Volume 3, Issue 7, July 2013 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

A Novel Approach for Modified Linear All-or-Nothing Transform for Preventing Fussy Blocking Attacks

U.Santhi Swaroopa*
M.Tech, Dept. of CSE,
ASIT, Gudur
India

P.Venkateswara Rao
Professor & HOD, Dept. of CSE
ASIT, Gudur
India

A. Rama Mohan Reddy
Professor, Dept. of CSE
S.V.U College of Engineering, Sri Venkateswara University, Tirupati,
India

Abstract — Wireless networks are made upon a shared medium makes it exposed to intentional interference attacks, known as blocking. Due to this intentional interference, a jammer can intensely reduce the network performance by launching a Denial-of-Service (DoS) attack. In this work, we identify the problem of fussy blocking attacks with internal information in wireless networks. In these attacks, the jammer is active only for small portion of time, fussy aiming messages of great significance. We show that fussy blocking attacks can be launched by behaving real-time packet categorization at the physical layer. To soothe these attacks, existing linear all-or-nothing transform before encryption process having a drawback of its linearity and non-randomizing character even if all particular input blocks are protected, a jammer can still obtain information about linear dependencies between several input blocks. To reduce the problem, we address a modified linear all-or-nothing transform after applying CBC mode of data encryption which converts the plaintext into cipher-text before converting into pseudo-messages. By masking the last block of pseudo-message makes the cipher-text more stronger. This construction is significantly more efficient and ensures that a jammer does not even know the exact cipher-text.

Keywords— Denial-of-Service, Fussy Blocking, Packet Categorization, Modified Linear AONT, CBC Transform

I. Introduction

Wireless technologies have become increasingly popular in our everyday business and personal lives. It enables atleast one device to communicate without materialistic connections -without requiring network or peripheral cabling. As we know that wireless networks deal as the transport device between devices and surrounded by devices. However, because of this wireless nature these are prone to multiple security threats in which one of the major serious security threat is blocking. Blocking can disrupt wireless transmission and can occur either unintentionally in the form of noise or interference at the receiver side. Blocking attacks may be viewed as a special case of Denial-of-Service (DOS) attacks [2]. In this paper, we identify the problem of fussy blocking with internal information. We believe a sophisticated jammer who is conscious of network mysteries and the details about performance of the network protocols at each and every layer in the network stack. The jammer exploits his internal knowledge for launching fussy blocking attacks in which specific messages of "great significance" are aimed. For illustration, a jammer can aim route-request/route-response messages at the routing layer to avoid route invention, or aim TCP acks in a TCP period to severely corrupt the output of an end-to-end flow. To commence fussy blocking attacks, the jammer must be able of implementing the "categorize-then-block" strategy before the finalization of a wireless transmission. Such strategy can be accomplished either by categorizing transmitted packets using protocol semantics, or by deciphering packets on the air. In the recent process, the jammer may decipher the first few symbols of a packet for regaining useful packet identifiers such as packet nature, sender and receiver address. After categorization, the jammer must induce an adequate number of bit errors, as a result that the packet cannot be regained at the receiver. Fussy blocking requires a concealed knowledge of the physical layer, as well as of the details of upper layers.

We trace out the feasibility of real-time packet categorization for launching fussy blocking attacks with the internal information. To soothe these attacks, [11][12] existing linear all-or-nothing transformations encryption process having a drawback of its linearity and non-randomizing character even if all particular input blocks are protected, a jammer can still obtain information about linear dependencies between several input blocks. To reduce the problem, we identify a new mode of using linear all-or-nothing transform in conjunct with CBC mode of data encryption. At the sender, we suggest applying an AONT after encryption. An AONT deals as a publicly known and entirely invertible post-processing step to a cipher-text after it is passed from a cipher block chaining encryption algorithm makes the plaintext into pseudo-message. Then the last block of pseudo-message is masked by a pseudorandom constant to make the cipher-text extra stronger and the inverse process is applied at the receiver in order to get the original plaintext. This construction is significantly more efficient and ensures that a jammer does not even know the exact cipher-text that has to be attacked.

