IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661,p-ISSN: 2278-8727, Volume 17, Issue 4, Ver. III (July – Aug. 2015), PP 31-34
www.iosrjournals.org

Enhancement of Security in DNA Based Cryptography

Ritu Mor* Praveen Kanth*
Research Scholar Department of Computer Science and Engineering, BRCM ,Bahal,Haryana,India
Assistant professor Department of Computer Science and Engineering,BRCM,Bahal,Haryana,India

Abstract: *DNA containing data obtained from more conventional binary storage media. Plaintext message data encoded in DNA strands by use of a (publicly known)* ***Key of*** *alphabet of short oligonucleotide sequences.*
Keywords: *Steganography, Decryption, cipher text, biological data.*

I. Introduction

A "wet" data base of biological data natural DNA obtained from biological sources may be recoded using nonstandard bases to allow for subsequent BMC processing.
DNA containing data obtained from more conventional binary storage media.
Input and output of the DNA data can be moved to conventional binary storage media by DNA chip arrays binary data may be encoded in DNA strands by use of an alphabet of short oligonucleotide sequences.

Associative Searches within DNA databases:

- Methods for fast associative searches within DNA databases using hybridization
- Database join operations and various massively parallel operations on the DNA data

II. DNA Storage Of Data

- A medium for ultra-compact information storage: large amounts of data that can be stored in compact volume.
- Vastly exceeds storage capacities of conventional electronic, magnetic, optical media.
- A gram of DNA contains 10^{21} DNA bases $= 10^{8}$ tera-bytes.
- A few grams of DNA may hold all data stored in world.
- Most recombinant DNA techniques are applied at concentrations of 5 grams of DNA per liter of water.

III. Area Of Applications Of Dna-Based Cryptography Systems

- the encryption of (recoded) natural DNA
- the encryption of DNA encoding binary data.

IV. Methods For 2d Data Input And Output:

- use of chip-based DNA micro-array technology
- transform between conventional binary storage media via (photo-sensitive and/or photo-emitting) DNA chip arrays.

V. Dna Steganography Systems:

- secretly tag the input DNA
- then disguise it (without further modifications) within collections of other DNA.
- original plaintext is not actually encrypted
- very appealing due to simplicity.

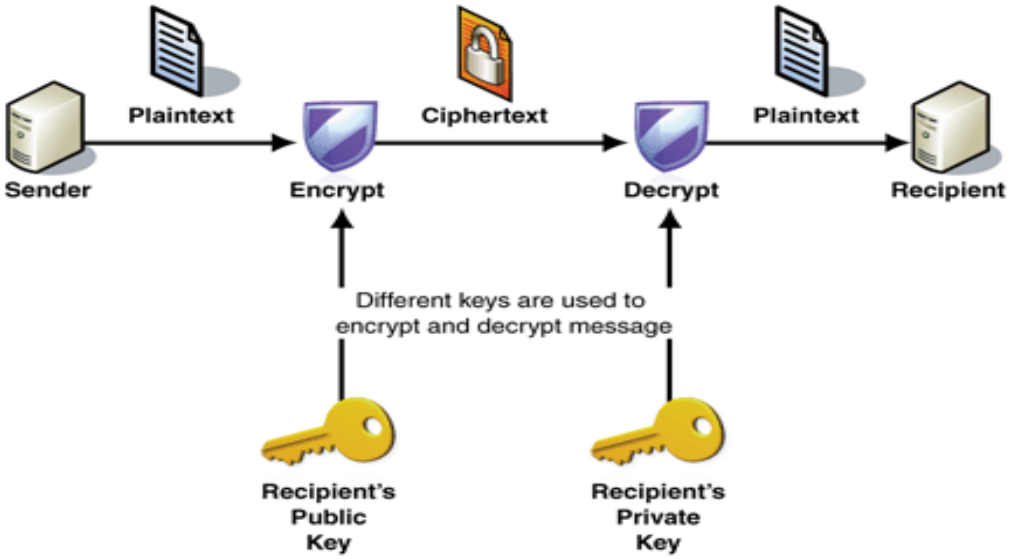
Example:

- DNA plaintext messages are appended with one or more secret keys
- Resulting appended DNA strands are hidden by mixing them within many other irrelevant DNA strands (e.g., randomly constructed DNA strands).

VI. Cryptography

Much of the theoretical work in cryptography concerns cryptographic primitives—algorithms with basic cryptographic properties—and their relationship to other cryptographic problems. More complicated cryptographic tools are then built from these basic primitives. These primitives provide fundamental properties, which are used to develop more complex tools called cryptosystems or cryptographic protocols, which

guarantee one or more high-level security properties.
Note however, that the distinction between cryptographic primitives and cryptosystems, is quite arbitrary; for example, the RSA algorithm is sometimes considered a cryptosystem, and sometimes a primitive. Typical examples of cryptographic primitives include pseudorandom functions, one-way functions, etc.



