Fuzzy Inference Based Text Secured Graphical User Interface Using RSA Algorithm

Bhavyatta Bhardwaj
Department of Information Technology
Dronacharya College of Engineering
Greater Noida, India
mahamayamohanty@yahoo.co.in

Mahamaya Mohanty
Department of Information Technology
Dronacharya College of Engineering
Greater Noida, India
bhavyatta@gmail.com

Surbhi Agrawal
Department of information Technology
Dronacharya College of Engineering
Greater Noida, India
surbhi_agarwal1992@yahoo.com

***Abstract*—** This paper uses RSA algorithm that converts a plain text to cipher text according to its methodology that is to be sent through a graphical user interface and the receiver of this can see both the encrypted and decrypted message with the help of fuzzy inference system.

Keywords-component; Fuzzy Inference, IF-THEN rule, GUI, RSA Algorithm, Public Key, Private Key

I. INTRODUCTION

Fuzzy logic [1] is a form of multi-valued logic or probabilistic logic where it deals with reasoning that is approximate rather than fixed and exact. When we compare it to traditional binary sets (where variables may take on true or false values) fuzzy logic variables may have a truth value that ranges in degree between 0 and 1 having membership values. Fuzzy logic has been extended to handle the concept of partial truth, where the truth value may range between completely true and completely false. Furthermore, when linguistic variables are used, these degrees may be managed by some specific functions.

II. TYPES OF FUZZY INFERENCES

Fuzzy Inference [4][7]is referred to as approximate reasoning which refers to computational procedures used for evaluating linguistic descriptions. The two important inferring procedures are
(i) Generalized Modus Ponen (GMP)
(ii) Generalized Modus Tollen (GMT)

GMP is formally stated as

IF x is A THEN y is B,
x is A'
---
y is B'

GMT is formally stated as

IF x is A THEN y is B,
x is B'
---
x is A'

III. RSA CRYPTOGRAPHIC ALGORITHM

RSA is the most popular and proven asymmetric key cryptographic algorithm. The RSA algorithm involves three steps: key generation, encryption and decryption.

Key generation

RSA involves two types of keys; one is a public key and the other is a private key**.** The public key can be known to 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. First we choose any 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 $n = p\,q$.

   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 $\varphi(n) = (p - 1)(q - 1)$, where φ is Euler's totient function.

4. Choose an integer e such that $1 < e < \varphi(n)$ and greatest common divisor $\gcd(e, \varphi(n)) = 1$; i.e., e and $\varphi(n)$ 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) has been shown to be less secure in some settings.[4]

5. Determine d as $d \equiv e^{-1} \pmod{\varphi(n)}$, i.e., d is the multiplicative inverse of e (modulo $\varphi(n)$).

   - This is more clearly stated as solve for d given $de \equiv 1 \pmod{\varphi(n)}$
   - This is often computed using the extended Euclidean algorithm.
   - d is kept as the private key exponent.

By construction, $d \cdot e \equiv 1 \pmod{\varphi(n)}$.

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.

IV. PROPOSED APPROACH

The first and most intuitive approach implements all possible combinations of the given fuzzy sets as rules. This way of doing so shows some drawbacks, which are handled by additional methods. Due to insufficient workspace coverage, some rules may never be fired. However, a diffusion procedure can be used to initialize the unfired rules. The choice of the number of fuzzy sets in each dimension carries significant consequences: it can be dynamically chosen within the second approach. When the number of combinations increases, it is necessary to limit the number of rules: the third approach initializes one rule per data pair. Finally, the decision trees are introduced at the end of this section. They

p, q, and $\varphi(n)$ must also be kept secret because they can be used to calculate d.

- An alternative is to be used by Public Key Cryptography Standard, 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 $\varphi(n)$ allows more choices for d. λ can also be defined using the Carmichael function, $\lambda(n)$.

Encryption

Alice transmits her public key (n, e) to Bob and keeps the private key secret. Bob then wishes to send message M to Alice.

He first turns M into an integer m, such that $0 \leq m < n$ 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 quickly using the method of exponentiation by squaring. Bob then transmits c to Alice.

