IJREAT International Journal of Research in Engineering & Advanced Technology, Volume 1, Issue 5, Oct-Nov, 2013
ISSN: 2320 - 8791
www.ijreat.org

An Opponent Resilient Secret Sharing Based on Advanced Decoding Mechanism

Sona G[1], Annapandi P[2] and Yamini B[3]

[1]**Information Technology, Dr.Sivanthi Aditanar College of Engineering, Tuticorin, Tamilnadu 628215, India**

[2]**Information Technology, Dr.Sivanthi Aditanar College of Engineering, Tuticorin, Tamilnadu 628215, India**

[3]**Information Technology, Dr.Sivanthi Aditanar College of Engineering, Tuticorin, Tamilnadu 628215, India**

Abstract

Earlier wireless Spread Spectrum communication takes place by setting up preconfigured keys among the communicating nodes that are constrained to possess synchronous behavior. This extends to several issues creating circular dependency problem, leading to insecure short-lived communication. In this paper, an opponent resilient secret sharing concept is introduced without any establishment of pre-shared keys by IFEB (Intractable Forward and Efficient Backward) decoding. It illustrates using TREKS at receiver side that enables secured transmission over wireless communication even when the node remains inactive and attaining jammer not to obtain the original data sent by the sender node. Main goal is to transmit the message in such a way that the time required to deliver the secret must be less than the time for the opponent to find key during transmission. Further, it come up with minimal storage overhead, cost effective and sustains long-lived secured communication among the interacting nodes.

Keywords: *Direct Sequence Spread Spectrum, zero preshared secret, anti-jamming, Message Extraction and Key Scheduling.*

1. Introduction

Wireless communication is often prone to jamming attack. Occurrence of these attacks results attacker to make delay, corrupting data and have their own control over the entire communication channel. The ability of the channel to recover back to its original state is necessary in wireless environment as this kind of communication becomes increasingly common for supervising physical infrastructure. Pre-configuration becomes possible when there exists small number of nodes. But this becomes impossible when various fabricators inscribe and depart the networks dynamically. Also this will create an issue causing dependency when keys are dispersed over air. The obstacle thus created is regarded as Circular Dependency Problem (CDP).

In this paper, an opponent resilient secret sharing mechanism is handled to overcome the CDP by using TREKS scheme. The scheme utilizes two main paradigms (*intractable forward decoding, efficient backward decoding).* Dimensions that make unique of this new approach than the previous versions include:

- Making jammer inefficient in terms of energy usage.
- Undetectable communication until the end of transmission.
- Allowing jammer channel oblivious.
- Synchronization not required.
- Efficient broadcast communication by the use of UDSSS (Uncoordinated Direct Sequence Spread Spectrum).
- No need of pre-established keys.

Intractable forward decoding is a powerful resilient method for sharing the secrets. Efficient backward decoding is based on message detection, block processing and key scheduling, making it fit for long-lived communication.

2. System Model

Our model conceives about performing SS communication in mobile ad hoc networks. Assumption made in regard with this model is nodes that are communicating share a medium with adversary. The following sections describe the model and common assumptions that are considered in this paper.

2.1 Opponent Model

The goal of sender is to establish an adversary resilient and energy efficient communication whereas the goal of the opponent is to prevent the receiver from decoding its messages and prevent successful reception of message. However, a jammer may simply increase the delay of the message extraction process or cause denial of service (DoS) attack on the receiver side. So, its secondary focus is to increase the computation and energy cost of the receiver by minimizing its jamming cost. The performance of the jammer is illustrated by determining the packet loss rate (PLR). Also evaluation is made by delay caused by its attacks during decoding process.

Types of Attack: Opponent we consider may

1. Create traffic by sending high power pulse in ongoing communication.
2. Cause delay in extracting the message.
3. Target to modify few bits in the message content.

In the following section we illustrate by protocol specific opponent strategies with expected result obtained to use it for real time application.

3. Forward Decoding and Scheduling Keys

We present the general idea for no pre configured key utilization and scheduling key for effective backward decoding and thereby enabling our method to possess optimal energy and storage cost with minimal overhead.

