2009 International Conference on Machine Learning and Computing
IPCSIT vol.3 (2011) © (2011) IACSIT Press, Singapore

Secure Data Transmission for ATA-based Mobile Virtual Storage System

Chee-Min Yeoh [1], Bee-Lie Chai [1], Hoon-Jae Lee [2], Hyotaek Lim [2]

[1] Department of Ubiquitous IT, Graduate School of Design and IT,
[2] Division of Computer and Information Engineering, Dongseo University, Busan 617-716, South Korea

Abstract. With the remarkable advancement in mobile technology and ubiquitous computing, virtual storage system, internet Advanced Technology Attachment (iATA) could be one of the possible solution to solve the storage limitation problem on mobile device. However, transmitting data over an open ubiquitous TCP/IP network without any security defense is exposed to malicious security threats. In this paper, we aim to design a secure data transmission for iATA protocol. Our proposed solution is designed with both security and efficiency in mind to satisfy resource constrained environments like mobile device. Significant experiments are conducted and the results so far prove to be most modest compared to the protection that the scheme can offer.

Keywords: Network Storage, WLAN, Dragon, Dragon-MAC, Mobile Computing

1. Introduction

In a rapid growing ubiquitous computing environment, mobile technology has playing an importance role to strengthen the mobility in collaboration, communication and information sharing. Nonetheless, with enlarging the mobile computation development, it is immense increasing the demand of mobile data storage. Due to data storage restricted, network storage could be one of the solutions. iATA, Internet Advanced Technology Attachment[1][2] allows ATA device commands to be transported over the TCP/IP network and mounted remotely from a wireless connected mobile device.

Although the use of iATA protocol increased the storage of mobile device, it exposed to malicious security attack in this opened ubiquitous network [7]. This is because iATA protocol transmits sensitive IP blocks of storage data over the insecure TCP/IP network without any security defence. It opened to adversaries to snooping, modifying, hijacking TCP connection, launching denial of service attacks or impersonates a legitimate and so forth. These will seriously impact the privacy and quality of the iATA protocol. Furthermore, mobile devices have limitation on computation resources which might increase the cost and huge performance degradation with the existing security protocols, having complicated and heavy security computation.

This paper contributes toward to design an efficient, lightweight and fast as well as support data authentication, confidentiality and integrity, and replay protection on a per-packet basis on iATA-based remote storage service for mobile devices. By looking at the advantages of Dragon [3] stream cipher, which is very fast in key stream generation and high throughput. It is very appropriate to implement and support the confidentiality in mobile device. Moreover, our scheme includes Dragon-MAC [4] which retained the structure of Dragon stream cipher for MAC generation to achieve two-party authentication and data integrity.

The following of this paper is organized as below: In the section 2 giving a brief overview and security analysis of iATA protocol, and then continuous with description of our proposed solution in section 3. Section 4 is our implementation and performance evaluation. Section 5 was concluding this paper.

2. Overview

2.1. Internet Advanced Technology Attachment (iATA)

iATA, internet Advanced Technologies Attachment is a new block-level network storage protocol, which allows sending ATA commands over TCP/IP network for mobile devices. In other words, Clients are allowed accessing to server's ATA disk remotely through internet connection.

Figure 1 is the illustration of TCP/IP layering model view. iATA protocol is an application layer protocol that lies on top of transport layer. iATA protocol will act as mediator to convert ATA command message to TCP/IP packet and send through the internet and iATA also responsible convert received TCP/IP packet message to ATA command message and send to appropriate layer. Similar to others TCP-based protocols, the Protocol Data Unit (PDU) transmits over the internet is encapsulated by TCP header, IP header and Ethernet frame. In the meanwhile, iATA common header segment and ATA command block or configuration/query message had injected into the data field and produced iATA PDU before encapsulated by TCP header as shown in Figure 2.

