ORIENTAL JOURNAL OF
COMPUTER SCIENCE & TECHNOLOGY
An International Open Free Access, Peer Reviewed Research Journal
Published By: **Oriental Scientific Publishing Co., India.**
www.computerscijournal.org

ISSN: 0974-6471
June 2012,
Vol. 5, No. (1):
Pgs. 107-114

Providing Security to Wireless Packet Networks by using Optimized Security Method

N. PAPARAYUDU, G. SURESH KUMAR and J. SRIKANTH

Department of Computer Science and Engineering, Aurora's Engineering College, Bhongir, (India).
E-mail: jsrikanth@aurora.ac.in

(Received: February 12, 2012; Accepted: June 04, 2012)

ABSTRACT

Now-a-days technology is growing very fast, due to rapid development of the technology in computer arena, communication through network become a habit to the users. Communication through network is happen using two channels i.e., by connection oriented and connection less. At present users prefer wireless networks for communication and transferring data due to its flexibility. So in this paper we are focusing on wireless networking, as it is not reliable we are proposing an optimized security technique to provide security to the communication on wireless. In this paper we mainly focus on packet scheduling which plays the vital role in the transmission of data over wireless networks. We are using optimized security technique to secure the packets at initial level itself while scheduling the packets.

Keywords: Real-Time Packets, Packet Scheduling, Wireless Networks, Security, Cryptography, Secret key, Bandwidth.

INTRODUCTION

The quirky thing about a wireless network is that you cannot always see what you are dealing with. In a wireless network, establishing connectivity is not a simple task like plugging in a cable, providing physical security is not easy by keeping unauthorized individuals out of a facility, and troubleshooting even trivial, issues can sometimes result in a few expletives being thrown in the general direction of an access point[4].

It should be noted that supporting efficient and reliable data transmission, especially real time data transmission, over wireless networks is extremely difficult and challenging because wireless networks must be facing more complicated environments compared with conventional wired networks.

For instance, wireless networks could be disturbed by radio wave and thunderstorms or blocked by physical objects like mountains or skyscrapers. Even worse, high mobility coupled with a variety of explosively increased users makes existing security policies in wireless networks inefficient or even useless, meaning that wireless networks can be easily attacked by computer viruses, worms, spy wares, and similar threats. These security threats cause downtime or continual patching in wireless networks and thus lead to severe disruption in wireless commercial business.

Therefore, boosting security of wireless networks has become one of the most important issues in the arena of wireless communications[1].

With the rapid growth of needs for wireless multimedia applications and wireless data services, it is expected that the future broadband wireless networks will support the transmission of heterogeneous classes of traffic (e.g., realtime and non-realtime traffic flows). The design of broadband wireless networks introduces a set of challenging technical issues. Among all these issues that need to be resolved, packet scheduling problem is one of the most important. It is well known that, the bandwidth resource of wireless networks is very scarce. Scheduling algorithms, which are in charge of the bandwidth allocation and multiplexing, have major influence on the network performance[2].

However, in packet cellular environments, user mobility and wireless channel error make it very difficult to perform either resource reservation or fair packet scheduling. While there have been some recent efforts to provide resource reservation for mobile flows in packet cellular networks, the problem of fair packet scheduling in wireless networks has remained largely unaddressed. In fact, even the notion of fairness in shared channel wireless networks has not been precisely defined[3].

At the packet level wireless networks are similar to wired networks in most ways. Wireless networks still use TCP/IP for data communication and abide by all of the same laws of networking as wired hosts. The major difference between the two networking platforms is found at the lower layers of the OSI model. Wireless networks communicate by sending data over the air as opposed to sending data across a wire. The air that wireless data is communicated on is a shared medium, and because of that special consideration must be given at the physical and data link layers to ensure that there are no data collisions and that data can be delivered reliably. These services are provided by different mechanisms of the 802.11 standard[4].

The primary difference between wireless and wired packets is the addition of the 802.11 header. This is a layer two header that contains extra information about the packet and the medium it is transmitted on. There are three types of 802.11 packets; data, management, and control.

Management

These packets are used to establish connectivity between hosts at layer two. Some important subtypes of management packets include authentication, association, and beacon packets.