3.1 No pre-shared key approach

Assuming S as sender, R as receiver and J as jammer each of them shares the same communication channel. M is the message that is to be shared from sender to receiver and l is the length of message in bits. Random keys are generated by utilizing AES encryption algorithm which has been utilized ever today in military application as it is offering more security.

- The key K is not known to anyone except S.
- The length l and key length k are public information.
- When the bits are not equally probable, they can be compressed.
- The use of AES algorithm is also regarded as public information.

3.2 Intractable Forward Decoding

We first introduce some important terminologies to illustrate our concept:

- $d \epsilon$ {-1, +1}: Data sent by S.
- $đ \epsilon$ {-1, +1}: Estimated data on receiver side.
- *n:* Factor by which data is spread.
- $pn_{i\epsilon\{1,..,n\}}$ ϵ {-1, +1}: ith chip designed cryptographically not known to the opponent.
- $r_{i\epsilon\{1,..,n\}}$ ϵ {-1, +1} : ith chip transmitted by opponent.
- E_b: Energy transmitted for each bit.
- $u_i = d\sqrt{E}_b/n(pn_i)$: Signal transmitted by sender.
- $I_{i\epsilon\{1,..,n\}}$: Signal indexed at chip level.
- $v_i =$
  $$\int_{\sqrt{\frac{E_b n}{J}}}^{\infty} \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}} dx$$
  Received signal indexed at chip.
- *BER* (E_b, J, m): receiver side Bit Error Rate.

The Bit Error Rate of the dispersed signal is given by

BER $(E_b, \mathrm{J}, m) =$ (1)

Equation (1) shows that when the spreading factor is increased by n, the opponent needs to scale up the energy by the factor maintained by BER.

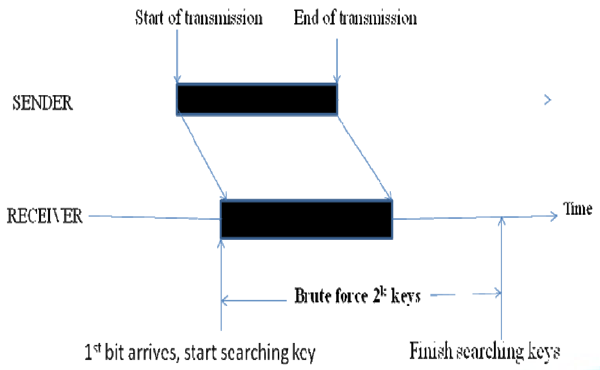

Fig.1 Message delivered before key is found by adversary

Figure 1 illustrates that the time taken by the jammer to find the key must always to be greater than the time required to transmit the message to ultimate receiver.

3.3 Key Scheduling

A sequence of keys that is $K_{1,}$ $K_{2,..,}$ K_n is known as schedule by setting i-1 MSB to some i-1 arbitrary value C. To spread keys we partition message into *k* segments that are derived cryptographically from $K_{i.}$
The symbols used to illustrate key scheduling process are described in Table 1.

Table 1: Summary of the symbols used

Notation	Definition
PN (.)	PN generating function
K_i	ith key in the schedule
K[m,.., n]	Kth substring from m to n bit
M[m,.., n]	Mth substring from m to n bit

The key scheduling algorithm is described in Figure 2.

```
routine TRANSMITTER (M, K)
        N1 ← M
        for i= 1,..,k do
                $K_i$[i,..,k] ← K[i,..,k]
                $K_i$[i,..,i-1] ← C[i,..,i-1]
                        $M_i$ ← $N_i$ [1,..,| $N_i$|/2]
                $PN_i$ ← PN ($K_i$)
                Disperse $M_i$ with $PN_i$
                $N_{i+1}$ ← $N_i$[|$M_i$ |+1,..,| $N_i$ |]
        end for
end routine
```

3.4 Security against Faster Key Searches

Routine that is devised above (figure 2) enable us to protect against brute-force jammers by choosing alternate message partition. We select an entropy *k* for each bit *b* such that the key search time is always greater than the transmission time so that the jammer would not be able to extract the message that is sent from the sender to receiver. We consider, the key scheduling approach would be adequate enough for providing security by means of cryptographically generating pseudo-random number of keys by utilizing AES-128 because of their best known cryptanalysis are close to brute force.

