International Journal of New Technology and Research (IJNTR)
ISSN:2454-4116, Volume-3, Issue-3, March 2017 Pages 87-89

Implementation of Enhanced Honey Encryption for IoT Security

Dr. Santhi Baskaran, S.V.L Sarat Chandra , P.Venkatesh , E.Silambarasan , M.Dinesh

Abstract— **Internet-of-Things (IoT) is becoming an emerging trend where most of the digital and physical things are connected and can be interacted with the help of Internet. Since the usage of IoT devices are increasing, there is a need for better security. In this paper, we propose a new cryptographic architecture for IoT devices which is based on honey encryption scheme.**

Index Terms— **Internet of Things, Security.**

I. INTRODUCTION

IoT is becoming an emerging trend where we can control all hardware devices with the help of software. Since IoT is carrying more sensitive information across the Internet, there is a need for better security. So by implementing cryptographic algorithms, the user can communicate with IOT devices in safe and secured manner [1]. The problem of data encryption prompted us to implement improvised honey encryption (HE). In computer security, honey commonly denotes a false resource
designed to lure or deceive an attacker. Honeypots, for example, are servers designed to attract attackers for observation and study. HE creates a cipher text that, when decrypted with an incorrect key or password, yields a valid-looking but bogus message, so that attackers can't tell when decryption has been
Successful [2]. By implementing our improvised architecture, we can achieve more security compared to simple honey encryption.

II. LITERATURE WORK

Even though many encryption algorithms exist, all are mostly based on password based encryption and vulnerable to many cyber-attacks. A recent study [3] reports that 1.08% of people chose the same password. Due to increase in computation speed of systems and distributed computing, the time takes to crack those encryption algorithm is decreasing. Since sensitive information is becoming vulnerable to attackers, there is a need for implementing advanced cryptographic algorithms. If we consider 64bit DES algorithm, by using brute force attack on a computer with dual Pentium processor running at 3.2GHz and 4x speed gain for every two years the following table show the time to decrypt [4].

TABLE I
TIME TAKEN TO DECRYPT THE 64-BIT DES ALGORITHM

Year	*Time*	*Time Unit*	*Year*	*Time*	*Time Unit*
2006	367	Centuries	2008	92	Centuries
2010	24	Centuries	2012	6	Centuries
2014	1.5	Centuries	2016	38	Years
2018	10	Years	2020	2.5	Years
2022	8	Months	2024	2	Months
2026	15	Days	2028	4	Days

From the above table we can understand that in future computers would take less time to crack the encrypted message. Hence there is a need for new encryption scheme that will provide barrier against brute-force attack. One of the solution to this problem is Honey encryption. Honey encryption constructs a cipher text that decrypts under any password to a plausible-looking message. It uses the property of distributed transforming encoder property for distribution of message seeds. By this honey encryption scheme, even though the attacker succeed in decrypting the cipher text he doesn't know which is correct one.

III. SYSTEM ARCHITECTURE

In our architecture, we use both symmetric and asymmetric key cryptosystem. First, the user interacts with the IoT device by sending a message which will be encrypted at user end. The IoT device which is attached to raspberry pi device will receive the encrypted message. Decryption is done at receiving side i.e on raspberry pi. The IoT device will respond according to the message which is decrypted.

.

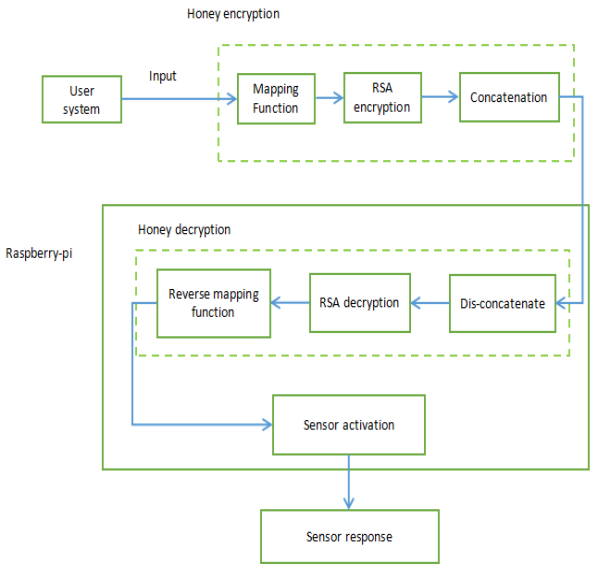


Fig. 1. Overall system architecture

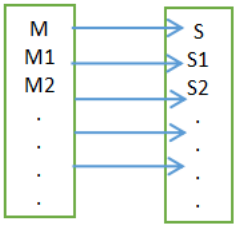
Dr. Santhi Baskaran, Department of Information Technology,Pondicherry Engineering College, Puducherry, India
S.V.L Sarat Chandra, Department of Information Technology,Pondicherry Engineering College, Puducherry, India
P.Venkatesh, Department of Information Technology,Pondicherry Engineering College, Puducherry, India
E.Silambarasan, Department of Information Technology,Pondicherry Engineering College, Puducherry, India
M.Dinesh, Department of Information Technology,Pondicherry Engineering College, Puducherry, India

