Enhancing the Security of Data Hiding Using Double DNA Sequences

Fatma E. Ibrahim[#1], H. M.Abdalkader[*2], M. I. Moussa [#3]

[#]Computer Science Department,
Faculty of Computers and Informatics, Benha University,
Benha, Egypt
[1]fatma.elsayed@fci.bu.edu.eg
[3]mahmoud.mossa@fci.bu.edu.eg

*Information System Department,
Faculty of Computers and Information, Menofia University,
Shebien, Egypt
[2]hatem6803@yahoo.com

***Abstract*— Data hiding in deoxyribonucleic acid (DNA) emerges as an important topic in information security community. In this paper a new data hiding algorithm based on DNA sequence is proposed. The algorithm uses two DNA reference sequences; the first sequence is used for encrypting the secret message. The second sequence is used for hiding the cipher message. A DNA coding is used to encode the plaintext instead of the classical 8-bit ASCII coding. One advantage of the proposed algorithm is that; it has a low modification rate. In addition, the length of the DNA reference sequence remains unchanged. Experimental results show a better performance of the proposed algorithm with respect to several parameters such as capacity(C), payload and bit per nucleotide (bpn).**

***Index Terms*— DNA, Data hiding, Information security, DNA coding.**

I. Introduction

As networks grow and interconnect with other networks, including the Internet those networks are exposed to a greater number of security risks. Information security plays an important role in protecting the valuable electronic information of organizations and users. An important means of protecting information is to "scramble" it so that no one can read it. This scrambling is a process known as cryptography. Whereas cryptography scrambles a message steganography hides the existence of the data. Steganography technologies are a very important part of the future of Internet security and privacy on open systems such as the Internet. Steganography can use image files, audio files, Hypertext Markup Language (HTML), word and PowerPoint files or even video files to contain hidden information [1,2,3,4].

Recently, the characteristics of DNA computing turned the researchers to utilize DNA in many fields especially in security. In DNA sequence, there are four kinds of bases, which are adenine (A), cytosine (C), guanine (G) and thymine (T). Those four bases are attached to the sugar/phosphate to form the complete nucleotide. There are 64 possible 3-letter combinations of the DNA coding units T, C, A and G. The 3-letter combinations are called codons. The codons are used either to encode one of the 20 amino acids used in the synthesis of proteins or as one of the three stop codons that signals the end of a sequence. That produces multiple codons, so most amino acids being encoded by more than one codon. While DNA can be decoded unambiguously, it is not possible to predict a DNA sequence from its protein sequence. Because most amino acids have multiple codons, a number of possible DNA sequences might represent the same protein sequence.

The first pioneering step done in the field of using DNA computing returns back to Leonard Adleman [5]. Real DNA sequences possess some properties which make it a perfect media for hiding data. One special property of a DNA sequence is that, there is almost no difference between a real DNA sequence and a fake one. Another useful key element is that there are roughly 163 million DNA sequences openly usable. Based on these properties hiding data in DNA sequence has been attracting much attention and research work has been carried out to propose several new methods [6,7,8,9,01,11,12]. The idea of using DNA computing in the fields of cryptography and steganography is a possible technology that may bring forward a new hope for powerful, or even unbreakable, algorithms.

In this paper a new data hiding method based on doubly selection from the open usable DNA sequences is proposed. The proposed algorithm is a composition of two techniques Cryptography and Steganography. A pair of DNA reference sequences is randomly selected from publically available database. The first DNA reference sequence is used to encrypt the message. The other DNA sequence is used to paste the encrypted message letters in its repeated nucleotides. The proposed algorithm has a significant feature, the host DNA sequence could be reconstructed after the reverse operation. This property is not only ensured the security of the secret data

but also preserved the functionality of the original DNA sequence.

The rest of the paper is organized as follow. Section 2 briefly introduces the related work. Section 3 presents the proposed method. Security analysis and experimental results are given in Section 4. Finally; the conclusion is given in Section 5.

II. RELATED WORK

Some existing algorithms for data hiding based on DNA sequence can be divided into three methods: the insertion method, the complementary pair method and the substitution method. In these methods; the secret message is embedded into a reference DNA sequence resulting in a new reference sequence with data hidden. Each method has a disadvantage; insertion method increases the redundancy and expands the length of the DNA reference sequence. DNA sequence resulting from the substitution method has a highly modification rate and the complementary pair method expands the reference DNA sequence in the process of embedding the secret message [13].