While with intractable forward decoding, it increases the computation complexity of the jammer from O (2*k*) to O (2^k) by offering higher security. This would ultimately result the jammer not to find the key soon.

4. Backward Decoding Mechanism

In Backward decoding approach, recipient can compute the key to find the end of message transmission. The computation cost complexity is reduced in receiver from O (2^k) to O (2*k*) so that transmission can take place earlier by keeping jamming resilient.

The procedure here follows computing two important levels. Level-1 involves finding correlation between the received signal and PN generated receivers MAC address. Level-2 infers the key where high correlation exists as detected from previous level. Figure 3 illustrates the two level processes that take place in backward decoding mechanism.

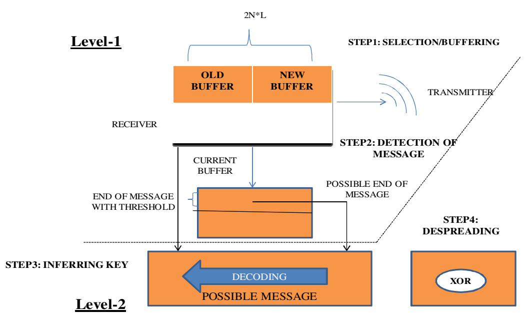

Fig. 3 Process of backward decoding.

4.1 Level-1 (Finding where the message ends)

Figure 3 depicts selection, buffering and end of message detection as part in level-1 process. Arrivals of incoming signals are queued in first-in-first-out (FIFO) order. To find the end of message, we have to compute the correlation between the signal and PN sequence in Spread Sequence systems. The process continues its iteration by checking for the values that are exceeding a given threshold. The values exceeding threshold results in providing answer in terms of false positive that the messages are obtained in wrong order but the entire messages are obtained without any loss.

4.1 Level-2 (Extraction of Message)

Figure 3 also shows the level-2 processing of backward decoding. For each possible end of message detection, we need to infer bitwise key. Here we deduce two possibilities for current key bit. Following figure 4 depicts the algorithm for extracting the message at receiver side.

```
routine KEY(Buf, PeakEoM[])
        for all j ϵ PeakEoM[] do
                peakPos ← n+j
                endIndex ← peakPos - 1
        for all p ϵ {1,..,k} do
                startIndex ← endIndex - |M| +1
                CountSuccess ← 0
                for all key c ϵ $K_{k-p}$ do
                success←PEAKDETECT(c,Buf,startIndex,endIn
                dex)
                   CountSuccess  ← CountSuccess + Success
        end for
        if CountSuccess =1 then
                $K_p$ ← c
        else
                abort
        end if
                endIndex ← startIndex
        end for
        m ← Despread(Buf[j – (nl) + 1,…,j], {$K_i$ })
        queue m into E[]
        end for
end routine
```

Fig. 4 Algorithm for extracting message.

This work enables to establish long term spread spectrum communication without any pre-configured key concept. With this accomplishment, it is possible for the sender and receiver to infer the spreading key in regular order.

5. Related Work

We present the general idea for no pre configured use of key through various works illustrated preciously in [1], [2], [3]. The survey mainly focuses on two key concepts: one is strategy to demolish the jamming attack and another criterion is establishment of key in different manner. [1] Illustrates by mainly focusing on reducing the effect of jammer attack to wider extent. In this paper, a control channel communication is handled by randomly distributing the key. With this random distribution of key it helps the communication channel to hide its location for certain time limit. Within that time interval, transmission of message takes place. This implies that it lessens the effect of jammer. The main drawback here is, it does not completely provides solution for eradicating the attack rather it only mitigates the effect of jammer when they cause attack over the communication channel. Paper [2] focuses on establishment of pre-shared keys during the presence of any jammer attack. The best example for this paper is use of Bluetooth communication over mobile phones in wireless environment. Normally this creates a circular dependency problem among the communicating nodes. In order to break this dependency, anti-jamming technique was introduced here that enable both sender and receiver to communicate even in the presence of jammer. The main disadvantage regarding this is it cannot be utilized. To overcome this difficulty, improvement for broadcast communication is put forth at paper [3] utilizing Uncoordinated Direct Sequence Spread Spectrum instead of Uncoordinated Frequency Hopping (UFH). Furthermore improvements were made regarding jamming attach. The work in [4] illustrates the attack of jammer can be found earlier at the physical layer than proceeding further at any other layers. For this a code tree technique is handled to mitigate the problem created by the jammer. To describe an efficient mechanism for handling all these drawback, my paper illustrates a mechanism for efficiently communicating between nodes without any pre-shared secrets. And also enable our approach to retain long lived communication among interacting nodes.