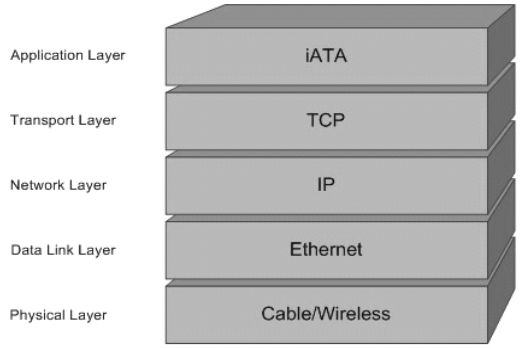

Figure 1: Protocol Stack of iATA

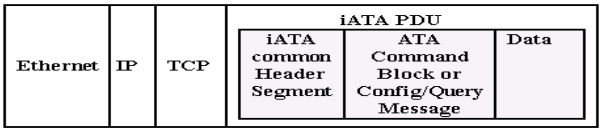

Figure 2: Encapsulation of iATA Protocol Data Unit

The iATA common header segment consists of 16 bytes of variable. It can be use to indicate the type of the following ATA operation message. It is either configuration/query message or ATA command. Besides that, the ATA common header segment consists other importance values as identification, naming, session identity, task tag field are used for session management and ATA request response mapping.

If the ATA common header indicates that this message operation is the configuration/query message, then that message used to exchange out-of-band iATA I/O message such as configuration parameter and system detail. On the other hand, if the ATA common header indicates that this message's operation is ATA command block, then the following message is used to carry read/write operation data. Most of the time, the iATA PDU are carrying ATA command block message. More details can be found in [1] [2].

2.2. Dragon

Dragon is a word based stream cipher was motivated by eSTREAM, ECRYPT stream cipher project and was subsequently selected as one of the Phase 3 candidates [8]. Dragon is constructed using a large single word based NLFSR, non-linear feedback shift register of 1024 bits, a state update function, denoted F function and 64-bits memory, denoted M.

F function is used in both key setup and keystream generation, which is a reversible mapping of six 32-bits words to six 32-bits words. Figure 3 shows the input words are denoted a, b, c, d, e, f and return the output words a', b', c', d', e', f'. The six component functions denoted G1, G2, G3, H1, H2 and H3 in this F function provide algebraic completeness and high non-linearity. These G and H functions are constructed from two 8 x 32-bit s-boxes to virtual 32 x 32 s-boxes.

Dragon has simple keying tactic using the 128-bit key and initialization vector. The 1024-bit internal state initially filled by concatenating the key and the initialization vector and it is divided into eight 128-bit words labelled W_0 to W_7 and then makes extensive use of F function involved 16 iterations of F functions processing the complementation and swapping the eight 128-bits as shown in Table 1.

Keystream generation for Dragon is using the same component with initialization. The large 1024 bits NLFSR divided into thirty two 32-bit words, denotes Bi, $0 \leqslant i \leqslant 31$. Every round, six words from the internal state are used as inputs to the F function including a 64-bit memory component, M which act as

counter. During each round of keystream generation provided the output of a 64-bit word k, updated state B and memory M. The gist of Dragon is designed with both security and efficiency. It is secure against all known cryptanalytic attacks and has efficient and fast rekeying performance. It is very suitable for mobile and wireless communications applications. Specifications of Dragon can be found in [3].

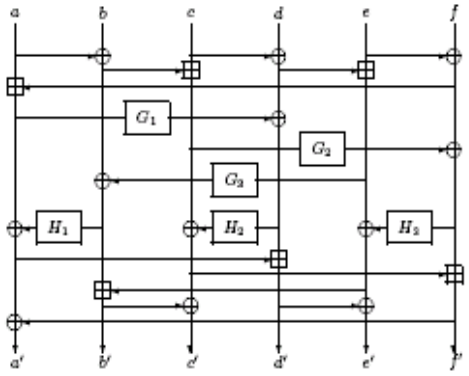

Figure 3: Dragon's F Function

Input = { *K, IV* } (256-bit) Input = { *k, iv* } (128-bit)

1. $W_0 || \ldots || W_7 = K\_K \oplus IV\_K \oplus IV\_IV$ (256-bit)