Control

Control packets allow for delivery of management and data packets and are concerned with congestion management. Common subtypes include Request-to-Send and Clear-to-Send packets.

Data

These packets contain actual data and are the only packet type that can be forwarded from the wireless network to the wired network. [4]

In this paper we are proposing a blow fish security technique to encrypt the data packets at the time of packet scheduling is done, due to this decryption of the packet will not be possible by the attacker on the packet even though the packets are delay.

That being said, securing wireless networks will continue to be a challenge for the foreseeable future.

I. Related Work

In the literature, there are many schemes and protocols devoted to deal with the problem of packet scheduling. We broadly classify existing scheduling algorithms into three categories. A brief introduction of them follows.

A. Algorithms in wireline environment:

Algorithms of first come- first-served (FCFS) and round robin (RR) are first developed for wireline networks. Their original versions and improved versions (e.g., Weighted Round Robin and Deficit Round Robin) are underlying schemes for wireless networks because of the simplicity and ease of implementation. Their drawback lies in the lack of consideration on the issues of bandwidth utilization and fairness guarantee.

B. Algorithms in wireless environment with GE-model:

The classical two-state Gilbert-Eilliot (GE) model is firstly used to model the wireless link variation. The channel is simply described to be either in "good" or "bad" state in this model. A survey of this class of scheduling algorithms can be found in[5]. Work in[6] devised Idealized Weight Fair Queuing (IWFQ) algorithm. In[7], Channel-condition Independent Fair Queuing (CIF-Q) algorithm was put forward. Both algorithms of IWFQ and CIF-Q need to simulate a virtual error-free fair queuing system, and try to schedule packets in the same order as the ideal reference system does. IWFQ and CIF-Q differ in the way they compensate the lagging flows. Authors of [8] presented Token Bank Fair Queuing (TBFQ) algorithm, which uses token pools and token bank to keep track of the service status of each flow, and dynamically regulate the flows' priorities to occupy the channel resource. All these three algorithms achieve good performance tradeoff among the three performance issues aforementioned. But they only work under the GE channel model, which is too coarse to characterize the fluctuation of wireless channel condition.

Since security concern plays a vital role in the design and development of wireless mobile commercial applications, international wireless organizations, wireless equipment providers and academic researchers made extreme efforts in maximizing the features of existing security mechanisms and finding innovative security policies of wireless networks. IEEE improved the security character of 802.11 by designing 802.1X and 802.11i for WLAN[9]. 802.1X, a port-level access control protocol, provides a security framework for IEEE networks, including Ethernet and wireless networks. The 802.11i standard, also still in draft, was created for wireless-specific security functions that operate with IEEE 802.1X. Cisco provides the solutions for wireless applications by using strong encryption technology and providing unified WLAN[10]. Papers addressing the security problems also provide valuable solutions for wireless business applications[11-14].

However, most of the efforts were made at the levels of protocols or systems; most existing approaches were focused on non-real time wireless applications. Packet scheduling plays an important role in achieving high performance in real-time wireless networks. A real time scheduler needs to guarantee both security and real-time constraints of packets even in the presence of hardware and software faults [15,16]. Real time scheduling algorithms can be classified into static[17,28] and dynamic[19,20] strategies.

II. Methodology

A. Problem Definition

According to Xiao Qin, Mohamed Alghamdi, Mais Nijim, Ziliang Zong, Kiranmai Bellam, Xiaojun Ruan, and Adam Manzanares *et al*, they are concentrating on the delay time of the packet delivery based on time scheduling; they are providing security at the time administrator identifies the delay of packet. The motto of this paper is to provide the security at the initial level while the packet is scheduled so that the attacker cannot do any kind of modification to the packet even though they trace the packet here we are concentrating on the security of the packet.

In this paper we are discussing how to secure the packet from the attackers instead of depending on the delay time of the packet and think that the packet is damaged, in the proposed system the duty of the admission controller is to check with the packet and its structure if it is in the correct format the admission controller pushes the packet.

B. Solution

As per the discussions, we are providing security to the packets while the packet is scheduling using BlowFish algorithm.

a) BlowFish

Blowfish is a variable-length key block cipher. It does not meet all the requirements for a new cryptographic standard discussed above: it is only suitable for applications where the key does not change often, like a communications link or an automatic file encryptor. It is significantly faster than DES when implemented on 32-bit microprocessors with large data caches, such as the Pentium and the PowerPC[22].

