Security Study of Keyed DNA Data Embedding

David Haughton and Félix Balado
School of Computer Science and Informatics
University College Dublin
Dublin, Ireland

Abstract—The emergent field of DNA data embedding has, in the past decade, seen the proposal of many methods to encode data in the genome of living organisms. These algorithms have primarily focused on incorporating error correction into encoded data. However the security aspects of data encoded in DNA have generally been overlooked. As with any data hiding method, information encoded in DNA can be protected with a secret key known only to authorized individuals. While there have been suggestions to incorporate keys, no well defined functions for mapping keys to parameters have been proposed yet. Furthermore there have been no security analyses of keyed DNA data embedding methods, in terms of quantifying how difficult it is for an attacker to find the key. In this paper, modifications to incorporate secret keys into a prominent embedding algorithm are proposed, and the security is analysed under realistic attack scenarios. The algorithm to be modified, called BioCode ncDNA has been proposed by us, and is designed for encoding data in vivo (in the cells of living organism) with high biocompatibility and near-optimum embedding rate.

Index Terms—Keyed DNA data embedding, BioCode, security aspects, key leakage, key estimation

I. INTRODUCTION

Recent efforts in the field of DNA data embedding have focused on encoding data robustly in the DNA of living beings. In this paper we consider instead how encoded data may be be protected against unauthorized access, which is an aspect not previously studied in this area. As an example of the relevance of security in DNA data embedding, consider the use of these techniques to encode ownership or tracking information. As an example of how encoded data may be be protected against unauthorized access, which is an aspect not previously studied in this area. As an example of the relevance of security in DNA data embedding, consider the use of these techniques to encode ownership or tracking information.

Embedding algorithms specifically oriented to using the DNA of living beings can also incorporate secret keys. However most proposed methods amount to using fixed keys, which can hence be assumed to be publicly known, and do not feature reliable mechanisms to secure them. In this paper we provide a method to secure the BioCode ncDNA algorithm using maximally long binary key sequences, and we analyse the security in terms of key leakage and key estimation from the view of an attacker seeking to discover secrets of the system.

Notation and Framework: In the following sets will be represented by upper-case calligraphic letters. DNA is composed of fundamental units called nucleotide bases and the set of possible bases is is given by \( \mathcal{X} = \{A, C, T, G\} \). DNA sequences are double stranded, with one strand defined by its complementary via Watson-Crick pairings (A-T, C-G). Row vectors are represented by lower-case bold letters with bars. A as ciphertext; we will keep this nomenclature here. Random variables are given by upper-case italicized letters. Probabilities are denoted as \( \text{Pr}(\cdot) \), the entropy function as \( H(\cdot) \) and the expected value as \( E(\cdot) \).

DNA contains two distinct region types, referred to as protein coding DNA (pcDNA) and non protein coding DNA (ncDNA). pcDNA encodes amino acids, which are fundamental to the development of organisms. A three base sequence, known as a codon, codes for an amino acid. There are three special codons, called start codons which instruct the genetic machinery to begin translation of DNA to amino acids.

The primary aim of DNA data embedding is to encode data without changing the biological processes of the host organism. In the past, ncDNA was thought to have no function, however recent research suggests that up to 80% of ncDNA may be responsible for regulatory functions. It is in this remaining 20% of ncDNA that ncDNA data embedding algorithms are designed to operate. To preserve this lack of function, a goal of ncDNA data embedding algorithms is to prevent the artificial creation of start codons. The only method to date which guarantees encoding data under this constraint...
is BioCode ncDNA \[5\].

The no start codons constraint requires that start codons may not appear in either of the two DNA strands. The set of start codons for eukaryotic organisms and their complementary codons in the opposite strand is \( C = \{ \text{ATG}, \text{CTG}, \text{TTG}, \text{CAT}, \text{CAG}, \text{CAA} \} \). The set of bases which make up the first two bases in \( C \) is given by \( D = \{ \text{AT}, \text{CT}, \text{TT}, \text{CA} \} \). If an element from \( D \) appears in a DNA sequence then the next possible base may form a start codon.