II. Literature Survey

Blocking problem has been addressed under various threat models. The impact of external fussy jammer aiming various control packets at the MAC layer is studied in the paper [9] by Thuente. Fussy blocking attack is based on protocol semantics, where they considered numerous packet identifiers for encrypted packets such as packet range, signal sensing and timing information of different protocols. Fussy blocking attacks have been experimentally implemented using software defined radio engines [10]. USRP2-based blocking platform called RFReact was implemented by Wilhelm [10] that enables fussy and reactive jamming. [11] They developed three schemes that prevent jamming attacks; they are Strong Hiding Commitment Scheme, Hiding Scheme based on Cryptographic Puzzle and Hiding Scheme based on All or Nothing Transformation.

III. Problem Statement

Consider the Fig.1. Nodes S and R communicate via a wireless connection. In the communication range of both S and R there is a jammer node J. When S broadcasts a packet m to R, node J categorizes m by receiving only the first few symbols of m. Then J corrupts m away from regain by interfering with its reception at R. We identify the problem of avoiding the jamming node from categorizing m in real time, thus reducing J's ability to perform fussy blocking.

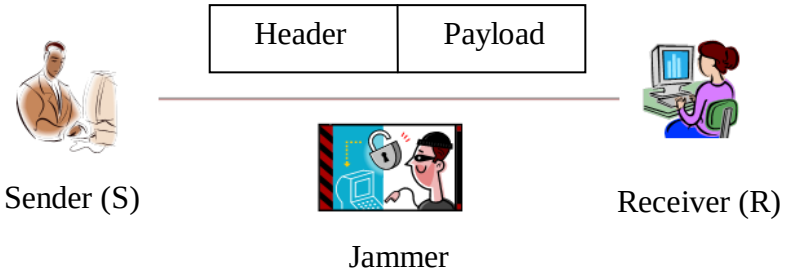

Fig.1. Awareness of Fussy Blocking Attacks

IV. Real-Time Packet Categorization

In this section, we describe how the jammer can categorize packets in real time, before the packet communication is finished. Once a packet is categorized, the jammer may choose to block it depending on his strategy. Consider the system architecture diagram represented in Fig.2. By the physical layer, a packet m is enciphered, interspersed, and inflected before it is transmitted over the wireless route. At the receiver, the signal is de-inflected, de-interspersed, and deciphered, to regain the original packet m.