IV. PROPOSED ENCRYPTION AND DECRYPTION PROCESS

In this we have used two key cryptosystem in which one key is symmetric key which will be known to both sender and receiver. For security reasons, the symmetric is generated randomly and should be alpha-numeric string. Another key is asymmetric key which uses public and private keys of sender and receiver where we have used RSA algorithm.

A. *Encryption process*

Let A and B be two communicating parties, where A wish to send message M to B. In process of encryption, the user will provide message (M) and symmetric key (k).The message is placed in message dictionary which will be mapped to the hash value of message (S) generated using SHA256 logic.The dictionary also contains some randomly selected valid strings (M1,M2,M3,..) which are mapped to the seed values (S1,S2,S3,…).



The key value is hashed using SHA256 with a salt value (R) which is randomly generated.This seed value is encrypted with
public key [5] of receiver B and concatenated with xor value of mapped value of message S and hash value of key and salt R i.e

$$C = H(K,R) \oplus S \,||\, RSA(Pub, R)$$

here 'C' is cipher text generated.The resultant cipher text is send to receiver B. The seed values S1, S2, S3,….. are obtained by xor of hash value of key K with salt R, S, and hash value of key K1 , k2, k3 …with salt R (here K1, K2, K3.. are bogus keys) i.e

$$S_i = H(K,R) \oplus S \oplus H(K_i, R)$$

The below figure represents the overall encryption process which is done at sender side.

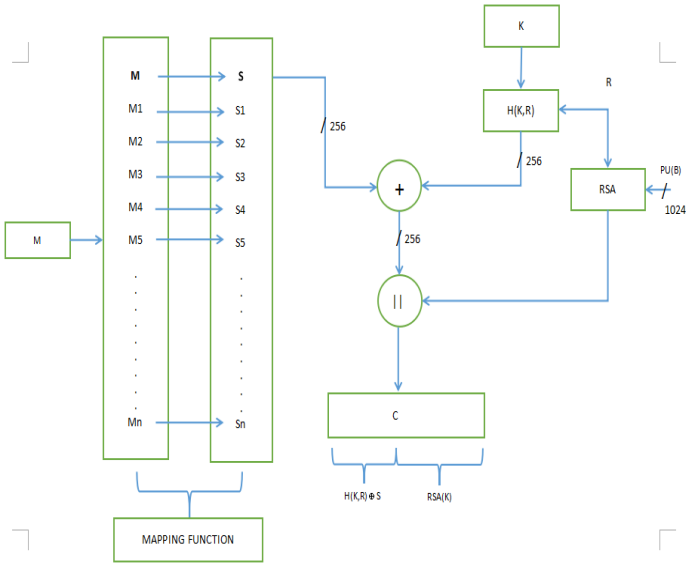
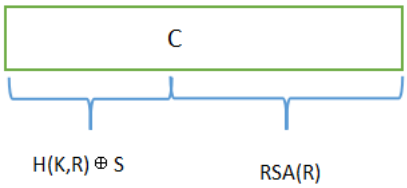


Fig. 2. Encryption Architecture

B. *Decryption process*

In the resultant cipher text which is generated by sender first 256 bits contains xor value of mapped value and hash value of key and the remaining content is RSA encrypted random string. First at receiving side, the RSA part is taken from cipher text and decrypt it with his public key to get the random string R.



Then the receiver will generate the hash value by using his symmetric key K and decrypts random string R. Finally the resultant value is xor with the first 256 bit of cipher text.The resultant will generate the value which will be reverse mapped to get resultant message i.e

$$H(K,R) \oplus S \oplus H(K, RSA(PR_b, RSA(Pub, R))) = S$$

The below figure represents the overall decryption process which is done at receiving side for decrypting the encrypted message.

