Automatic Key Renovation over Serial Link

Peeyush Jain
Centre for Development of Advanced Computing
Mumbai, India

Zia Saquib
Centre for Development of Advanced Computing
Mumbai, India

Sharda Saiwan
Centre for Development of Advanced Computing
Mumbai, India

ABSTRACT

Key management is a core mechanism to ensure the security of information in communication channels. Secret key is used for encryption and decryption in cryptographic security modules. Security of communication relies on secure and robust distribution of secret keys. Key need to be renewed periodically to prevent compromise the key. Most of the key distribution protocols consider the pre-distribution of secret key only but did not handles the Key Renovation. The Automatic Key Renovation has several issues when implemented on serial communication link. When a single channel is used for data communication as well as for key distribution, both communicating systems need to be well synchronized at the time of renewal of key. This paper focuses on such synchronization issues during the key renovation and introduces a solution to resolve it. The proposed solution is independent of the platform and the underlying key distribution protocol. It also ensures zero data loss and minimum delay on data communication.

Keywords

Key Renovation, Information Security, Serial Communication, Key Distribution, Secret Key, Bump-in-the-wire, Synchronization

1. INTRODUCTION

Serial communication [1] is the process of sending a stream of data sequentially over a single channel. Whereas in parallel communication [2], data is sent in parallel over multiple channels and synchronized at the receiving end. Serial communication is used for most computer networks where the cost of cable and synchronization difficulties makes parallel communication impractical.

A server and a client communicating over a serial link i.e. two way communications. To secure the communication, a Bump-in-the-wire (BITW) solution [3, 4] is introduced as shown in Figure 1. BITW device work as an Encrypting Transceivers to encrypt the plain text to cipher text and vice versa. With BITW solution, legitimate communication still flows seamlessly between the master and remote devices, but a potential attacker cannot manipulate, inject, or interpret the sensitive contents of the encrypted frames. BITW solutions provide security without replacing or modifying the existing legacy systems. To operate, a BITW has two serial ports, one designated plaintext call as local interface and the other designated cipher text call as remote interface. A message received from a Server or client on a BITW's local interface port will be encrypted and sent out the BITW's remote interface port. Encrypted messages arrive at the BITW at other end to be decrypted and verified, and sent to the local attached unit. In addition, BITW can be enabled to transfer commands and responses to control the cipher text port.

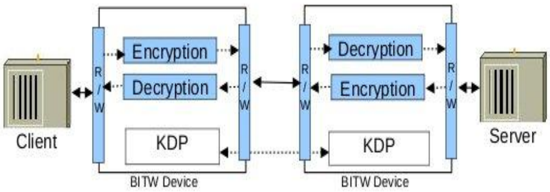

Fig 1: Communication over serial link through BITW device

Data encryption-decryption takes place when data is passing through BITW using the same key at both ends. For example when Server sends the data to client, data will encrypt through the BITW device at Server side and forward to client on remote interface. At the other end data will be decrypted at BITW device located at client side. In the Similar manner client will sends the data to server. Same key is stored at both end BITWs at server and client for encryption and decryption. A key distribution protocol [6, 10] is used to establish the secret key at both ends. In Figure 1, KDP (Key Distribution Protocol) uses the same channel for key distribution as used for data communication. Lot of work has been done on cryptography protocols to provide the cyber security on serial links [3,4,5,9]. Still there is need to solve the Automatic Key Renovation problem over the serial link. Performing key renovation on same channel, on which the data is already running have several issues, will be discussed in Section 2. An efficient algorithm is proposed for Automatic key renovation on serial link, will be discussed in Section 3. Finally includes the conclusion in Section 4.

2. KEY ISSUES

To prevent the compromise of a secret key, key needs to be renewed periodically. Resources like communication link, memory and computational power is required for key renovation [7]. There are synchronization issues when key is to be distributed on the same channel on which two parties are already communicating. For example if key distribution is started over a same channel on which data is already passing then how the receiving device differentiate that data coming from other end is for key distribution or regular data. So an efficient mechanism to synchronize the key distribution and regular data communication is required. Following issues need to be considered

Encryption and decryption of data must be done usingsame key i.e. if encryption of data takes place at one end using a key then decryption must be done using same key at other end.