6. Performance Evaluation

We evaluate the performance of TREKS in terms of the Packet Loss Rate (*PLR*) as a function of communication/jammer energy, computation cost, and storage cost. We will also focus on two jammers: (1) additive white Gaussian jammer (whose energy is reduced by a factor *n*), and (2) jammers spreading a signal with the receiver MAC address. Without knowing the beginning of the transmission, the jammer is forced to operate as a memory-less jammer.

Simulation Setup: We use NS-2 to simulate the communication, jamming, and message extraction under various settings of the configurable parameters to depict different types of jammers under different scenarios. All the graphs are based on 10K simulation runs of same parameter setting. The variables of our simulations are:

Table 2: Simulation Parameters

Spreading factor, n	100
Packet size, l	1033 bits
Key size, n	19
Jammer power to Signal power Ratio, JSR	[1,…,100]
Normalize signal power	0 dBW
Noise power	-20 dBW

6.1 TREKS vs. Gaussian Jammers

We consider the case where the sender and receiver communicate under a white Gaussian jammer. Interference results in Gaussian noise of energy reduced by a factor *n*.

Packet Loss Rate (PLR)
The *PLR* under our model implies one of the following:
(a) Key Infer Failure
(b) EoM missing and
(c) High BER

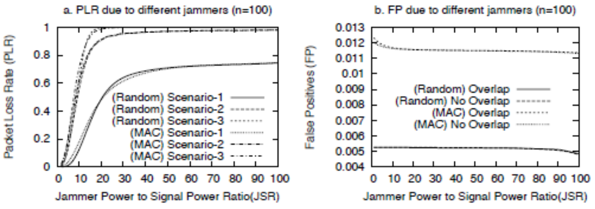

Fig. 5 Expected result in computing jammer performance.

False Positives
The number of False Positives (FP) encountered during the EoM detection process affects the performance of TREKS in terms of its computational delay. In fact, we use the PLR and the number of FPs observed while running TREKS at a fixed noise level of 0dB to choose the peak detection threshold used in Algorithm-2.The increase in computation cost is negligible compared to the decoding cost, which itself is less than double the cost of decoding in traditional SS.

Computation Cost
Operation Using GPU Lab Computer

- FFT benchmark 1*ms* 28*ms*
- Key Inferring - 1*ms*
- Signature Verification - 1*ms*

Expected result shows the computation cost of TREKS performed in computer versus using a GPU NVidia GeForce8800 GTX. Using the latter, we can accelerate the FFT computation by 28 times as revised from previous research. The specification of our lab computer is a 64-bit Intel(R) Core(TM) 2 CPU 6400 @2.13GHz with 3GB memory. It clearly shows that with appropriate Off-the-shelf hardware, TREKS can operate in real time with its total execution time under 3ms.

Storage Cost
The storage cost of TREKS accounts for total number of messages recovered at the end of message extraction and the size of the FIFO used in buffering the signal.

6.2 TREKS vs. λ-Jammers

Consider a discretized time with timeslots of duration *nl* chips. We define two different kinds of jammers that take parameters λ and *JSR*. λ represents the probability that a jammer sends a jamming message at a given timeslot that corresponds to discretization of a Poisson memory less jammer to a Bernoulli jammer, and *JSR* is the jammer to signal power ratio. The cost of the jammer is λ *(JSR*, and its goal is to maximize the *PLR* for a given budget. In our simulation, we assume that the sender is always sending messages. Note that the actual jammer impact will be less than the simulation graph's because the jammer does not know when a transmission occurs. Thus, a source transmitting with probability μ would cause a jammer efficiency decrease by a factor of $1/\mu$.