A novel method used both encryption and data hiding together to communicate data securely was proposed in [14]. Firstly; the secret message is encrypted using DNA and Amino Acids-Based Playfair cipher. Secondly; the encrypted data is hidden into some reference DNA sequence using an insertion technique. In order to recover the embedded secret data, the receiver carries out the inverse process with the help of the both the secret key and the reference DNA sequence. Another approach is the complementary pair of rules of DNA which is used to hide data [15]. Both of sender and receiver have a DNA reference sequence. The binary message is encoded to DNA sequence and then applies a selected complementary pair rule. The index of each couple of nucleotides in DNA reference sequence is produced and the resulting numbers are sent as a cipher.

In [16] a modification of the original substitution method was introduced. The capacity of the updated version has been doubled. Original substitution method hides one secret bit per nucleotide where the modified one hides two secret bits per nucleotide. However, the modified algorithm increases the capacity it still suffer from a high modification rate in DNA sequence. This modification attracts the attention of attackers. Another improvement of the original substitution method is proposed to minimize modification rate in DNA reference sequence [17]. Some kind of injective mapping is established between one complementary rule and two secret bits. Based on this mapping mechanism, the hiding scheme can hide two secret bits by replacing one character. The improved algorithm maintains the length of the resulting faked DNA sequence; these issues ensure robustness and security of the hiding algorithm.

A different novel data hiding method based on DNA coding and using word document as host file is proposed in [18]. Firstly, the plaintext uses DNA coding instead of using 8-bit ASCII. Secondly, the method generates two random aided DNA sequences to encrypt and conceal the cipher sequence. Finally, it hides the cipher sequence into a word document by substituting the least significant 2-bit of the three color components. A novel reversible data hiding scheme based on histogram technique is used to transform the DNA sequence into a binary string and then combines several bits into a decimal integer. These decimal integers are used to generate a histogram [19]. Afterwards, the proposed scheme uses a histogram technique to embed secret data. This scheme has some key features it maintains the length of the DNA sequence used and has a small modification rate. In [20] a randomly chosen DNA sequence is used to hide secret message. This DNA sequence is indexed and used to hide secret message. The message is converted to binary using 8-bit ASCII representation then encoded to DNA using binary coding. Finally message index is produced and sent to the receiver.

In [21] a new data embedding method was designed to insert binary data into noncoding regions of DNA sequence in order to preserve its biological function. To reach a high standard of security and capacity, data is secured by cryptographic technique, translated, and then encoded accordingly into DNA sectors using a reference-dependent mechanism. Error correction method using Reed-Solomon encoder and parity check gives a good error handling against substitution and insertion or deletion mutations. This method gives comparably higher bit per nucleotide (bpn) value without disturbing host organism's protein synthesis process.

III. THE PROPOSED SCHEME

The main idea of the proposed scheme is to encrypt the secret message and hide it to ensure security and robustness. A double simultaneous random selection from a publically available DNA database S is taken. The resulting selection is a pair (S, S') which is a combination of two DNA reference sequences. The proposed algorithm consists of two phases. In the first phase, the secret message (P) is encoded to DNA sequence (DP) where each letter is replaced by three nucleotides. This replacement is known for both sender and receiver and kept secret. Algorithm 1 can be used to generate total 4^3= 64 DNA codons. The NUM_FORMAT is a combination of three digits and the DNA (NUM_FORMAT) transfers the resulting number to DNA codons (ex. 122 converts to CGG). Thus we describe a 1-1 map from all possible permutations between these 64 codons and the English alphabets (26 capital letters, 26 small letters), the numbers (0,1,..,9) and some punctuation marks. The first selected DNA sequence S is used for encryption of DP using XOR operation to produce the encrypted (DP'), XOR yields zero if the two given inputs are the same, and one otherwise. The ID of S is appended at the beginning of the resulting sequence to generate (IDsDP').

In the second phase, the other DNA sequence S' is used to hide the encrypted secret data. Where DNA sequences contain only four bases {A, C, T, G}, approximate repeats are well-known to frequently appear inside long DNA sequences. The proposed algorithm utilizes this property to hide the result from XOR operation (IDsDP') into the other DNA reference

sequence S` according to Table I. The first row of the table is the letter from the reference sequence S`, the msg column shows the encrypted secret message letter and the sbs column shows the substituted symbol with respect to the message. Some DNA sequences contain non-labeled nucleotide (N) as well as the other nucleotides. The number of this nucleotide is low. We avoid hiding (IDsDP') in N nucleotides to keep on the properties of the real DNA sequence. The new data hiding algorithm steps are shown as in Fig. 1.