Key renovation mechanism should not initiate until all the data encrypted at one end will be decrypted at other end. Otherwise after encryption of data at one end, key renovation start immediately before reach the data at other end. So key will be updated at both end and decryption of data, which was encrypted using old key, takes place using the new key that causes wrong output.

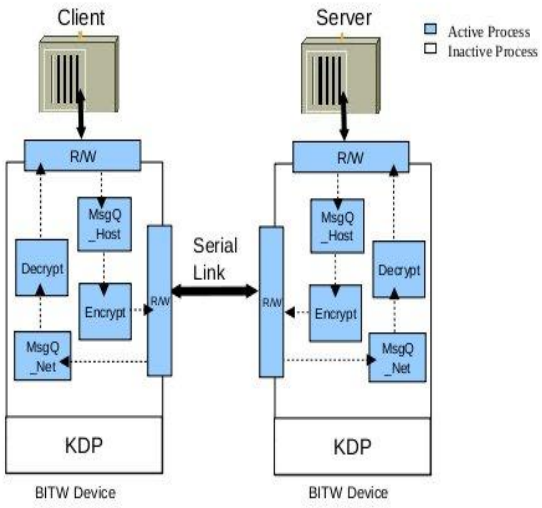

Fig 2: Data Communication (DC) Mode

There are many blocking and non-blocking methods used for synchronous communication [8]. A sender can block sending more data until the current one has been successfully reached at the destination. In non-blocking methods, the sender stores data in a buffer and continue sending without waiting for the successful arrival at the destination. On the other hand, a receiver can also block reading if there is no data available or it can continue reading without blocking. In second case, the receiver reads the data if available or otherwise indicates that no data available. Several clock synchronization algorithms are also available for synchronous communication in distributed systems [8]. Server-client communication can be synchronized by initially setting their internal clocks and maintained by a central server. But in case, the data is encrypted by sender with in a timestamp t1 and decrypted by receiver in another timestamp t2 i.e. there is a time interval (t2 - t1) to reach the data at destination. If the key is renewed in between this time interval (i.e. t2 – t1), then data will be incorrectly decrypted at the receiving end, because the receiver will try to decrypt the same data with new key which is encrypted by sender with old key. Therefore, the data counter based approach is used. The amount of data exchanged is used to trigger the key renovation.

3. PROPOSED SOLUTION NOTATIONS

Port0 ------> Serial port on BITW device designated for plaintext to communicate with host

Port1 ------> Serial port on BITW device designated for cipher text to communicate with network

MsgQ_Host ------> Message Queue to store the data received at Port0

MsgQ_Net ------> Message Queue to store the data received at Port1

Counter1 ------> Counter to count the number of encryptions at BITW at Server side

Counter2 ------> Counter to count the number of decryptions at BITW at Client side

Counter1 = Counter2 = 0

Splchar ------> Special Character which never occur in data traffic

RN -----> Read from Network at Port1

WH -----> Write to Host at Port0

RH -----> Read from Host at Port0

WN -----> Write to Network at Port1

E -----> Encryption

D -----> Decryption

KDP -----> Key Distribution Protocol

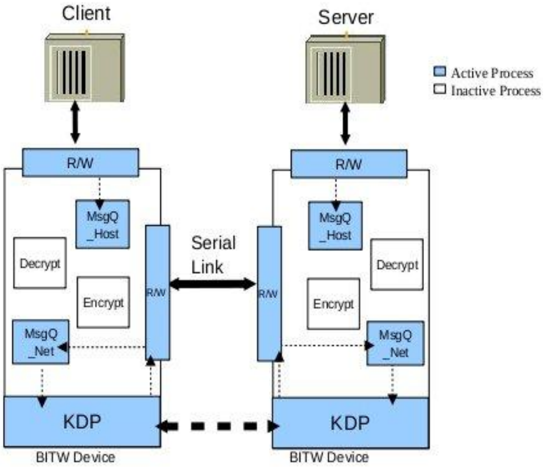

Fig 3: Key Distribution (KD)