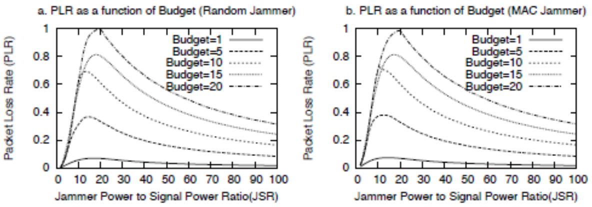

Fig. 6 Expected result in computing jammer performance under fixed budget.

Jammer Types in our simulation:

- (Random) Jammer-1: Inserts an *l*-bit message, each bit spread with a random PN-sequence.
- (MAC) Jammer-2: Inserts an *l*-bit message, each bit spread with the PN-sequence generated using the MAC address of the receiver as the seed.

Jamming Scenarios: Consider a data message that occurs inside a two timeslot (TS) window. Now, a jammer message might occur in the first, second, both or none of the timeslots. This gives rise to following possible scenarios:

- Scenario-1: Jammer message occurs in the first TS.
  Impact: Key inferring.
- Scenario-2: Jammer message occurs in second TS.
  Impact: EoM detection.
  Packet Loss Rate (PLR)
- Scenario-3: Jammer message intersects both TS.
  Impact: Key inferring and EoM detection.
- Scenario-4: Does not occur during those two TS.
  Impact: None.
- Scenario-5: Jammer's packet is perfectly synchronized with the sender packet at the receiver side.
- *Impact*: If perfect synchronization was possible, then there is a 0.5 probability that the last bit of the message is jammed, hence causing to miss the EoM.

Case of the MAC Jammer: The MAC jammer outperforms the Random jammer only in terms of the numbers of FP produced. However, Figure 7 shows that by the third stage of key inferring, almost all of the FPs are detected. Thus, its impact in terms of computation and delay is negligible compared to decoding cost. In terms of PLR, it is a very close race between the MAC jammer and the Random jammer with MAC jammer winning by a slight margin. This is simply because only the last bit of the message is spread with receiver's MAC address.

Case of Perfect Synchronization (Scenario 5): We believe that it is very hard for the jammer to attain Scenario 5, i.e., achieve perfect synchronization, because under our mechanism the jammer does not know when the communication is happening, and only one (last) bit of the packet is actually spread with receiver's MAC address. Therefore, the probability of Scenario 5 is $1/n$.

4. Conclusions

We introduce a method for achieving SS anti-jamming without a pre-configured key sharing approach. Our approach is supposed to obtain nil energy overhead in comparison with conventional SS communication. Our solution relies on intractable forward-decoding and efficient backward-decoding mechanisms. We propose several algorithms to optimize the decoding and show that the computational cost of despreading is less than twice the conventional SS cost. Our method has additional benefits of delayed detection, destination-oriented transmission making jamming infeasible and keeping its impact to minimal by prohibiting jammers from simultaneously jamming multiple receivers. Also it enable long lived communication by computing energy efficiency through varying paradigm.

References

[1] P. Tague, M. Li, and R. Poovendran, Probabilistic Mitigation of Control Channel Jamming via Random Key Distribution, Proc. IEEE 18th Ann. Int'l Symp. Personal, Indoor, and Mobile Radio Comm. (PIMRC), 2007.

[2] M. Strasser, C. Popper, S. Capkun, and M. Cagalj, Jamming Resistant Key Establishment Using Uncoordinated Frequency Hopping, Proc. IEEE Symp. Security and Privacy (ISSP), 2008.

[3] C. Popper, M. Strasser, and S. Capkun, Jamming-Resistant Broadcast Communication without Shared Keys, Proc. USENIX Security Symp., 2009.

[4] J. Chiang and Y.-C. Hu, Cross-Layer Jamming Detection and Mitigation in Wireless Broadcast Networks, Proc. ACM MobiCom, 2011.