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.

generate incomplete rules but require a predetermined fuzzy partitioning. The approach proposes the use of RSA algorithm that converts a plain text to cipher text according to its methodology that is to be sent through a graphical user interface and the receiver of this can see both the encrypted and decrypted message with the help of fuzzy inference system. Both the generalized fuzzy inference system GMT and GMP are used.

GMP is formally stated as:

IF x is Plain Text, THEN y is Cipher Text,

x is Plain Text'

---

y is Cipher Text'

GMT is formally stated as

IF x is Plain Text, THEN y is Cipher Text,

x is Cipher Text'

---

x is Plain Text'

VI.CONCLUSION

Fuzzy inference systems (FIS) are widely used for process simulation or control. They can be designed either from expert knowledge or from data. For complex systems, FIS based on expert knowledge only may suffer from a loss of accuracy. This is the main incentive for using fuzzy rules inferred from data. Here the design is basically a FIS from data which can be decomposed into two main phases: automatic rule generation and system optimization for text messaging through GUI. Rule generation leads to a basic system with a given space partitioning and the corresponding set of rules. System optimization can be done at various levels. Variable selection can be an overall selection, or it can be managed rule by rule. Rule base optimization aims to select the most useful rules and to optimize rule conclusions.

V.RESULT ANALYSIS

Here the result in GMP is y and the result of GMT i.e. x is tested with the respective key that we have generated during key generation of RSA. This makes the test features secured enough so that the third unauthorized party cannot make any immediate changes nor can make the duplicate illegal use of the GUI

REFERENCES

[1] Kosko, Bart, "Fuzzy Thinking: The New Science of Fuzzy Logic", Warner, 1993 [For technical details, see Kosko,Bart, "Fuzzy cognitive maps", International Journal of Man-Machine Studies 24:65-75, 1986.]

[2] McNeill, Daniel, and Freiberger, Paul, "Fuzzy Logic: The Discovery of a Revolutionary Computer Technology",
Simon and Schuster, 1992. ISBN 0-671-73843-7. [Mostly history, but many examples of applications.]

[3] Brubaker, D.I., "Fuzzy-logic Basics: Intuitive Rules Replace Complex Math," EDN, June 18, 1992.

[4] D. Basin, M. Clavel, J. Doser, and M. Egea. Automated analysis of security-design models. Information and Software Technology, 51(5):815{831, 2009.

[5] D. Basin, J. Doser, and T. Lodderstedt. Model driven security: From UML models to access control infrastructures. ACM Transactions on Software Engineering and Methodology, 15(1):39{91, 2006.

[6] K. Blankenhorn and W. Walter.
Extending UML to GUI modeling. http://www.bitfolge.de/pubs/MC2004_Poster_Blankenhorn.pdf, 2004.

[7] I. Rojas, H. Pomares, J. Ortega, and A. Prieto, "Self-organized fuzzy system generation from training examples," *IEEE Trans. Fuzzy Syst.*,vol. 8, pp. 23–26, Feb. 2000.

[8] D. Leitch and P. Probert, "New techniques for ge netic development of fuzzy controllers," *IEEE Trans. Syst., Man, Cybern. C*, vol. 28, pp.112–123, Aug. 1998.

[9] A. Gonzales and R. Perez, "Slave: A genetic lea rning System Based on an Iterative Approach," *IEEE Trans. Fuzzy Syst.*, vol. 7, pp. 176–191,
Apr. 1999.

[10] C.-C.Wong and S.-M. Her, "A self-generating method for fuzzy systems design," *Fuzzy Sets Syst.*, vol. 103, pp. 13–25, 1999.

[11] F. Guély, R. La, and P. Siarry, "Fuzzy rule base learning through simulated annealing," *Fuzzy Sets Syst.*, vol. 105, pp. 353–363, 1999.

[12] M. Russo, "Fugenesys—A fuzzy genetic neural system for fuzzy modeling," *IEEE Trans. Fuzzy Syst.*, vol. 6, pp. 373–388, Aug. 1998.