The security of keyed BioCode ncDNA will be analysed under two attack scenarios which were originally proposed to study cryptanalysis, and which have also been used in data hiding research \[7\]:

1. Watermark only attack (WOA), in which the attacker has access to \( n \) instances of \( \bar{y} \) alone, denoted \( \bar{y}^{(1:n)} = [\bar{y}^{(1)}, \bar{y}^{(2)}, \ldots, \bar{y}^{(n)}] \). As we will see in the following section, each \( y_i \in \bar{y} \) is encoded with separate parameters, corresponding to part of the key. Therefore only other encoded DNA sequences, constructed using the same key, aid the attacker.

2. Known message attack (KMA), which assumes that the attacker has access to pairs of \( \bar{y} \) and the corresponding \( \bar{m} \). As with the previous notation, we will denote the vector of pairs which contribute to obtaining the key for base \( y_i \) as \( \bar{y}^{(i)} = [\bar{y}^{(1)}, \bar{y}^{(2)}, \ldots, \bar{y}^{(n)}] \). For notational convenience the subscript \( i \) will be excluded in the remainder of the paper. A vector of random encoded bases is given by \( \bar{Y} \) and the set of all encoded sequences of length \( n \) is \( \bar{Y}^n \).

The unconstrained embedding case assumes that bases may be freely appended. Let \( \mathcal{M} \) denote a set containing binary digits and \( \mathcal{S} \) a set containing bases. Under these assumptions, a mapping of \( \mathcal{M} \rightarrow \mathcal{S} \) such as \{00, 01, 10, 11\} \( \rightarrow \{\text{A, C, T, G}\} \) is possible. This mapping can be seen particular permutation of the elements in \( \mathcal{S} \), and thus as one particular key. Then the number of possible keys per encoded base is \( |\mathcal{S}|! = 5184 \), giving a key length of 12.34 bits/base/key. In order to denote a permutation of a table column we can use a vector in which the value at each index denotes the position of the element at the corresponding index in \( \mathcal{S} \). For example as shown Table \[\ref{tab:encoding}\] the “AT” column contains \( S_{\text{AT}} = \{\text{A, T, C}\} \) which corresponds to the permutation \( \sigma_{\text{AT}} = [0, 1, 2] \). Thus we need four such permutation vectors per encoded base to represent the key.

### II. Embedding Algorithms

#### A. Unconstrained ncDNA Embedding

The unconstrained embedding case assumes that bases may be freely appended. Let \( \mathcal{M} \) denote a set containing binary digits and \( \mathcal{S} \) a set containing bases. Under these assumptions, a mapping of \( \mathcal{M} \rightarrow \mathcal{S} \) such as \{00, 01, 10, 11\} \( \rightarrow \{\text{A, C, T, G}\} \) is possible. This mapping can be seen particular permutation of the elements in \( \mathcal{S} \), and thus as one particular key. Then the number of possible keys per encoded base is \( |\mathcal{S}|! = 24 \).

According to the discussion above this allows for a key length of \( \log 24 \) bits/key/base = 4.58 bits/key/base. However we still need to determine how to map a permutation to a bit sequence representing the key. A lookup table is unfeasible because it grows exponentially with the length of \( \bar{y} \). Fortunately, to map a binary key sequence to an actual permutation we can use adaptive arithmetic encoding as described in \[8\] Since this mapping is near-optimal a key length of \([4.58 \times \ell]\) bits/key is implementable, where \( \ell \) is the length of the encoding in bases.

#### B. BioCode ncDNA

BioCode ncDNA is a data embedding algorithm which prevents the creation of start codons in \( \bar{y} \). At each step \( i \) in the encoding process the trailing two bases \( \bar{d} = [y_{i-1}, y_{i-2}] \) are monitored. During construction of \( \bar{y} \) if \( \bar{d} \) has the potential to create a start codon, then the choice of \( y_i \) is limited to those bases which will not create a start codon. This amounts to using Table \[\ref{tab:encoding}\] at each point in the encoding process.