Algorithm 1. Generating DNA codons

```
for i = 0 to 3
  for j = 0 to 3
    for k = 0 to 3
     do
         NUM_FORMAT = i j k
          codon = DNA(NUM_FORMAT)
    end for
   end for
 end for
```

We refer to nucleotides in DP' by {A(DP'), C(DP'), G(DP'), T(DP') }. Table I is used to hide DP' in ***S′*** by replacing the second repeated letter in ***S′*** with one of the four letters {A, C, G, T} according to the encrypted message. Accomplishing those replacements are done based on the following rules; (i) no replacement for the second repeated letter in ***S′*** when A(DP') is found, (ii) the second repeated letter replacement in S' is A-C, C-A, G-T, T-G when C(DP') is found, (iii) the second repeated letter replacement in ***S′*** is A-G, C-T, G-A, T-C when G(DP') is found, (iv) the second repeated letter replacement in ***S′*** is A-T, C-G, G-C, T-A when T(DP') is found.

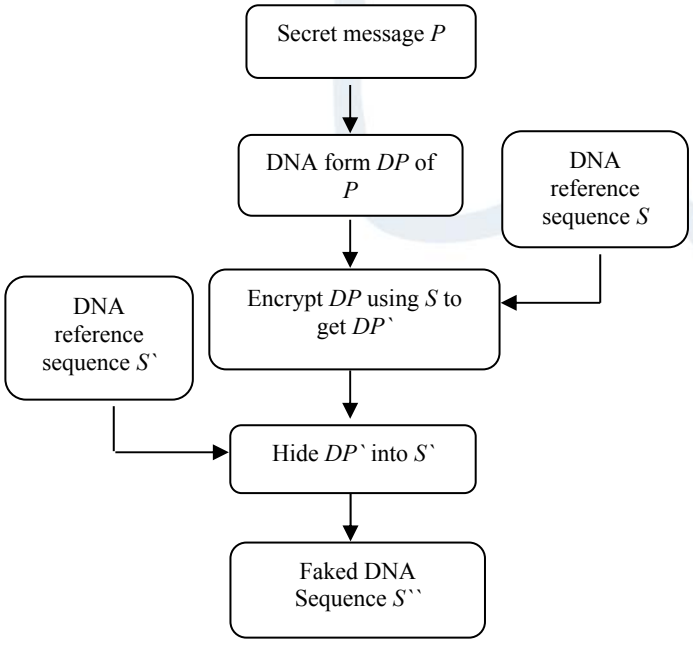

Fig. 1. Data Encryption and hiding digram.

TABLE I. Hiding and recovery table

A		C		G		T	
msg	sbs	msg	sbs	msg	sbs	msg	sbs
A	A	A	C	A	G	A	T
C	C	C	A	C	T	C	G
G	G	G	T	G	A	G	C
T	T	T	G	T	C	T	A

The ID of S` is appended at the beginning of the resulting sequence to generate (IDs' (IDsDP')') before sending the fake sequence to the receiver. The replacement rules in Table I is used in hiding processes and their inverses are used in recovery processes respectively.

A. Example Scenario

The following example illustrates step by step the encryption and hiding processes in details:
Let the secret message ***P*** be "Hello".
The pair of two DNA reference sequences are (S, S') such that;
S = "AGAGCAAGCCTTCTTCTTCCATA"

S`= A**A**T**T**C**C**A**A**AGA**A**ACAGACTCTACAGC**C**AGCGA**A**
G**G**CATG**G**AT**T**GCTG**G**GCA**A**ACAG**G**CA**A**AGAG**G**
AT**T**CG".

The detailed steps:

- Encode ***P*** to DP (Algorithm 1), where each letter in ***P*** is replaced by three DNA bases.
  DP = ACTCTGGCCGCCGGA
- Apply the binary XOR operation between DP and ***S*** and delete the extra unused nucleotide to get the following encrypted nucleotides DP`.
  DP` = DP XOR S = ATTTGGGTATGGTCT
- Append ID of S at the beginning of DP`
- Read S` and find out the second repeated nucleotide.
  S'= A**A**T**T**C**C**A**A**AGA**A**ACAGACTCTACAGC**C**AG
  CGA**A**G**G**CATG**G**AT**T**GCTG**G**GCA**A**ACAG**G**C
  A**A**AGAG**G**AT**T**CG".
- Apply the replacements rules in table 1 to get S``as follow:
  AATACGATAGAGACAGACTCTACAGCTAGCGAG
  GCCATGGATATGCTGAGCAGACAGCCACAGAGC
  ATTCG.
- Append ID of S` at the beginning of S`` and sent it to the receiver.