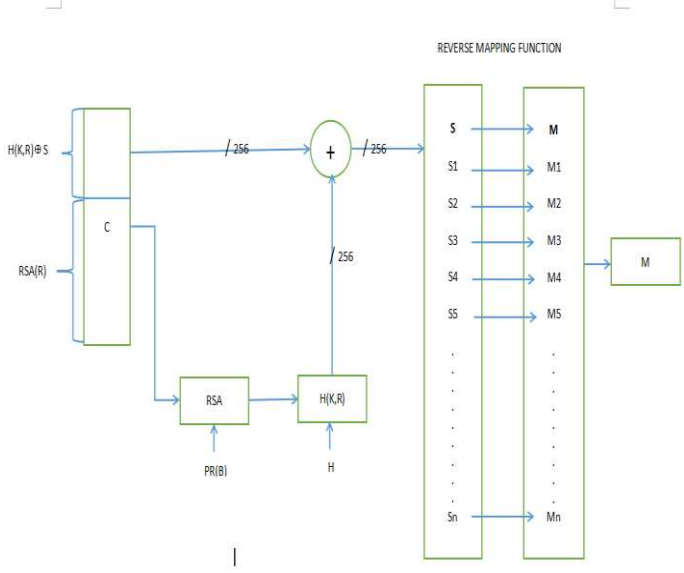


Fig. 2. Decryption architecture

Let us consider a scenario where an attacker knows the value of private key and decrypted the random string R. If the attacker
tries to decrypt the message using key which is in key list (k1, k2, k3,…) then the result is

$$H(K,R) \oplus S \oplus H(K1, RSA(PR_b, RSA(Pub, R))) = S1$$

Here the S1 maps to M1 which is bogus message.

C. *Sensor activation*

In this, the output can be verified by functioning of sensor. First, the user will provide message and password through which the message is encrypted. In process of decryption, the user has to provide key. When the key given by user is correct, then one sensor will be activated. If an attacker tries to decrypt the message using brute-force attack, and if he used a password which is in selected password list then it will return bogus message to attacker. After returning bogus message, another sensor will be activated indicating original user that he is being attacked. This is done by connecting sensors to the GIPO (General Purpose Input &Output) pins of the raspberry pi.

V. CONDITIONS

- The salt value (R) must contain minimum 10 characters which includes uppercase, lowercase and integer values.
- In calculation of hash value, total number of rounds should be at least 100000.
- Frequently used passwords such as ‘12345’, ‘password’ [6] etc are used for the values of keys K1, K2, K3,… so that it is easy to fool the attacker.
- If the sender uses the key K which is in key values K1, k2, K3,.. then the seed value obtained by value of K_i must be updated with the value obtained by K.

VI. RESULTS

Security for honey encryption is directly proportional to number of message seeds used i.e more the message seeds, more the security. Let us consider a scenario where the attacker applies brute force attack on honey encrypted message, there by obtaining message seed values used. After that, the attacker has to choose correct message from the obtained message seeds.
In our architecture, we used two keys in which one is used for RSA encryption. So when the attacker tries to decrypt the message which is encrypted by our encryption scheme, first he has to decrypt the RSA key and then he has to choose the correct message from message seed. Hence by implementing this architecture, with limited message seed space, we are getting more security.

VII. CONCLUSION

Security is one of the major challenges faced by IoT industry. In future most of the standard encryption algorithms that exist today can be broken with in seconds. Hence by implementing this type of encryption technique, both security and integrity is preserved. Since in our proposed architecture we calculate hash value for every message which takes more time to encrypt the message for IoT devices with low processing speed. In future improvements can be done in reducing the computation of message to seed conversion process.

REFERENCES

[1]. Mohammad Abdur Razzaque; Marija Milojevic-Jevric; Andrei Palade; Siobhán ClarkeI,EEE “Middleware for Internet of Things” , IEEE Internet of Things Journal Volume: 3, Issue: 1Pages: 70 - 95,Year: 2016.
[2]. A. Juels and T. Ristenpart, “Honey Encryption: Security beyond the Brute-Force Bound,” Advances in Crypto logy—Eurocrypt 2014, LNCS 8441, Springer, 2014, pp. 293–310;
[3]. Joseph Bonneau. The science of guessing: analyzing an anonymized corpus of 70 million passwords. In Security and Privacy (SP), 2012 IEEE Symposium on, pages 538–552. IEEE, 2012.
[4].https://jscott.net/crypto/64bitcrack.htm#128BIT.
[5].https://courses.csail.mit.edu/6.857/2015/files/tyagi-wang -wen-zuo.pdfs
[6]. http://www.telegraph.co.uk/technology/2017/01/16/ worlds-common-passwords-revealed-using/