$W_0 || \ldots || W_7 = k || k' \oplus iv' || iv || k \oplus iv' || k' || k \oplus iv || iv' || k' \oplus iv$ (128-bit)

2. M = 0x0000447261676F6E

Perform steps 3-8 16 times

Table 1: Dragon's Key Initialization Function

2.3. Dragon-MAC

Dragon-MAC is a message authentication code which retained the structure of Dragon stream cipher and shared its F function for MAC generation. It is reversible mapping of 192-bit to 192-bit function to supply 4 bytes output that served as a MAC. Dragon-MAC makes use of the high non-linearity and low autocorrelation effect provided by S-boxes in the construction of the F state update function (described in section 2.2). Table 2 is Dragon-MAC algorithm. The plaintext of data packet is encrypted with Dragon stream cipher with encryption key, K_e and generating ciphertext, Ct in 32-bit words. Subsequently, the MAC encryption key, K_m is fed into the F function structure through input a, b, c, d. F function will go through 16 clock cycles to XOR 32-bit Ct with 32-bit a. Eventually , the 32-bit MAC can be obtained by XOR-ing all the outputs of F function. Dragon-MAC in detail please refers to [4].

Let *Pt* denote the plaintext
Let *Ct* denote the ciphertext
Let K_e denote the encryption Key
Let K_m denote the MAC encryption Key
Let *Ct[i]* denote the *i-th* 32-bit word of ciphertext

1. $Ct = E_{Ke}(Pt)$
2. $\{a, b, c, d\} = K_m$ *(128-bit)* $\{e, f\} = \{0x00004472, 0x61676F6E\}$
3. for *i* from max length down to 1 perform step 4

4. $\{a', b', c', d', e', f'\} = F (Ct[i] \oplus a , b , c , d , e , f)$

Table 2: Dragon-MAC generation

3. Proposed Solution

In this paper, our proposed module consists of Dragon stream cipher as cryptographic encryption algorithm and Dragon-MAC as cryptographic authentication algorithm, this security module is directly embedded into the iATA layer. We proposed to use Encrypt-then-MAC mechanism to protect the iATA PDU. This is because Mihir bellare and Chanathip Namprempre[5] had proven that Encrypt-then-MAC is the most secured and favourable choice to facilitate authenticated encryption scheme.

In the design, first of all we collected the iATA header and iATA data from iATA layer which going to send out to receiver, the iATA header and data will apply Dragon stream cipher encrypt into encrypted iATA PDU. This not only protected the data confidential on iATA data contents but also the iATA header which contains sensitive information related to the disk identity and functionality away from adversaries data packet snooping. Then, the result of this ciphertext will pass into Dragon-MAC algorithm to generate four bytes of Message Authentication Code (MAC) and inserted into the end of the encrypted PDU. Based on

Zoltak et. al[6] and Karlof et al.[7] research, four bytes of MAC length provides a well adequate security level and acceptable implementation of the system. Therefore, the four bytes MAC generated by Dragon-MAC is practical and sufficient for our scenario.

In the typical iATA communication, iATA PDU transported with variance of length in TCP, Transport Control Protocol data segment. However TCP does not have any mechanism to determine the iATA PDU's length. Therefore, the length of message is defined in iATA common header message to specific the starting and ending of the iATA message. In the proposed module, the entire iATA header is encrypted together with iATA data, it's causing the lost of iATA message boundary. As a result, a four bytes field will be added in front of the encrypted iATA PDU to specify the length of the original iATA PDU. Figure 4 illustrates encapsulation of modified iATA Protocol Data Unit in our design. At the receiver side, decapsulation process will be reversed of encapsulation process. Whenever received the encrypted iATA PDU, it will be take the encrypted message passes into the Dragon-MAC algorithms to calculate the four bytes MAC and compare with the received four bytes MAC, if the MAC values is matched it will proceed to decrypte the encrypted iATA PDU with Dragon stream cipher and pass to iATA layer application, but if that is mismatch, receiver will drop that iATA PDU message.