The data recovery process; for the receiver to extract the secret data he has to reverse the encryption and hiding steps. The receiver has to extract the IDs for both DNA reference used from the received sequence. The sender uses Algorithm 2 to encrypt and hide the secret message.

Algorithm 2. Data hiding algorithm

Preprocessing:

1. The set of all publicly DNA sequences $S = \{S_i \mid 1 <= i <= 163 \times 10^8\}$.
2. Consider the set of all unordered pairs of sequences of S $S \times S = \{(S, S') \mid S, S' \in S\}$.
3. Prepare DNA reference sequence S' by extracting all non-labeled nucleotide (N) if exist.

Input: Two DNA sequences (S, S`) and the secret message ***P***

Output: A faked DNA sequence S`` with secret message ***P*** hidden

Step1. Encode the secret message ***P*** to DNA sequence DP using Algorithm 1.

Step2. Generate the encrypted message DP` by performing the bitwise XOR of the secret message DP and the reference sequence S. *DP` = DP **XOR** S.*

Step3. Append the ID of S at the beginning of resulting DP` to get IDsDP'

Step4. Read the second reference DNA sequence S` and mark the second repeated characters.

Step5. Hide the encrypted message DP` in S ' using the replacement rules in Table I.
S`` ⟵ S`

Step6. Append the ID of S` to the beginning of resulting S`` = (IDs' (IDsDP')') and sent it to the receiver.

The sender sends the faked DNA sequence (IDs' (IDsDP')') without any other DNA sequences, DNA-like sequences, or any sequences of numbers to the receiver. The receiver processes the Data Recovery Algorithm 3 to recover the original message. The algorithm 3 shows the formal data recovery algorithm to extract secret message ***P***.

IV. SIMULATION RESULTS AND SECURITY ANALYSIS

A. Simulation Results

A number of experiments conducted and evaluated based on some parameters such as capacity, payload and bpn. The definition of capacity, denoted by ***C***, is the total length of the increased reference sequence after the secret message is hidden within it. The payload, denoted by ***P***, is the remaining length of the new sequence after extracting out the reference DNA sequence. The bpn is the number of bits hidden per character. We perform an experimental study of our algorithm using eight openly DNA sequences downloaded from National Center for Biotechnology Information (NCBI) database [22] and listed in Table II. The embedding capacity for the proposed algorithm for the eight DNA sequences used is listed in Table III.
Table IV shows the performance of the proposed algorithm in compare with Shiu et al.'s two schemes (Insertion method (Ins) and Substitution method (Sub)) and Guo et al.'s previous work. The parameter used in this table is bpn, which can be computed by bpn=|M|/C where |M| is length of the secret message, and C is the capacity.

Algorithm 3. Data Recovery Algorithm

Input: A faked DNA sequence S``= ((IDs' (IDsDP')')) with secret message hidden

Output: The secret message *P*.

Step1. Extract the first bases that represent ID of S` which the sender used to hide data.

Step2. Extract all non-labeled nucleotide (N) from S` if exist.

Step3. Find out the second repeated nucleotide in S'.

Step4. Extract DP` sequence from S`` using the replacement inverse rules in table 1

Step5. Extract the first bases form DP` that represent ID of S.

Step6. Decrypt DP` as follow: use commutative property of XOR
DP` XOR S = (DP XOR S) XOR S
= DP XOR (S XOR S)
= DP

Step7. Decode DP to letters where each three nucleotides represent an English alphabet.

Step8. Get plaintext *P*.

TABLE II. The tested DNA sequences

Locus	Specifies definition	No. of nucleotides
AC153526	Mus musculus10 BAC RP23-383C2	200,117
AC166252	Mus musculus6 BAC RP23-100G10	149,884
AC167221	Mus musculus10 BAC RP23-3P24	204,841
AC168874	Bos taurus clone CH240-209N9	206,488
AC168897	Bos taurus clone CH240-190B15	200,203
AC168901	Bos taurus clone CH240-18511	191,456
AC168907	Bos taurus clone CH240-19517	194,226
AC168908	Bos taurus clone CH240-95K23	218,028

According to Table IV the bpn is limited comparing with Shiu et al.'s two schemes because it is restricted to repeated characters only. The proposed scheme mainly hides data in repeated characters and also avoids hiding data in N nucleotides. The proposed method preserves the length of original DNA strand so the payload is always 0, similar with Shiu et al.'s substitution method and Guo et al.'s method.