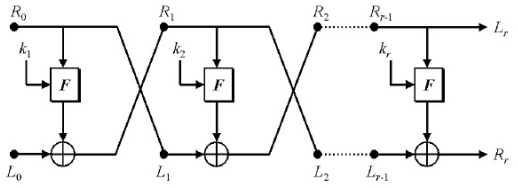

Fig. 1. Feistel Cipher

b) Algorithm

Blowfish is a variable-length key, 64-bit block cipher. The algorithm consists of two parts: a key-expansion part and a data- encryption part. Key expansion converts a key of at most 448 bits into several sub key arrays totaling 4168 bytes. Data encryption occurs via a 16-round Feistel network. Each round consists of a key-dependent permutation, and a key- and data-dependent substitution. All operations are XORs and additions on 32-bit words. The only additional operations are four indexed array data lookups per round[22].

Sub keys

Blowfish uses a large number of sub keys. These keys must be pre-computed before any data encryption or decryption.

1. The P-array consists of 18 32-bit sub keys:
P1, P2,..., P18.
2. There are four 32-bit S-boxes with 256 entries each:
S1,0, S1,1,..., S1,255;
S2,0, S2,1,..,, S2,255;
S3,0, S3,1,..., S3,255;
S4,0, S4,1,..,, S4,255.
The exact method used to calculate these sub keys will be described later.

Encryption

Blowfish is a Feistel network consisting of 16 rounds (see Figure 1). The input is a 64-bit data element, x.
Divide x into two 32-bit halves: xL, xR
For i = 1 to 16:
xL = xL XOR Pi
xR = F(xL) XOR xR
Swap xL and xR
Next i
Swap xL and xR (Undo the last swap.)
xR = xR XOR P17
xL = xL XOR P18
Recombine xL and xR
Function F (see Figure 2):
Divide xL into four eight-bit quarters: a, b, c, and d
F(xL) = ((S1,a + S2,b mod 232) XOR S3,c) + S4,d mod 232
Decryption is exactly the same as encryption, except that P1, P2,..., P18 are used in the reverse order.

Implementations of Blowfish that require the fastest speeds should unroll the loop and ensure that all sub keys are stored in cache.

Generating the Subkeys

The sub keys are calculated using the Blowfish algorithm. The exact method is as follows:

1. Initialize first the P-array and then the four S-boxes, in order, with a fixed string. This string consists of the hexadecimal digits of pi (less the initial 3). For example:
   - P1 = 0x243f6a88
   - P2 = 0x85a308d3
   - P3 = 0x13198a2e
   - P4 = 0x03707344
2. XOR P1 with the first 32 bits of the key, XOR P2 with the second 32-bits of the key, and so on for all bits of the key (possibly up to P14). Repeatedly cycle through the key bits until the entire P-array has been XORed with key bits. (For every short key, there is at least one equivalent longer key; for example, if A is a 64-bit key, then AA, AAA, etc., are equivalent keys.)
3. Encrypt the all-zero string with the Blowfish algorithm, using the subkeys described in steps (1) and (2).
4. Replace P1 and P2 with the output of step (3).
5. Encrypt the output of step (3) using the Blowfish algorithm with the modified subkeys.
6. Replace P3 and P4 with the output of step (5).
7. Continue the process, replacing all entries of the P- array, and then all four S-boxes in order, with the output of the continuously-changing Blowfish algorithm[22].

In total, 521 iterations are required to generate all required sub keys. Applications can store the sub keys rather than execute this derivation process multiple times.

c) The system model and assumptions

In recent studies, a system model is proposed for wireless channel as an NN switch. Although each wireless node may have a single transmitter and a single receiver, it is common that the transmitter and receiver are combined in a transceiver. As such, a node cannot transmit and receive packages simultaneously. In our switch model, there exists a packet scheduler matching transmitters to corresponding receivers. The detailed information regarding the switch model can be found in [21]. In addition to the switch, other three key components in the system include a Security Level Controller (SLC), an Admission Controller (AC), and an EDF (Earliest Deadline First) scheduler as depicted in Fig. 1. This architecture is designed for a link between two nodes in a wireless network. All packets are submitted independently to the wireless link with arrival rates abided by Poisson distribution. The function of the Admission Controller is to determine whether incoming packets can be accepted or not. The Security Level Controller aims at increasing security levels of real-time packets residing in the Accepted queue that can be finished before their deadlines. The EDF scheduler makes use of the Earliest Deadline First policy to schedule admitted packets in which security levels are maximized by the Security Level Controller.

