476
34	0.044430810	84	0.039610045	134	0.039068873	184	0.038859981	234	0.038264198	284	0.037478176
35	0.044196173	85	0.039660120	135	0.039082438	185	0.038870406	235	0.038183305	285	0.037457702
36	0.044418109	86	0.039485056	136	0.039232338	186	0.038859923	236	0.038178614	286	0.037439128
37	0.044154948	87	0.039256312	137	0.039480368	187	0.038949816	237	0.038124453	287	0.037491050
38	0.044340622	88	0.038846529	138	0.039573471	188	0.038957515	238	0.038156148	288	0.037345551
39	0.044612811	89	0.038896204	139	0.039536819	189	0.038845082	239	0.038044631	289	0.037298590
40	0.044160185	90	0.039254015	140	0.039547381	190	0.038837864	240	0.038157654	290	0.037281122
41	0.044034310	91	0.039046109	141	0.039837397	191	0.038772069	241	0.038248956	291	0.037355257
42	0.043311780	92	0.039206282	142	0.039838927	192	0.038774664	242	0.038150057	292	0.037368860
43	0.043036271	93	0.038757619	143	0.039668253	193	0.038828015	243	0.038188741	293	0.037358143
44	0.042368913	94	0.039038682	144	0.039880325	194	0.038837768	244	0.038158533	294	0.037375233
45	0.041804901	95	0.039216698	145	0.039840312	195	0.038796842	245	0.038157977	295	0.037317911
46	0.041198100	96	0.038910734	146	0.040019519	196	0.038748500	246	0.038044556	296	0.037227836
47	0.041137300	97	0.038585420	147	0.040313541	197	0.038767627	247	0.038083456	297	0.037197418
48	0.041084762	98	0.038376440	148	0.040235891	198	0.038779532	248	0.038024775	298	0.037097300
49	0.041178632	99	0.038423239	149	0.040114457	199	0.038690794	249	0.038044936	299	0.037079497
50	0.041490700	100	0.038409775	150	0.040009092	200	0.038659371	250	0.038081366	300	0.037037551