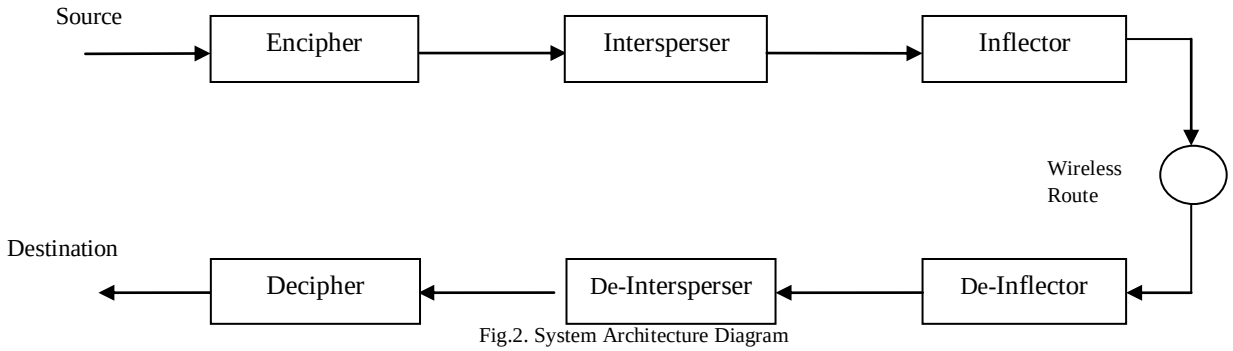

Fig.2. System Architecture Diagram

V. Hiding Scheme Based on Modified Linear Aont

In this section, we intend a solution based on Stinson-like Linear All-Or-Nothing Transformations (L-AONT) that proposes a modest computational nature. We suggest applying an AONT after encryption [7]. An AONT deals as a publicly known and entirely invertible post-processing step to a cipher-text after it is passed from a cipher block chaining encryption algorithm. A transformation ϕ, mapping message $y=(y_0, y_1, .., y_s)$ to a sequence of pseudo-message $z=(z_0, z_1, .., z_s)$, is an AONT if (a) ϕ is a bijection, (b) it is computationally impossible to know the exact plaintext, if one of the modified pseudo-message is unknown, and (c) ϕ and its inverse ϕ^{-1} are efficiently computable.

The advantage of this approach is the fact that the input of AONT is strongly randomized and thus, AONT itself does not necessarily need to be randomizing transform. As a consequence, we can use simpler AONT constructions which will significantly improve the processing speed.

In the first step the message $x = (x_1, x_2, .., x_s)$ is encrypted in the regular CBC mode with a randomly chosen initialization vector IV, in the second step the intermediate result, $y = (y_0, y_1, .., y_s)$ is transformed by a stinson-like linear AONT into $z = (z_0, z_1, .., z_s)$ and in the third step the last block z_s is masked by a pseudorandom constant k' which has been derived from the secret key k.

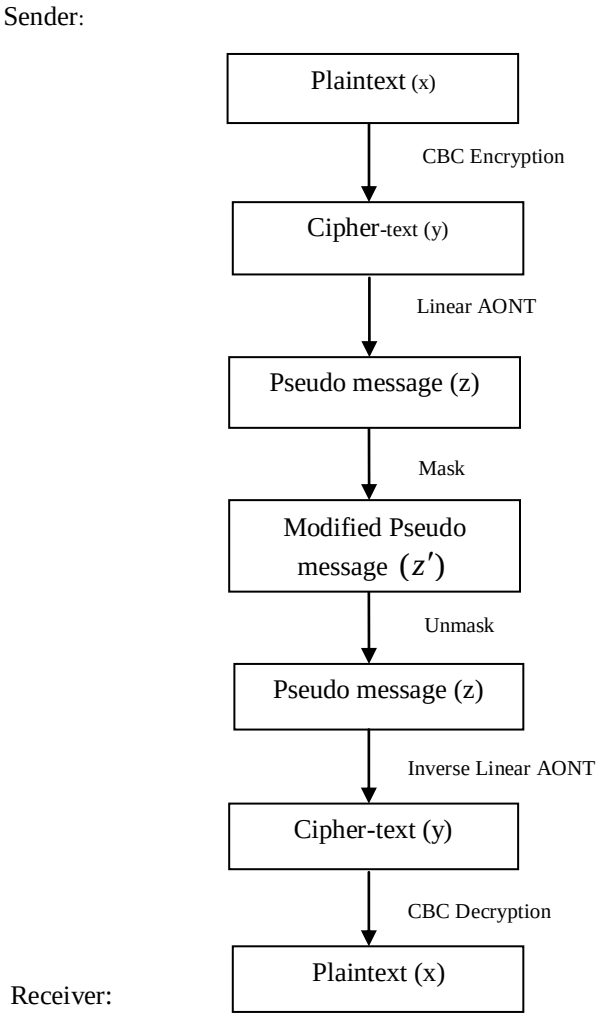

Fig.3. Linear AONT based Hiding Scheme

At the Sender

Step 1: Plaintext – Ciphertext

The purpose of the first step is encrypting and randomizing the data. The intermediate encryption output $y = (y_0, y_1, .., y_s)$ is a cryptographically strong pseudorandom sequence i.e., it is computationally infeasible to distinguish y from a sequence of uniformly distributed random bits.

In CBC mode, each block of plain-text is XORed with the previous cipher-text block before being encrypted with a chosen secret key. This way, each cipher-text block depends on all plain-text blocks processed up to that end. To create each message distinctive, an initialization vector (IV should be a random number) must be used in the first block.