<table>
<thead>
<tr>
<th>( \bar{d} )</th>
<th>AT</th>
<th>CT</th>
<th>TT</th>
<th>CA</th>
<th>( {X^2 \setminus D} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( S_{\bar{d}} )</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>( \bar{y} )</td>
<td>A</td>
<td>A</td>
<td>A</td>
<td>C</td>
<td>A</td>
</tr>
<tr>
<td>( \bar{M} )</td>
<td>C</td>
<td>C</td>
<td>C</td>
<td>G</td>
<td></td>
</tr>
</tbody>
</table>

| \( \bar{M} \) | 0 | 0 | 0 | 00 |
|---|---|---|---|
| 10 | 10 | 10 | 01 |
| 11 | 11 | 11 | 10 |
| 00 |

### TABLE I: Encoding states of BioCode ncDNA. Given the trailing two bases \( \bar{d} \), the next base to be encoded is from \( S_{\bar{d}} \). Each bit message found in \( \bar{M} \) corresponds to a base in \( S_{\bar{d}} \). \( X^2 \) represents the set of all possible dinucleotides.

First, to define the relationship between a key and BioCode ncDNA we must consider all the states of the encoder given by Table \[\ref{tab:encoding}\]. These states correspond to the set of sets \( \mathcal{E} = \{\text{AT, CT, TT, } \{X^2 \setminus D\}\} \). The actual mapping given in Table \[\ref{tab:encoding}\] can be seen as a particular permutation of the elements in \( S_{\bar{d}} \), and thus as one particular key. The number of possible keys per encoded base is \( \prod_{\bar{d} \in \mathcal{S}} |\mathcal{S}_\bar{d}|! = 5184 \), giving a key length of 12.34 bits/base/key. In order to denote a permutation of a table column we can use a vector in which the value at each index denotes the position of the element at the corresponding index in \( S_{\bar{d}} \). For example as shown Table \[\ref{tab:encoding}\] the “AT” column contains \( S_{\text{AT}} = \{\text{A, T, C}\} \) which corresponds to the permutation \( \sigma_{\text{AT}} = [0, 1, 2] \). Thus we need four such permutation vectors per encoded base to represent the key.

### III. Security Analysis: WOA

The security analyses, provided in this section, proceeds by developing an information-theoretical key leakage measure for data embedding in DNA. The measure used is Shannon’s equivocation, which is simply a conditional entropy \( H(A|B) \) measuring the uncertainty about \( A \) given \( B \). A null equivocation means, in theory, that knowledge of \( B \) provides full knowledge of \( A \). Thus, for the WOA scenario we can use equivocation, as given below, to measure the security of embedded information.

\[
H(K|Y) = - \sum_{k \in \mathcal{K}} \sum_{y \in \mathcal{Y}^n} \Pr(y) \Pr(k|y) \log \Pr(k|y).
\]

#### A. Unconstrained ncDNA Embedding

Given that all keys have equal probability of being selected and that encoded bases are uniformly distributed, for the
unconstrained case Equation [1] becomes
\[
H(K|Y) = 24 \times 4^n \times \frac{1}{4^n} \times \frac{1}{24} \log \frac{1}{24} = 4.58 \text{ bits.} \quad (2)
\]

The above value remains the same regardless of how many samples of \( y \) an attacker has. Therefore there is no information leakage in this case.

B. BioCode ncDNA

For BioCode ncDNA there is a small amount of information leakage. Given a sufficient number of encoded sequences an attacker would be able to infer which base maps to a 0 for \( d = AT, d = CT \) and \( d = TT \). Observe that in Table I each of these states contain the message set \( M_d = \{0, 10, 11\} \). Due to \( \bar{m} \) being uniformly random, the probability of 0 occurring is \( \Pr(0) = \frac{1}{2} \) and therefore the probability of the corresponding base is also \( \frac{1}{2} \). The probability of the other two cases occurring is \( \Pr(10) = \frac{1}{4} \) and \( \Pr(11) = \frac{1}{4} \). Therefore in the worse case the probability of selecting the correct key is reduced from \( \Pr(0) = \frac{1}{2} \) to \( \Pr(K|Y) = \frac{1}{2}^{5184} \) to \( \Pr(K|Y) = \frac{1}{192} \).