4. ALGORITHM

As discussed in the previous sections, there are synchronization issues in automatic key renovation during regular data communication over same channel. As shown in Figure 2, there is a serial communication taking place between Server and Client. Server sends the data to client through BITW device on single channel serial link. BITW at Server side store the data in a MsgQ_Host, which is to be encrypted using a pre-distributed key and pass to the remote interface. At the client end, BITW stores the data in MsgQ_Net and decrypt it and pass to the client. Similarly communication takes place from client to server as well.The problem comes when key renovation is required. To renew the key, the key distribution protocol is called, which starts communicating with other end on same channel on which data communication is already going on. But the problem is to differentiate the data coming from other end is for key distribution or normal data communication. There are some other synchronization issues as discussed in previous section which need to be considered during key renovation. An algorithm to solve such synchronization issues is proposed. There are two modes of communication defined i.e. data communication (DC) mode and key distribution (KD) mode. In DC mode, regular data transfer through serial channel using BITW device for encryption/decryption as shown in Figure 2. Now when key need to be renewed, system need to be switched in KD mode and communication will start between key distribution protocols at both ends through same channel using the MsgQ_Link as in DC mode as shown in Figure 3. In KD mode, encryption of data coming from host is inactive. To prevent the data loss, MsgQ_Host at both end continue to receive the data from host and store until key renovation complete.

At the time of key renovation, system needs to switch from one mode to other i.e. to start the key renovation, system switch to KD mode and after completion it will switch back to DC mode. But an Efficient algorithm is required to switch from one mode to other and also should take care of other synchronization issues discussed in earlier section. An algorithm is proposed, which consider such synchronization issues and efficiently switch the system from one mode to other mode without any data loss.

At Server side

```
while(1){
        if(Counter1 < N) {
                pop(MsgQ_Host) --> buffer
                encrypt(buffer) --> buffer
                Counter1 ++
                buffer --> write(Port1)
        }
        else {
                stop  pop(MsgQ_Host)
                pop(MsgQ_Net) --> buffer
                decrypt(buffer) --> buffer
                while(buffer != Splchar) {
                        buffer --> write(Port0)
                        pop(MsgQ_Net)     --> buffer
                        decrypt(buffer)       --> buffer
                }
                KDP()
                Counter1 = 0
        }
} //End
```

At Client side

```
while(1){
         if (Counter2 < N) {
                pop(MsgQ_Net) --> buffer
                decrypt(buffer) --> buffer
                Counter2 ++
                buffer --> write(Port0)
        }
        else {
                stop  pop(MsgQ_Host)
                encrypt(Splchar) --> buffer
                buffer --> write(Port1)
                KDP ()
                Counter2 = 0
        }
}//End
```