TABLE III. The embedding capacity

Locus	Capacity C	No. of (N) Nucleotides	The embedding capacity (bit)
AC153526	200,117	0	115,392
AC166252	149,884	0	868,96
AC167221	204,841	0	115,392
AC168874	206,488	1300	115,560
AC168897	200,203	5186	113,184
AC168901	191,456	250	111,704
AC168907	194,226	809	112,648
AC168908	218,028	918	127,184

The modification rate of the fake DNA sequence for the proposed scheme is approx. 28.4% and this for Shiu et al.'s original substitution scheme is approx. 94.83%. The proposed scheme allows the receiver to obtain a better quality fake DNA sequence lower modification rate. This ensures the security of the proposed scheme. The proposed algorithm has many features; (i) no expand for the length of the DNA reference sequence used to embed data into, (ii) no change in the non-labeled nucleotide (N), (iii) a low modification rate compare with the previous works, (iv) The average of bpn is higher than bpn of Guo et al.'s previous works.

B. Security Analysis

The strength and robustness of the proposed algorithm are based on the following:

- The reference DNA sequence ***S*** used in encryption; there are roughly 163 million DNA sequences publicly available. Thus, the probability of an attacker making a successful guess is :

$$\frac{1}{1.63\times10^{8}}$$

- The reference DNA sequence ***S′*** which is used to hide the secret message, the probability of the attacker making a successful guess for the second selection ***S′*** is

$$\frac{1}{1.63\times10^{8}}$$

- Table I used in hiding (recovery), there are totally $24^4 = (4!)^4$ possible situations used for hiding (recovery), so the probability of an attacker making a successful guess is $1/24^4$.

- There are total p_{64}^{64} kinds of permutation between the English alphabets (26 capital letters, 26 small letters), ten digital (0,1,...,9), and two punctuation marks and 64 codons which are generated from algorithm 1.

Therefore the probability of guessing the secret message is:

$$\left(\frac{1}{1.63\times10^{8}}\right)^{2} \times \frac{1}{24^{4}} \times \frac{1}{p_{64}^{64}}.$$

V. CONCLUSION

In this paper a new data hiding algorithm based on DNA sequences has been proposed. Since DNA sequences have some interesting properties which is used for data hiding purposes. The proposed algorithm used a 6-bit DNA coding instead of 8-bit ASCII codes where each character of plaintext was denoted by three nucleotides. This coding increased the embedding capacity. The proposed scheme hide data in repeated characters this minimize the modification rate. Initially, the plaintext was encrypted using a DNA reference sequence. Then, the encrypted message was hidden into another DNA reference sequence. According to the security analysis, it is very difficult for attacker to identify the secret message. Simulation results showed a better performance for the proposed data hiding algorithm.

REFERENCES

[1] C. Chin-Chen, W. Wen-Chuan and C. Yi-Hui, "Joint coding and embedding techniques for multimedia images," Information Sciences, vol. 178, no. 18, pp. 3543-3556, 2008.

[2] H. Chun-Hsiang and W. Ja-Ling, "Fidelity-guaranteed robustness enhancement of blind-detection watermarking schemes," Information Sciences, vol. 179, no. 6, pp. 791-808, 2009.

[3] T. Hung- Hsu and S. Duen- Wu, "Color image watermark extraction based on support vector machines," Information Sciences, vol. 177, no. 2, pp. 550-569, 2007.

[4] T. Liu and W. Tsai, "A new steganographic method for data hiding in microsoft Word documents by a change tracking technique," IEEE Transactions on Information Forensics Security, vol. 2, no. 1, pp. 24-30, 2007.

[5] A. Leonard, "Molecular computation of solution to combinatorial problems," Science, vol. 266, pp. 1021-1024, 1994.

[6] C. Chang, T. C. Lu, Y. Chang and C. 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.

[7] C. Clelland, V. Risca and C. Bancroft, " Hiding messages in DNA microdots," Nature, vol. 399, pp. 533-534, 1999.