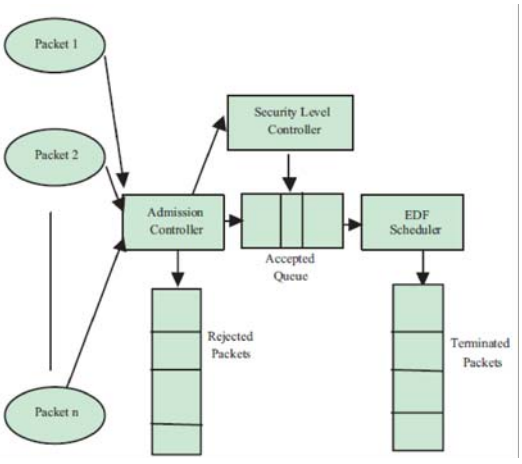

Fig. 1. The Architecture of Network System

d) The packet model

Our packet model assumes that all packets have soft deadline and each packet is independent of one another. We also assume that packets' arrival times follow the classical Poisson distribution. Packet Pi is represented as a tuple (AT_i, PT_i, SL_i, D_i), where AT_i and PT_i denote the arrival time and the processing time of packet *i*. SL_i and D_i represent the security level and soft deadline of packet *i*. Besides, without loss of generality we assume that each packet is assigned a quality of security measured as a security level SL_i that in the range [1, 2, ..., 10], where 1 and 10 are the lowest and highest levels of security. For example, if packet *i* has a value of 1 as a security level, this means that the packet has the lowest security level. Although wireless network devices are unable to determine security levels, packets' security levels can be straightforwardly derived from the security requirements of applications[1].

To calculate the security overhead without loss of generality, we make use of formula (1) to model the security overhead envisioned as the extra processing time experienced by packet *i*.

$$SOi = ETi * (SLi/R) \quad ...(1)$$

where SO_i is the security overhead of packet *i*, SL_i is the security level provided to packet *i*, ET_i is the transmission time of the packet. And *R* is set to 10. Thus, the total processing time WL_i of packet *i* can be expressed as:

$$WLi = ETi + SOi = ETi * (1 + SLi/R) \quad ...(2)$$

e) The SPSS Algorithm

The main goal of this study is to maximize the overall system performance, which reflects the guarantee ratio and security level. To achieve this goal, we designed the SPSS scheduling algorithm with security awareness. SPSS aims to maintain high guarantee ratios while maximizing the security levels. We can accomplish high performance and high security level by applying the Security Level Controller to our SPSS algorithm.

Fig. 2 below outlines the flow chart of the security-aware packet-scheduling algorithm (SPSS) for wireless links. The SPSS algorithm