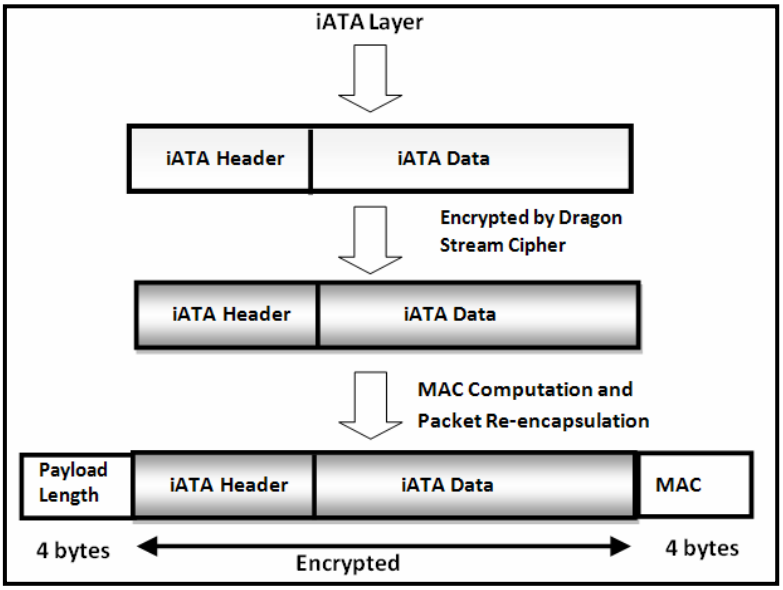

Figure 4: Re-Encapsulation of iATA

During encryption and decryption process, the plaintext of iATA PDU with its padding are queued in a buffer stream, each iteration, 32 bits of plaintext are taken from the buffer and perform an XOR operation with the keystream generated from Dragon stream cipher to produce the ciphertext. The keystream is changed before it enters the next iteration of the encryption process. Padding is needed to avoid keystream synchronizing issues. By using this modified iATA PDU, it is able to secure against active and passive security threat including eavesdropping, masquerade of identity, message modification, insertion, deletion and others.

For the purpose of performance evaluation, we have made some code changes in order to support our security module. Next section will present about the performance analysis of our proposed solution.

4. Implementation and Performance Evaluation

In our experiment, there are four main hardware devices that are needed such as router, wireless access point, Personal Data Assistant (PDA) and Personal Computer (PC). Referring to Figure 5, Site 1 is the iATA client site which consists of 802.11b wireless access point, router and a PDA. The PDA has 300MHz S3C2440 Samsung processor with 64 MB of RAM and equipped with wireless capability. This PDA runs Microsoft Mobile Pocket PC 2003 powered by Windows CE 4.21 and is installed with iATA client driver. On the other site is iATA server site which contains router and a 1GHz Intel Pentium III Coppermine PC with 512MB RAM. This machine runs Fedora Core 6 Linux Kernel 2.6.18 and installed with iATA server

driver which exported 2GB of ATA disk over the network. The utilization of two different operating system platforms between iATA server and client demonstrated that iATA protocol is platform independent and works well in two non-identical OS platforms.

In our experiment, we employed Windows CE Remote Performance Monitor, version 5 to inspect the memory utilization of our proposed solution. This utility provides the memory statistic by measuring current memory utilization, in which 0 indicates no memory use and 100 indicates full memory use. We measure the memory consumption from the moment we load the iATA program to un-mount iATA disk which includes mount iATA disk, read 10MB raw file and so on. In our experiment, we compare the memory usage of original iATA protocol with our proposed solution which equipped with security scheme.

Figure 6 depicts the memory consumption of original iATA protocol and iATA protocol with security scheme. At time 5, the memory usage is increase from 51 to 52 for both versions when iATA protocol is mounting the disk. However, when iATA protocol is executing read operation at time 15, an obvious increment can be seen for iATA protocol with security scheme compare with the original version at the same level of memory consumption which is 52. Even though there is an addition memory incurred in our proposed solution, it is considered modest compared to the protection that it can offer.