Algorithm 1 CBC Encryption

1. Input : $x = (x_1, x_2, .., x_s)$
2. for $i \leftarrow$ 1 to s

3. $y_{i-1} = IV$
   Return IV
4. Choose key k
   $y_i \leftarrow E(k \oplus (x_i \oplus y_{i-1}))$
   Return y_i
5. Close for loop
6. Output : $y = (y_0, y_1, .., y_s)$

Step 2: Ciphertext – Pseudomessage

In the second step, we use a stinson-like linear AONT. Stinson [12] proposed the following AONT construction:

Let $y = (y_0, y_1, .., y_s)$ be a message consisting of blocks $y_i \in F_q$, where F_q is a finite field of order $q > 2$. AONT transform $\phi(y)$ is defined as $y.M^{-1}$ where M is an invertible $S \times S$ matrix with entries from F_q such that no entry is equal to zero [6].

$$M^{-1} = \begin{bmatrix} 110....0 \\ 011....0 \\ : \\ 000....1 \\ \lambda 00....1 \end{bmatrix}$$

The constant q must be a prime power and $\lambda \in F_q$ must be different from +1 and -1 to make the generating matrix invertible. The operations + and . denote the addition and multiplication in F_q.

Given $y = (y_0, y_1, .., y_s)$, one can compute $z = \Phi(y)$ as follows:

Algorithm 2 Linear AONT

1. Input : $y = (y_0, y_1, .., y_s)$
2. Choose λ
   $z_0 \leftarrow y_0 + \lambda . y_s$
   Return z_0
3. for $i \leftarrow$ 1 to s
4. $z_i \leftarrow y_i + y_{i-1}$
   Return z_i
5. Close for loop
6. Output : $z = (z_0, z_1, .., z_s)$

Step 3: Pseudomessage – Modified Pseudomessage

The function $f_w : \{0,1\}^n \rightarrow \{0,1\}^n$, used for computing k' in the third step of our construction, is a so-called w-slow one-way function which fulfils the following two procedures:

1. given any $x \in \{0,1\}^n$ the computation of $y = f_w(x)$ is at-least w-times slower than one execution of the encryption function e_k.

2. given any $y \in \{0,1\}^n$ it is computationally infeasible to compute $x = f_w^{-1(}(y)$.

We propose f_w as follows: $f_w(k) = k_{w+1}$ where k_i is computed according to the following recurrence: $k_0 = 0$, $k_1 = k$, and $k_i = e_{k_{i-1}}(k_{i-1} \oplus k_{i-2})$ for $i > 1$.

The last block of Pseudo message is masked by a pseudorandom constant k' which has been derived from the secret key k.

Algorithm 3 Mask

1. Input: $z = (z_0, z_1, .., z_s)$
2. $z_{s'} \leftarrow z_s \oplus k'$
3. Output : $z = (z_0, z_1, .., z_{s'})$

At the Receiver

Step 4: Modified Pseudomessage – Pseudomessage

The last block of Pseudo message is unmasked by a pseudorandom constant k' which has been derived from the secret key k.

Algorithm 4 Unmask

1. Input : $z = (z_0, z_1, .., z_{s'})$
2. $z_s \leftarrow z_{s'} \oplus k'$
3. Output : $z = (z_0, z_1, .., z_s)$

Step 5: Pseudomessage – Ciphertext

The inverse transformation $y = \phi^{-1}(z)$ can be efficiently computed as follows

Algorithm 5 Inverse Linear AONT

1. Input : $z = (z_0, z_1, .., z_s)$
2. if s is odd

   $$y_0 \leftarrow \frac{1}{1-\lambda}(z_1 - \lambda.\sum_{i=2}^{s}(-1)^i z_i),$$

   Return y_0
3. else

   $$y_0 \leftarrow \frac{1}{1+\lambda}(z_1 + \lambda.\sum_{i=2}^{s}(-1)^i z_i),$$

   Return y_0
4. for $i \leftarrow 1$ to s
5. $y_i \leftarrow z_i - y_{i-1}$

   Return y_i
6. Close for loop
7. Output : $y = (y_0, y_1, .., y_s)$

Note that if q is a power of two, there is no distinction between the operations + and -, both can be implemented by a binary XOR. Consequently, one does not need to distinguish between even and odd s during the inverse transform, and the factors $(-1)^i$ can be disregarded as well. Hence, the first step of the inverse transform will be simplified to $y_0 = \frac{1}{1 \oplus \lambda}(z_1 \oplus \lambda . \sum_{i=2}^{s} z_i)$. Also the restrictions regarding the value λ will be reduced to $\lambda \neq 1$.