Explanation

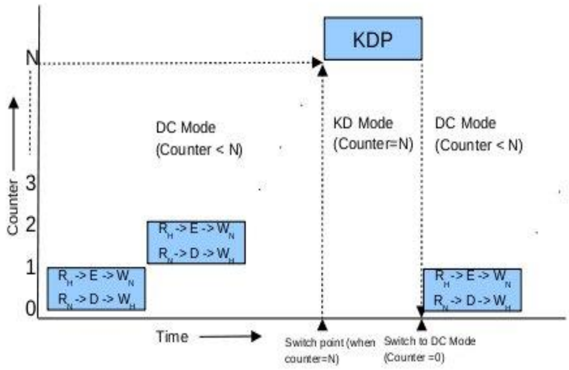

Fig 4: Switching between DC and KD mode

Using this algorithm, system will switch to KD mode synchronously when key renovation is required and come back to DC mode after completion of key renovation. Number of encrypted messages send from server to client through Port1 is considered as a parameter for key renovation. Key will be renewed when counter reaches to 'N'. Initially Counter is set to '0' and increase with number of encryption happens at server side. When counter reaches at 'N', system switch to KD mode and key renovation will start to renew the key using key distribution protocol and counter will be reset to '0'. After completion of key renovation, system again switch to DC mode as shown in Figure 4 and normal communication will continue through BITW device at both ends securely using a new key for encryption-decryption.When counter reach to 'N', Server will immediately stop to pop the data from MsgQ_Host for encryption and block the data forward to client as shown in Figure 3, but MsgQ_Host (at server side) will continue to get the data from server and store until key renovation will complete. Then it waits for the client for confirmation to start key distribution protocol. BITW at client also have the same counter, which will increase with decryption of data coming from other end. When counter at client side is also reached to 'N', it will complete the running process and block to pop the data from MsgQ_Host for encryption and send a special character (Splchar) encrypted with current key, not used in data communication, as a confirmation message to server to start the key renovation process.Simultaneously it calls key distribution protocol (KDP) and wait for server response to start key distribution. When server will get the special character from client side, immediately it calls the key distribution function. System is now in KD mode. In KD mode, encryption and decryption modules in BITW device at both ends would be inactive. MsgQ_Net will continue to get the data from other end, as system is now in KD mode and data is for key distribution so instead of forwards it for decryption, it will pass the data to key distribution protocol as shown in Figure 3. MsgQ_Host at client side also will continue to get the data from client and store until key renovation will complete. After completion of key renovation, counter will be reset to '0' and the system will switch back to DC mode. Once the system enters in DC mode, both BITW devices start processing messages from their MsgQ_Host using new keys and resume data communication on same channel until next key renovation called.

4.1 Advantages

Ensures data encrypted at one end will be decrypted at otherend using the same key.

Ensures Key renovation starts simultaneously at both ends.

Ensures on channel data is completely processed before the key renovation starts.

No data loss: MessageQ are used to store the data at the time of Key renovation.

Saving resources in terms of communication. No need for extra communication resources required.

5. CONCLUSION

In cryptographic security modules, security of communication relies on secure and robust distribution of secret keys. Key need to be renewed periodically to prevent compromise the key. The Automatic Key Renovation has several synchronization issues when implemented on serial communication link. When a single channel is used for data communication as well as for key distribution, both communicating systems need to be well synchronized at the time of renewal of key. Server and Client are communicating with each other over serial channel. The problem comes when key renovation is required. As there is single channel, so problem arises to differentiate the data coming over serial channel is normal communication or for key distribution. Also there are issues of delay and data loss during key renovation. In this paper, an efficient key renovation algorithm is proposed to resolve such problems. The communication is divided in two modes: DC mode (Data Communication) and KD mode (Key Distribution). In DC mode, data is transferred between Client and server through serial channel. When key need to be renewed, system will switch to KD mode and the algorithm proposed will take care of synchronization issues. This algorithm is used to synchronously switch the system from one mode to other without any data loss. It also take care all other synchronization issues as discussed. After the renewal of key, the system will switch back to DC mode. Once the system enters in DC mode, Client and Server resume data communication on same channel using the new key until next key renovation called. Introduced work provides a reasonable resource saving. In future, this algorithm will be very useful in security modules during the key renovation when a single channel is used for data communication as well as for key distribution.

6. REFERENCES

[1] TechnicalTutorial,20021206,http://iitkgp.vlab.co.in/userfiles//SERIAL%20COMM.pdfhttp://www.bookrags.com/research/serial-and-parallel-transmission-csci-02/

[2] P.P.TsangandS.W.Smith.YASIR:ALowLatency,HighIntegrity Security Retrofit for Legacy SCADA Systems (Extended Version). Technical Report TR2008-617, Dartmouth College, Computer Science, Hanover, NH, April 2008.

[3] Predictive YASIR: High Security with lower latency in legacy SCADA. Rouslan V. Solomakhin, Patrick P. Tsang and Sean W. Smith

[4] Schweitzer Engineering Laboratories Inc., SEL-3021-2 Serial Encrypting Transceiver (www.selinc.com/SEL-3021-2).

[5] Peeyush Jain and Zia Saquib, "Analysis of Different key Distribution Schemes for Distributed Sensor Networks, " LNCS-CCIS, 2011

[6] Secret Key revocation in Sensor Networks, by YoungJae Maeng, Abedelaziz Mohaisen, and DaeHun Nyang, Springer-Verlag Berlin Heidelberg 2007.

[7] http://www.slideshare.net/guest61205606/communication-and-synchronization-in-distributed-systems-2323829

[8] 11711-2010 - IEEE Trial-Use Standard for a Cryptographic Protocol for Cyber Security of Substation Serial Links.

[9] Mohaisen, A., Nyang, D.: Hierarchical grid-based pairwise key pre-distribution scheme for wireless sensor networks. In: EWSN, pp. 83–98 (2006).