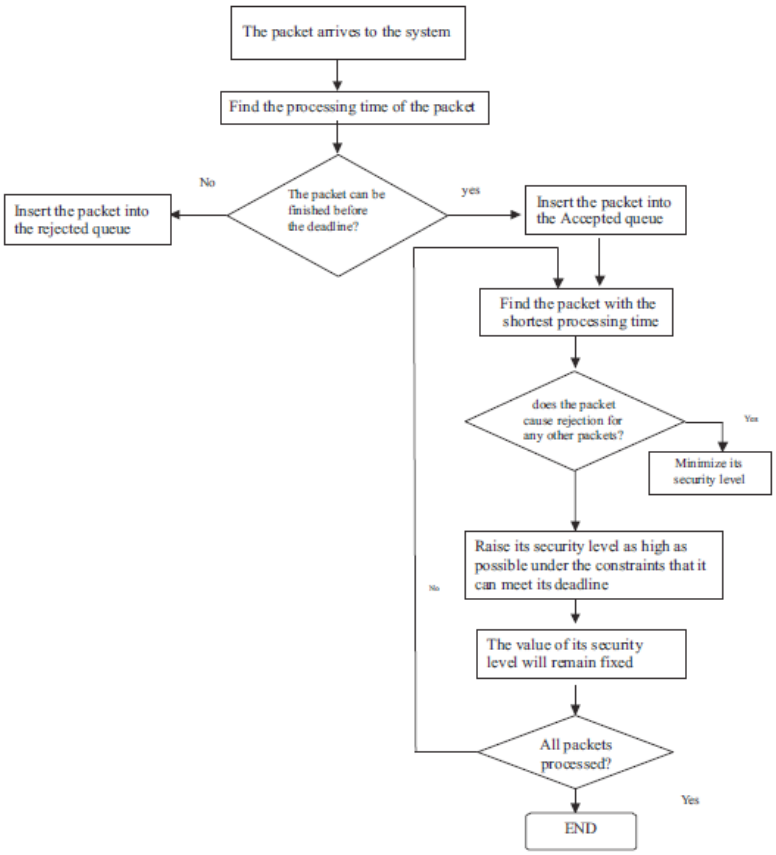

Fig. 2. The SPSS Algorithm

strives to maximize the security level of a packet residing in the accepted queue while making the best effort to guarantee its deadline. If the deadline of the packet can be met, the packet will be admitted in the accepted queue. Otherwise, the packet will be dropped and placed in the rejected queue. The following constraint shows whether the packet is equipped to meet its deadline.

where *STi* is the start time of transmission of the i^{th} packet, *CTi* is the completion time of the transmission, and *di* is the packet's deadline. The packets stored in the accepted queue are scheduled depending on their specified deadlines, meaning that the packets with earlier deadlines will be processed first. The SPSS algorithm initializes the security levels of all packets to the minimum levels. Then, SPSS gradually enhances the security level of each packet *Pi* under the condition that (1) the current packet *Pi* can be transmitted before its deadline; and (2) the deadlines of the packets being processed later than *Pi* also can be guaranteed. The above criterion is important and reasonable because if a packet is admitted to the real-time wireless link, then the packet's timing constraint has to be guaranteed. In other words, the SPSS algorithm ensures that an admitted packet is not adversely affected by subsequently admitted packets[1].

The following steps delineate the procedure of the SPSS scheduling.

Step 1:

Initialize the scheduler; the security values of incoming packets; and the number of rejected packets is set to zero. Wait for any incoming packets.

Step 2:

If a packet *I* arrives and it is the only packet available, process the packet immediately using

its highest security level. The starting time *(STi)* and the completion time *(CTi)* of the packet are calculated. Step 3: All the packets arriving in the scheduler during the time period *[STi, CTi]* are temporarily stored into a waiting queue in the non-decreasing order of their deadlines. The starting time of the next packet *STi* + 1 is set to *CTi*.

Step 4

*T*he admission controller is responsible for deciding whether a packet in the waiting queue can be accepted by considering the deadline of this packet. If the packet's deadline and security requirement can both be guaranteed, the packet will be forwarded into the accepted queue (step 3 and step 5). Otherwise, being put into the rejected queue will drop the packet; the number of rejected packets is increased by one.

Step 5

The security level controller raises the security levels of all the packets residing in the accepted queue as high as possible. The enhancements of the security levels for real-time packets residing in the accepted queue are subject to the following two constraints: (1) Increasing of an accepted packet's security level should still guarantee the deadline of the packet. (2) The increase of security levels must not lead to any rejection of currently accepted packet. Step 6: At this point, the security level *SLi*+1 of the next starting packet is maximized. The packet's completion time *CTi*+1 is calculated. Steps 3-6 are repeatedly executed until all the arriving packets are processed in one run. [1]

CONCLUSION

Providing security at the time of packet scheduling will help the admission controller to push the packets instead of wasting time by rejecting the packet and providing the level of security to the packet based on the rejection. That being said, securing wireless networks will continue to be a challenge for the foreseeable future.