IV. SECURITY ANALYSIS: KMA

For the KMA scenario, the key estimation error and equivocation will be used as metrics to evaluate the embedding methods. The probability functions developed in this section apply to both cases. Firstly, the calculation of \( \Pr(K|y, m) \) will be explained. We must note that identical \((m, y)\) pairs add no information more than a single pair. The function \( \beta(\cdot) \), given below, simply counts the number of unique \((m, y)\) pairs. Since \( m \to y \), this counting function gives the required result by examining \( y \) alone.

\[
\beta(y) = \sum_{s \in S} \mathbb{I}\left( \left\{ \sum_{i=1}^{n} \mathbb{I}\{y(i) = s\} > 0 \right\} \right),
\]

where \( \mathbb{I}\{A\} \) is the indicator function which evaluates to 1 if \( A \) is true and 0 otherwise. In the equation above the set \( S \) represents the symbols available for encoding. The cardinality of this set is denoted by \( u = |S| \). Given the definition of \( \beta(\cdot) \) above, it is clear that \( \beta(y) \leq u \). The probability of selecting the correct key given \( n \) pairs of \((m, y)\) is

\[
\Pr(K|y, m) = \frac{1}{(u - \beta(y))!}. \quad (3)
\]

Next, to compute \( \Pr(\beta(Y) = b|(Y, M)) \) let us first define a function \( \omega(\cdot, \cdot, \cdot) \) which gives the total number of possible rearrangements of \( b \) unique symbols. In this function, given next, the multinomial coefficient is used to compute all possible rearrangements of \( n \) symbols. Of these \( n \) symbols \( k_1, k_2, \ldots, k_b \) represent the count of each possible symbol in the encoding set. The result of summing over all multinomial coefficients, where \( k_i > 0 \), is the total number of possible rearrangements of \( n \) symbols, of which \( b \) are unique. This result must be multiplied by the number of ways this situation may arise, which is given by the binomial coefficient of \( \binom{n}{b} \).

\[
\omega(n, b, u) = \binom{u}{b} \times \sum_{k_1 + \cdots + k_b = n} \binom{n}{k_1, k_2, \ldots, k_b}.
\]

The number of rearrangements of an \( n \) length vector, using all possible symbols is \( u^n \), therefore to compute \( \Pr(\beta(Y) = b|(Y, M)) \), \( \omega(n, y, u) \) must be divided by \( u^n \).

\[
\Pr(\beta(Y) = b|(Y, M)) = \frac{\omega(n, b, u)}{u^n} \quad (4)
\]

A. Unconstrained ncDNA Embedding

If the probability \( \Pr(\beta(y)|(Y, M)) \) from the previous section is applied to the unconstrained case, the following set of equations are obtained,

\[
\begin{align*}
\Pr(\beta(Y) = 1|(Y, M)) &= \frac{1}{4^{n-1}} \quad (5) \\
\Pr(\beta(Y) = 2|(Y, M)) &= 6 \times \frac{2^{n} - 2}{4^{n}} \quad (6)
\end{align*}
\]

1) Key Estimation: Intuitively, the expected Hamming distance between keys, \( E(d_h(K, \tilde{K})|(Y, M)) \) shows how many bits of the key are revealed for a particular \((Y, M)\). As discussed previously, the key length is \( \log(4!) \) bits/key/base and, given \( \beta(y) \) unique \((m, y)\) pairs there will be \( \log(4! - \beta(y))! \) remaining bits of the key with unknown value. The expected Hamming distance of these remaining bits is 0.5. Therefore, \( E(d_h(K, \tilde{K})|(Y, M)) \) can then be written as

\[
E(d_h(K, \tilde{K})|(Y, M)) = \sum_{b=1}^{2} \Pr(\beta(Y) = b|(Y, M)) \times \frac{\log((4! - b)!)}{2 \times \log(4!)}. \quad (7)
\]

2) Equivocation: Given that \( y \) and \( m \) are uniformly distributed and every key has equal probability of being selected, incorporating Equation [3] into Equation [1] amounts to

\[
H(K|(Y, M)) = -\Pr(\beta(Y) = 1|(Y, M)) \log \frac{1}{6} - \Pr(\beta(Y) = 2|(Y, M)) \log \frac{1}{2}.
\]