Step 6: Ciphertext – Plaintext

In CBC mode, each block of cipher-text is XORed with the previous cipher-text block before being decrypted with the same secret key. This way, each plaintext block depends on all cipher-text blocks processed up to that end. To create each message distinctive, the same initialization vector must be used in the first block.

Algorithm 6 CBC Decryption

1. Input: $y = (y_0, y_1, .., y_s)$
2. for $i \leftarrow 1$ to s
3. $y_{i-1} = IV$
4. $x_i \leftarrow D(k \oplus y_i) \oplus y_{i-1}$

   return x_i
5. close for loop
6. Output: $x = (x_1, x_2, .., x_s)$

VI. Conclusion

We identified the problem of fussy blocking attacks with internal information in which the jammer is under wireless network, thus being conscious of the protocol specifications and shared network mysteries. We showed that the transmitted packets can categorize by jammer in real-time by deciphering the first few symbols of a partial transmission. To soothe these attacks, existing linear all-or-nothing transform before encryption process having a drawback. So, we proposed a modified linear all-or-nothing transform after applying CBC mode of data encryption. The advantage of this approach is the input of AONT is strongly randomized so that AONT itself does not necessarily need to be transformed randomly. In this it converts the plaintext into cipher-text before converting into pseudo-message and by masking the last block of pseudo-message which obtains the cipher-text more stronger. This construction is significantly more efficient and ensures that a jammer does not even know the exact cipher-text.

References

[1] W. Xu, W. Trappe, Y. Zhang, and T. Wood, "The feasibility of launching and detecting jamming attacks in wireless networks," in *Proceedings of the ACM MOBIHOC05*, 2005.

[2] Wood and J. Stankovic. *Denial-of-service in sensor networks*. IEEE Computer, 35(10):54-62, October 2002.

[3] IEEE Computer Society. IEEE Standard 802.15.4-2006: *Wireless medium access control and physical layer (PHY) specifications for low-rate wireless personal area networks WPANs)*. http://www.ieee802.org/11/, Sept. 2006.

[4] A. Proa˜no and L. Lazos. *Selective jamming attacks in wireless networks*. In Proc. of IEEE ICC, pages 1–6, May 2010.

[5] J. Menezes, P. C. van Oorschot, and S. A. Vanstone. Handbook of Applied Cryptography. CRC, 1996.

[6] C. J. Colbourn and J. H. Dinitz, Eds. The CRC Handbook of Combinatorial Designs, CRC Press, 1996.

[7] Eli Biham. Cryptanalysis of multiple modes of operation. 1995. Pre-Proceedings of ASIACRYPT '94. Submitted to J. Cryptology.

[8] A. Desai, *The security of All-or-nothing encryption: protecting against exhaustive key search*, Crypto '00, Springer Verlag, 2000.

[9] D.Thuente and M.Acharya. *Intelligent jamming in wireless networks with applications to 802.11b and other networks*. In proceedings of the IEEE Military Communications Conference MILCOM, 2006.

[10] M. Wilhelm, I.Martinovic, J.Schmitt and V.Lenders. *Reactive jamming in wireless networks: How realistic is the threat?* In proceedings of WiSec, 2011.

[11] Alejandro Proa˜no and Loukas Lazos. *Packet-Hiding Methods for Preventing Selective Jamming Attacks*. IEEE TRANSACTIONS ON DEPENDABLE AND SECURE COMPUTING, VOL. 9, NO. 1, JAN-FEB 2012

[12] D. Stinson. Something about all or nothing (transforms). Designs, Codes and Cryptography, 22(2):133–138, 2001.