REFERENCES

1. Xiao Qin, Mohamed Alghamdi, Mais Nijim, Ziliang Zong, Kiranmai Bellam, Xiaojun Ruan, and Adam Manzanares, "Improving Security of Real-Time Wireless Networks Through Packet Scheduling", IEEE Transaction on Wireless Communications, Vol. 7, No. 9, Page No: 3273-3279 (2008).
2. Rong Yu, Zhi Sun, and Shunliang Mei , "Packet Scheduling in Broadband Wireless Networks Using Neuro-Dynamic Programming", 1550-2252/$25.00 IEEE, page No: 2776-2780 (2007).
3. Songwu Lu, Vaduvur Bharghavan, R. Srikant, "Fair Scheduling in Wireless Packet Networks", IEEE/ACM Transactions on Networking, Vol. 7, No. 4, Page No: 473-489 (1999).
4. Chris Sanders, "Analyzing Wireless Network Security at the packet Level", Articles Misc Network Security, (2010).
5. Y. Cao and V. O. K. Li, "Scheduling algorithm in broad-band wireless networks," in Proc. of the IEEE vol. 89, no. 1, pp. 76-87, (200l).
6. S. Lu and V. Bharghavan, "Fair scheduling in wireless packet networks," IEEE/ACM Trans. Networking, vol. 7, no. 4, pp. 473-489, (1999).
7. T. S. Eugene Ng, I. Stoica, and H. Zhang, "Packet fair queueing algorithms for wireless networks with location-dependent errors," in Proc. INFOCOM98, pp. 1103-1111, (1998).
8. W. K. Wong, J. Zhu, and V. C. M. Leung, "Soft-QoS provisioning using the token bank fair queueing scheduling algorithm," IEEE Wireless Communication Magazine, (2003).
9. T. Karygiannis and L. Owens, "Wireless Network Security 802.11, Bluetooth and Handheld Devices," chapter 2, http://csrc.nist.gov/ publications/nistpubs/ 800-48/ NIST SP 800-48.pdf.
10. White paper of Cisco, "Building the mobile business with a unified wireless network http://epsfiles.intermec.com/eps files/epswp/CISCO SecuringWirelessLAN

wpweb.pdf.

11. M. A Badamas, "Mobile computing systems–security considerations," Information Management and Security, pp. 134–136 (2001).
12. O. M. Karygiannis, Wireless Network Security, NIST special publication 800–48.
13. K. Siau, E. P. Lim, and Z. Shen, "Mobile commerce: promises, challenges and research agenda," J. Database Management, vol. 12, pp. 4–19, (2001).
14. X. Qin and H. Jiang, "Dynamic, reliability-driven scheduling of parallel real-time jobs in heterogeneous systems," in Proc. Int'l Conf. on Parallel Processing, pp. 113–122, (2001).
15. X. Qin, H. Jiang, D. R. Swanson, "An efficient fault-tolerant scheduling algorithm for real-time tasks with precedence constraints in heterogeneous systems," in Proc. Int'l Conf. on Parallel Processing, British Columbia, Canada, pp. 360–368, (2002).
16. J. C. Palencia and H. M. Gonzalez, "Schedulability analysis for tasks with static and dynamic offsets," in Proc. 19th IEEE Real-Time Systems Symp. pp. 26–37 (1998).
17. T. F. Abdelzaher and K. G. Shin, "Combined task and message scheduling in distributed real-time systems," IEEE Trans. Parallel and Distributed Syst., vol. 10, no. 11, (1999).
18. M. A. Palis, "Online real-time job scheduling with rate of progress guarantees," in Proc. 6th Int'l Symp. Parallel Architectures, Algorithms, and Networks, pp. 65–70 (2002).
19. G. Manimaran and C. S. R. Murthy, "An efficient dynamic scheduling algorithm for multimachine real-time systems," IEEE Trans. Parallel and Distributed Syst., vol.9, no. 3, pp. 312–319, (1998).
20. S. Al-Harthi and R. Rao, "A switch model for improving throughput and power fairness in Bluetooth piconets," in Proc. Globecom, pp.1279–1283 (2003).
21. B. Schneier, "Description of a New Variable-Length Key, 64-Bit Block Cipher (Blowfish)", *Fast Software Encryption, Cambridge Security Workshop Proceedings (December 1993)*, Springer-Verlag, pp. 191-204 (1994).
22. Gupta and S. Gupta, "Securing the wireless Internet," *IEEE Commun. Mag.*, pp. 68–74, (2001).