[8] A. Leier, C. Richter, W. Banzhaf and H. Rauhe, " Cryptography with DNA binary strands," BioSystems, vol. 57, no. 1, pp. 13-22, 2000.

[9] B. Shimanovsky, J. Feng and M. Potkonjak, "Hiding data in DNA," in Proc. of the 5th International Workshop on Information Hiding, LNCS, Netherlands., 2002.

[10] C. Jie, "A DNA-Based Biomolecular Cryptography Design," IEEE International Symposium on Circuits and Systems, vol. 3, pp. 822-825, 2003.

[11] H. Miki, K. Hiroaki and O. Kazuhiro, "Design of True Random One-Time Pads in DNA XOR Cryptosystem," Springer , vol. 2, pp. 174-183, 2010.

[12] E. I. Fatma, I. M. Mahmoud and S. A. Hatem, "A Symmetric Encryption Algorithm based on DNA Computing," International Journal of Computer Applications, vol. 97, no. 16, pp. 41-45, 2014.

[13] H. Shiu, K. Ng, J. Fang, R. Lee and C. Huang, " Data hiding methods based upon DNA sequences," Information Sciences, vol. 180, pp. 2196-2208, 2010.

[14] A. Khalifa, S. Reda and A. Atito, " DNA-Based Data Encryption and Hiding Using Playfair and Insertion Techniques," Journal of Communications and Computer Engineering , vol. 2, no. 3, pp. 44-49, 2012.

[15] R. A. Mohammad., N. Pourya., Ordi, A. and R. N. Mohammad. "DNA Base Data Hiding Algorithm," International Journal on New Computer Architectures and Their Applications, vol. 2, no. 1, pp. 183-192, 2012.

[16] J. Taur, H. Lin, H. Lee and C. Tao, " Data hiding in DNA sequences based on table lookup substitution," International Journal of Innovative Computing, Information and Control , vol. 8, no. 10, pp. 6585-6598, 2012.

[17] C. Guo, C. Chang and Z. Wang, " A new data hiding scheme based on DNA sequence," International Journal of Innovative Computing, Information and Contro, vol. 8, no. 1, pp. 136-149, 2012.

[18] H. Liua, D. Lin and A. Kadir, "A novel data hiding method based on deoxyribonucleic acid coding," Computers and Electrical Engineering , vol. 39, pp. 1164–1173., 2013.

[19] Y. Huang, C. Chang and W. CY, "A DNA-based data hiding technique with low modification rates," Springer Science+Business Media, 2012.

[20] B. Debnath, K. B. Samir, "Hiding Secret Data in DNA Sequence, " International Journal of Scientific & Engineering Research, vol. 4, no. 2, pp. 1-4, 2013.

[21] S. Kevin, K. KiRyong, L. SukHwan and K. Seong-Geun, "High Capacity Data Hiding Method in DNA with Mutation Handling," ACM, vol. 3, pp. 56-63, 2014.

[22] "http://www.ncbi.nlm.nih.gov," NCBI Database.

TABLE IV. Capacity, payload and bpn comparison with previous work

Locus	No. of nucleotides	Category	Shiu's Method		Guo et al.'s Method	Proposed Method
			Ins	Sub		
AC153526	200,117	Capacity	280,117	200,117	200,117	200,117
		Payload	80,000	0	0	0
		bpn	0.57	0.80	0.434	0.577
AC166252	149,884	Capacity	229,884	149,884	149,884	149,884
		Payload	80,000	0	0	0
		bpn	0.70	1.00	0.442	0.580
AC167221	204,841	Capacity	284,841	204,841	204,841	204,841
		Payload	80,000	0	0	0
		bpn	0.56	0.78	0.424	0.563
AC168874	206,488	Capacity	286,488	206,488	206,488	206,488
		Payload	80,000	0	0	0
		bpn	0.56	0.77	0.446	0.560
AC168897	200,203	Capacity	280,203	200,203	200,203	200,203
		Payload	80,000	0	0	0
		bpn	0.57	0.80	0.451	0.565
AC168901	191,456	Capacity	271,456	191,456	191,456	191,456
		Payload	80,000	0	0	0
		bpn	0.59	0.84	0.439	0.583
AC168907	194,226	Capacity	274,226	194,226	194,226	194,226
		Payload	80,000	0	0	0
		bpn	0.58	0.82	0.444	0.580
AC168908	218,028	Capacity	298,028	218,028	218,028	218,028
		Payload	80,000	0	0	0
		bpn	0.54	0.73	0.443	0.583