Team information

Name: Dolunay Dagci, Jake Clouse

Identification number: 001395389, 001147536

Contact information: ldagci@albany.edu, jclouse@albany.edu

Title

Alternative N-bit Key Data Encryption for Block Ciphers

Description

This paper offers a different method for creating an n-bit, n-block, and key cipher that can be created from a key of reasonably short length. The suggested method is able to reduce post-encryption patterns, which can be utilized in symmetric or asymmetric cryptography to unlock the encryption key and compromise security.

Motivation

We wanted to tackle something that would be challenging but also have real world applications. As we have learned in our 526 class, just because a cipher seems to be secure doesn't mean that it is. This paper has explored post-encryption pattern (PEP) left behind by RSA and alerted us how this can be used to help unravel a cipher. It explores the idea of adding additional processes to obfuscate the residual data and make a rock-solid cipher.

Methodology

We will be using n-bits and n-blocks data encryption to overcome post-encryption patterns, which uses neither random generator to produce an initialization vector, as in classical cipher block chaining, nor the same key or large keys to encrypt each data block of the message resulting in storage difficulties. Our method primarily depends on the confidentiality of the message encryption key (ENCK), which is only known by the sender and the recipient, and on the assumed irreversibility of the selected HASH function, and the output of the decryption algorithm is the recovered MSN (message) that is restored by appending the decrypted blocks.

N-bits data encryption with a block cipher that needs no traditional initialization vector, doesn't recycle the same key to encrypt blocks, and is encrypted with its own encryption key. For the input, you will have to provide 3 things: You message to encrypt (*n* bytes in length), the number of blocks in your message (n), and the encryption key. The crux of this encryption scheme rests on two main points: 1) the secrecy of the encryption key and 2) the resilience of the HASH function to being reversed (SHA 256). We will have to XOR the receiver private key with the sender public key (or vice versa) using SECP256K1 (elliptic curve cryptography algorithm).

We need to grab the HASH value of the combination of the plaintext message size and the encryption key size (both in bits). We take this, store it in variable e, then take a HASH of the encryption key, and store it into variable H1. We then encrypt the plaintext message with the given encryption key (using XOR), then store it into block 0 (remember that the block must be the length of the encryption key in bits). We use e in conjunction with H1 to generate a new HASH, then save the output in H1. Now we can use H1 as a key to encrypt the rest of blocks of data (then overwriting H1 all the while). At the end of our encryption there are n encrypted blocks of data that we can then transfer to the receiver.

For our decryption, we need to know the encryption key (again, we are assuming that sender and receiver have found a safe, pre-existing way to transmit the key). We first need to take a HASH of the encryption key and store that into H1. Then we just have to reverse the process going block by block, XORing each with the encryption key and storing the new HASH in H1. This has the added benefit of ensuring the integrity of the restored plaintext by computing its HASH and comparing it with e.

Resources Used

An Explanation of Cipher Block Chaining: https://www.ibm.com/docs/en/linux-on-systems?topic=operation-cipher-block-chaining-cbc-mode

Guide to Guide to Elliptic Curve Cryptography: http://ir.juit.ac.in:8080/jspui/bitstream/123456789/5680/1/Guide%20to%20Elliptic%20Curve%20Cryptography.pdf

National Insitute of Standards and Technology Block Ciphers: https://csrc.nist.gov/projects/block-cipher-techniques

NIST Special Publication 800-135 Revision 1 Recommendation for Existing Application-Specific Key Derivation Functions: https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-135r1.pdf

HMAC: Keyed-Hashing for Message Authentication: https://www.rfc-editor.org/rfc/pdfrfc/rfc2104.txt.pdf

Schedule

Task	Deadline Date
Project Proposal	02/24
Create the project codebase, create issues and distribute tasks	02/28
Weekly Meeting	02/27
Weekly Meeting	03/06
Project progress 20%	03/10
Weekly Meeting	03/13
Weekly Meeting	03/20
Project progress 40%	03/24
Weekly Meeting	03/27
Project interim check	03/30
Weekly Meeting	04/03

Project progress 60%	04/07
Weekly Meeting	04/10
Weekly Meeting	04/17
Project progress 80%	04/21
Weekly Meeting	04/24
Weekly Meeting	05/01
Project completed 100%	05/03
Final	05/04
Project Demo	05/05

References

“Alternative N-bit Key Data Encryption for Block Ciphers” by Kayque M. C. Damasceno, Carlos A. de Moraes Cruz , Anderson V. C. de Oliveira , Luís S. O. de Castro from Federal University of Amazonas (Brazil).

Common Cryptographic Architecture (CCA): Cipher Block Chaining (CBC) mode. (n.d.). Common Cryptographic Architecture (CCA): Cipher Block Chaining (CBC) Mode. https://www.ibm.com/docs/en/linux-on-systems?topic=operation-cipher-block-chaining-cbc-mode

Block Cipher Techniques | CSRC. (2017, January 4). Block Cipher Techniques | CSRC. https://csrc.nist.gov/projects/block-cipher-techniques

Krawczyk, H., Bellare, M., & Canetti, R. (n.d.). *RFC 2104: HMAC: Keyed-Hashing for Message Authentication*. RFC 2104: HMAC: Keyed-Hashing for Message Authentication. https://doi.org/10.17487/RFC2104

Hankerson, Menezes, & Vanstone. (2004). *Guide to Elliptic Curve Cryptography* [PDF]. Springer-Verlang. http://ir.juit.ac.in:8080/jspui/bitstream/123456789/5680/1/Guide%20to%20Elliptic%20Curve%20Cryptography.pdf

Dang, Q. (2011, December 1). *SP 800-135 Rev. 1, Recommendation for Existing Application-Specific KDFs | CSRC*. SP 800-135 Rev. 1, Recommendation for Existing Application-Specific KDFs | CSRC. https://doi.org/10.6028/NIST.SP.800-135r1