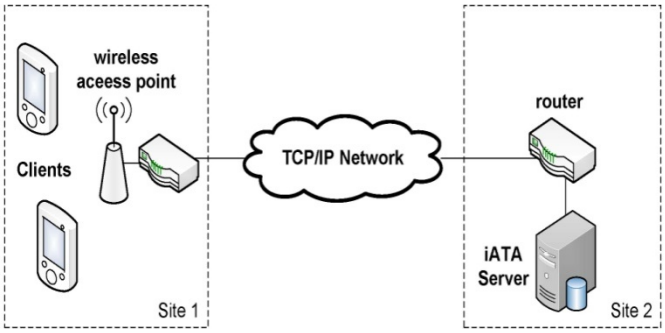

Fig. 5: Environment Setup

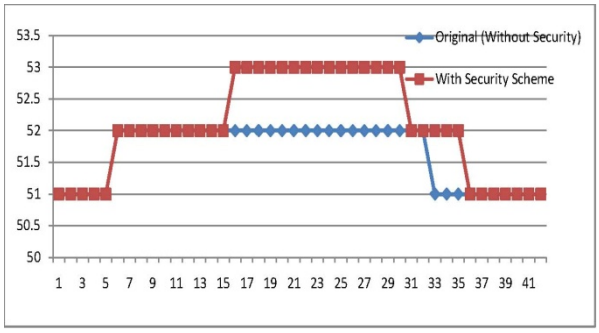

Fig. 6: Memory Utilization

5. Conclusion

In this paper, we have proposed a lightweight and secure data transmission for ATA-based mobile virtual storage system (iATA). In our security scheme, we employed Dragon as our cryptographic encryption algorithm and Dragon-MAC as our cryptographic authentication algorithm to provide per-packet shield against various malicious security attacks. Several directions for future research arise from our solution. First, we intend to compare the performance with other security algorithms in iATA protocol. Then, we would like to assess the performance of our approach in a multiple client environment and evaluate the CPU utilization.

6. References

[1] Yu-Shu They, Chee-Min Yeoh, Hoonjae Lee, Hyotaek Lim, Design and Implementation of ATA-Based Virtual Storage System for Mobile Device. *MUE 2008*: 490-495

[2] Chee-Min Yeoh, Yu-Shu They, Hoon-Jae Lee, Hyotaek Lim, Design and Implementation of iATA on Windows CE Platform: An ATA-Based Virtual Storage System, cmc, vol. 3, pp.85-89, *2009 WRI International Conference on Communications and Mobile Computing*, 2009

[3] K. Chen and M. Henricksen and W. Millan and J. Fuller and L. Simpson and E. Dawson and H. Lee and S., Moon, Dragon: A fast word based stream cipher. *ECRYPT Stream Cipher Project Report* 2005/006, 2005

[4] Shu Yun Lim, Chuan Chin Pu, Hyo Taek Lim, and Hoon Jae Lee, Dragon-MAC: Securing Wireless Sensor Networks with Authenticated Encryption, *Cryptology ePrint Archive*, Report 2007/204

[5] M. Bellare and C. Namprempre.: Authenticated encryption, Relations among notions and analysis of the generic composition paradigm, *Proceedings of the 6th International Conference on the Theory and Application of Cryptology and Information Security: Advances in Cryptology, Lecture Notes In Computer Science*, Vol. 1976, pp. 531-545.

[6] B. Zoltak, An Efficient Message Authentication Scheme for Stream Ciphers, Cryptology ePrint Archive 2004

[7] B. Aboba,J. Tseng, J. Walker, V. Rangan, F. Travostino, RFC3723: *Securing Block Storage Protocols over IP, RFC Editor United States Year of Publication:* 2004

[8] ECRYPT Node of Excellence. eSTREAM PHASE 3. Available at http://www.ecrypt.eu.org/stream/index.html.