One or more cryptographic primitives are often used to develop a more complex algorithm, called a cryptographic system, or cryptosystem. Cryptosystems are designed to provide particular functionality (e.g. public key encryption) while guaranteeing certain security properties. Cryptosystems use the properties of the underlying cryptographic primitives to support the system's security properties. Of course, as the distinction between primitives and cryptosystems is somewhat arbitrary, a sophisticated cryptosystem can be derived from a combination of several more primitive cryptosystems.

VII. Key Generation

RSA involves a public key and a private key. The public key can be known by everyone and is used for encrypting messages. Messages encrypted with the public key can only be decrypted in a reasonable amount of time using the private key. The keys for the RSA algorithm are generated the following way:

1. Choose two distinct prime numbers p and q.
   - For security purposes, the integers p and q should be chosen at random, and should be of similar bit-length. Prime integers can be efficiently found using a primality test.
2. Compute n = pq.
   - n is used as the modulus for both the public and private keys. Its length, usually expressed in bits, is the key length.
3. Compute $\varphi(n) = \varphi(p)\varphi(q) = (p - 1)(q - 1) = n - (p + q - 1)$, where φ is Euler's totient function. This value is kept private.
4. Choose an integer e such that $1 < e < \varphi(n)$ and $\gcd(e, \varphi(n)) = 1$; i.e., e and φ(n) are coprime.
   - e is released as the public key exponent.
   - e having a short bit-length and small Hamming weight results in more efficient encryption – most commonly $2^{16} + 1 = 65,537$. However, much smaller values of e (such as 3) have been shown to be less secure in some settings.[5]
5. Determine d as $d \equiv e^{-1} \pmod{\varphi(n)}$; i.e., d is the modular multiplicative inverse of e (modulo φ(n)).

- This is more clearly stated as: solve for d given $d \cdot e \equiv 1 \pmod{\varphi(n)}$
- This is often computed using the extended Euclidean algorithm. Using the pseudocode in the Modular integers section, inputs a and n correspond to e and φ(n), respectively.
- d is kept as the private key exponent.

The public key consists of the modulus n and the public (or encryption) exponent e. The private key consists of the modulus n and the private (or decryption) exponent d, which must be kept secret. p, q, and φ(n) must also be kept secret because they can be used to calculate d.

- An alternative, used by PKCS#1, is to choose d matching $de \equiv 1 \pmod{\lambda}$ with $\lambda = \text{lcm}(p-1, q-1)$, where lcm is the least common multiple. Using λ instead of φ(n) allows more choices for d. λ can also be defined using the Carmichael function, λ(n).

Encryption
Alice transmits her public key (n, e) to Bob and keeps the private key d secret. Bob then wishes to send message M to Alice.
He first turns M into an integer m, such that $0 \leq m < n$ and gcd(m, n) = 1 by using an agreed-upon reversible protocol known as a padding scheme. He then computes the ciphertext c corresponding to

$$c \equiv m^e \pmod{n}$$

This can be done efficiently, even for 500-bit numbers, using Modular exponentiation. Bob then transmits c to Alice.
Note that at least nine values of m will yield a ciphertext c equal to m,[note 1]

Decryption
Alice can recover m from c by using her private key exponent d via computing

$$m \equiv c^d \pmod{n}$$

Given m, she can recover the original message M by reversing the padding scheme.
(In practice, there are more efficient methods of calculating c^d using the precomputed values below.)

A worked example
Here is an example of RSA encryption and decryption. The parameters used here are artificially small, but one can also use OpenSSL to generate and examine a real keypair.
1. Choose two distinct prime numbers, such as

$$p = 61 \text{ and } q = 53$$

2. Compute n = pq giving

$$n = 61 \times 53 = 3233$$

3. Compute the totient of the product as φ(n) = (p − 1)(q − 1) giving

$$\varphi(3233) = (61-1)(53-1) = 3120$$

4. Choose any number 1 < e < 3120 that is coprime to 3120. Choosing a prime number for e leaves us only to check that e is not a divisor of 3120.