B. BioCode ncDNA Embedding

In the unconstrained embedding case the encoder is constant in one encoding state. However, for BioCode ncDNA the encoder is always in one of 5 states, given by Table I. This must be factored into \( \Pr(\beta(y)|(Y, M)) \). To accomplish this the steady state probability of the encoder being in each of these states will be used. The probability of transitioning between the states in \( \mathcal{E} \) can be represented by a Markov chain. This may then be used to obtain the steady state probabilities, as detail in [5]. Let the set \( \mathcal{E}' \) represent a particular state, where \( \mathcal{E}' \in \mathcal{E} \) and \( u = |\mathcal{E}'| \). Then the steady state probability of \( \mathcal{E}' \) is \( \Pr(\mathcal{E}') \). The function \( \rho(\cdot) \), defined below, represents the sum of bases in \( n \) samples of ciphertext which belong to state \( \mathcal{E}' \).
The probability that exactly \( c = \rho(\mathcal{E}') \) samples of \( \mathcal{E}' \) arise is given by \( \Pr(\rho(E) = c; (Y, M), \mathcal{E}') \) below. It may be explained as the probability of that state occurring \( e \) times multiplied by the probability of it not occurring \( n - e \) times. This result is multiplied by the number of times this situation may occur, given by the binomial coefficient of \( \binom{n}{e} \).

\[
\Pr(\rho(E) = c; (Y, M), \mathcal{E}') = \binom{n}{e} \times (1 - \Pr(\rho(\mathcal{E}')))^{n-e} \times \Pr(\rho(\mathcal{E}'))
\]

The probability above may then be used to determine the probability of having \( b \) unique \((m, y)\) pairs as

\[
\Pr(\beta(Y) = b; (Y, M), \mathcal{E}') = \sum_{i=0}^{n} \Pr(\beta(Y) = b, (Y, M) \times (1 - \Pr(\rho(\mathcal{E}')))^{n-e} \times \Pr(\rho(\mathcal{E}'))
\]

where \((Y, M) \times (1-1)\) represents \( i \) random \((y, m)\) pairs.

1) Key Estimation: Adjusting Equation (7) to incorporate the probability that the encoder is in state \( \mathcal{E}' \) yields

\[
E(d_{k}(\hat{K}, K); (Y, M), \mathcal{E}') = \frac{\sum_{b=0}^{u-2} \log((u-b)!)}{2 \times \log(n!)} \times \Pr(\beta(Y) = b; (Y, M), \mathcal{E}')
\]

The formula above is then weighted by the amount each state contributes to the key and summed to give

\[
E(d_{k}(\hat{K}, K); (Y, M), \mathcal{E}') = \frac{\sum_{\mathcal{E}' \in \mathcal{E}} \Pr(\rho(\mathcal{E}'))}{4! + 3! \times 3} \times \Pr(K|\mathcal{M}, Y, \mathcal{E}') 
\]

2) Equivocation: Equivocation may then be obtained by summing the probability of having \( b \) pairs of \((m, y)\) multiplied by \( \log \frac{1}{u-b!} \) for every \( \mathcal{E}' \in \mathcal{E} \).

\[
H(K|Y, M) = \sum_{\mathcal{E}' \in \mathcal{E}} \frac{\sum_{b=0}^{u-2} \Pr(\rho(Y) = b; (Y, M), \mathcal{E}') \log \frac{1}{u-b!}}{4! + 3! \times 3}
\]

V. RESULTS AND CONCLUSION

The results shown in this section are for the KMA scenario. The empirical results have been obtained by encoding random messages and computing the required probabilities by examining \((m, y)\). Figure 1 and Figure 2 show that, on average, the attacker will be able to estimate the key for unconstrained ncDNA embedding after only 10 \((m, y)\) pairs. However BioCode ncDNA requires 110 \((m, y)\) pairs. This is a clear advantage of BioCode ncDNA over the unconstrained case.

In conclusion, this paper marks the first proposed method and analysis of keyed DNA data embedding. We have proposed modifications to one of the most relevant DNA data embedding algorithms presented to date. The theoretical analysis accurately models the empirical results as shown by Figure 1 and Figure 2. The analysis presented here for BioCode ncDNA can also be applied directly to BioCode pcDNA.

REFERENCES