Let $e = 17$

5. Compute d, the modular multiplicative inverse of e (mod φ(n)) yielding,

$$d = 2753$$

Worked example for the modular multiplicative inverse:

$$e \times d \bmod \varphi(n) = 1$$
$$17 \times 2753 \bmod 3120 = 1$$

The **public key** is (n = 3233, e = 17). For a padded plaintext message m, the encryption function is

$$c(m) = m^{17} \bmod 3233$$

The **private key** is (d = 2753). For an encrypted ciphertext c, the decryption function is

$$m(c) = c^{2753} \bmod 3233$$

For instance, in order to encrypt m = 65, we calculate

$$c = 65^{17} \bmod 3233 = 2790$$

To decrypt c = 2790, we calculate

$$m = 2790^{2753} \bmod 3233 = 65$$

Both of these calculations can be computed efficiently using the square-and-multiply algorithm for modular exponentiation.

In real-life situations the primes selected would be much larger; in our example it would be trivial to factor n, 3233 (obtained from the freely available public key) back to the primes p and q. Given e, also from the public key, we could then compute d and so acquire the private key.
Practical implementations use the Chinese remainder theorem to speed up the calculation using modulus of factors (mod pq using mod p and mod q).

The values d_p, d_q and q_{inv}, which are part of the private key are computed as follows:

$$d_p = d \mod (p-1) = 2753 \mod (61-1) = 53$$
$$d_q = d \mod (q-1) = 2753 \mod (53-1) = 49$$
$$q_{\text{inv}} = q^{-1} \mod p = 53^{-1} \mod 61 = 38$$
$$\Rightarrow (q_{\text{inv}} \times q) \mod p = 38 \times 53 \mod 61 = 1$$

Here is how d_p, d_q and q_{inv} are used for efficient decryption. (Encryption is efficient by choice of public exponent e)

$$m_1 = c^{d_p} \mod p = 2790^{53} \mod 61 = 4$$
$$m_2 = c^{d_q} \mod q = 2790^{49} \mod 53 = 12$$
$$h = (q_{\text{inv}} \times (m_1 - m_2)) \mod p = (38 \times -8) \mod 61 = 1$$
$$m = m_2 + h \times q = 12 + 1 \times 53 = 65$$

VIII. Challenges

- Intruders: those who capture the packet and alter the information
- Users with limited privileges should not be able to access unauthorized information
- Crypto analyst: those who decrypt cipher text into plain text without key.

IX. Objective And Methodology

- There are multiple enhancements in security mechanism.
- The presence of intruder should be detected to prevent an unauthorized access of information by adding some delimiter at the end of encrypted text and same delimiter should be used during decryption.
- Some time information to be sent are multiple and merged using delimiter into plain text then at the time of decryption plain text is split again in multiple pieces of information.
- Allow authentic access to the information on the basis of privilege levels of user.
- To protect information from cryptanalyst IP Filter would be attached in decryption module.

X. Conclusion

Presented an initial investigation of DNA-based methods for Cryptosystems.Main Results for DNA one-time-pads cryptosystems:

- Gave DNA substitution and XOR methods based on one-time-pads that are in principle unbreakable.
- Gave an implementation of our DNA cyptography methods including 2D input/output.

XI. Future Scope For Dna Steganography:

- a certain class of DNA steganography methods offer only limited security; can be broken with some reasonable assumptions on entropy of plaintext messages.
- modified DNA steganography systems may have improved security.

References

[1]. Logik Bomb: Hacker's Encyclopedia (1997)
[2]. Hafner, Katie; Markoff, John (1991). Cyberpunk: Outlaws and Hackers on the Computer Frontier. New York: Simon & Schuster. ISBN 0-671-68322-5.
[3]. Sterling, Bruce (1992). The Hacker Crackdown. Bantam. ISBN 0-553-08058-X.
[4]. Slatalla, Michelle; Joshua Quittner (1995). Masters of Deception: The Gang That Ruled Cyberspace. HarperCollins. ISBN 0-06-017030-1.
[5]. Dreyfus, Suelette (1997). Underground: Tales of Hacking, Madness and Obsession on the Electronic Frontier. Mandarin. ISBN 1-86330-595-5.
[6]. Verton, Dan (2002). The Hacker Diaries : Confessions of Teenage Hackers. McGraw-Hill Osborne Media. ISBN 0-07-222364-2.
[7]. Thomas, Douglas (2002). Hacker Culture. University of Minnesota Press. ISBN 0-8166-3345-2.
[8]. Taylor, Paul A. (1999). Hackers: Crime in the Digital Sublime. Routledge. ISBN 978-0-415-18072-6.
[9]. Introduction to Cryptography
[10]. http://en.wikipedia.org/wiki/Cryptography
[11]. Fundamentals of Cryptography: Algorithm and Security Services byProfessor Guevara Noubir
[12]. http://www.ccs.neu.edu/home/noubir/Courses/CSU610/S06/cryptography.pdf.