# Sensitive Data Protection Using Transient Authentication

**Ms. Khairnar Snehal S.[1]. Prof.Paikrao R.L.[2]**

**[1] A.V.C.O.E. Sangamner,**

**1 A.V.C.O.E. Sangamner.**

*Abstract:* **Wireless devices like laptops are prone to theft and loss due to their small size and the characteristics of their common usage environment, because laptops allow users to work while they are away from their desk. Unfortunately, this is also where the information in these devices is, most at risk. Existing schemes for securing data either do not protect information in the device after it is stolen or require bothersome re-authentication. We provide a secure scheme which protects the sensitive data of the user in these devices. We solve the problem existing systems with Transient Authentication, in which a small hardware token (Mobile Phone) continuously authentication framework to secure sensitive and confidential data on laptop. We implemented this system and the results were outstanding.**

**Keywords: Authentication, Bluetooth, Mobile computing.**

## I. INTRODUCTION

Powerful and affordable laptops have brought users to an unprecedented level of convenience and flexibility. Laptops let users work anywhere, anytime. Unfortunately, physical security is a major problem for these devices. Since they are designed for mobile use, they are often exposed in public places such as airports, coffee houses, and taxis, where they are vulnerable to theft or loss. Along with the value of lost hardware, users are worried about the exposure of sensitive information. People store vast amounts of personal and confidential data on their laptops and the loss of a device may lead to the exposure of bank credentials, passwords, client data, and military secrets.

In presently available schemes the sensitive data in laptops can be protected by using various encryption methods, but the challenge in securing the sensitive and confidential data is not encrypting it but authenticating the current user. The device must obtain the correct evidence of the user's identity and authority before granting access to data. This evidence could be in the form of a password, a smart card inserted into a reader, or biometric data from a fingerprint or iris scanner. But, how often must an authentication should take place by the user? Current systems require users to re-authenticate each time the device performed any operation on sensitive data. This would quickly render the system unusable and many users would disable the authentication system out of annoyance.

Another mechanism would require the user to "unlock" the device once at boot. Thiswould enhance the users experience but leave data vulnerable if the device were lost or stolen. These two models highlight an inherent tension between security and usability.

Transient Authentication resolves this tension. Users can have a small token (Mobile Phone) with modest computational resources. It constantly authenticates the device on behalf of the user. The limited short wireless range serves as a proximity cue, letting a device take steps to protect its data when the user leaves the physical area. We assume that since users have the token which is been frequently used by her, it is far less likely to be misplaced or stolen.

## II TRANSIENT AUTHENTICATION

Transient Authentication is standing on the following four principles:

*A] Access Capabilities to Authorized Users.*

The computer system should carry out the critical operations only when the authorized user is present. Thus, all encryption keys must reside solely on the token, which is in her possession at all times and hence it is far less likely to be stolen or misplaced. The keys must be flushed from the cache of computer system in absence of the user.

*B] No Burdensome involvement of User.*

Users tend to immediately disable inconvenient and cumbersome security mechanisms. But, anecdotal evidence proves that users conveniently accept infrequent insertion of authentication codes. Transient Authentication requires user participation that is convenient. Users will also quickly disable the system with poor performance, thus to ensure complete adoption, the additional overhead of key authentication, communication, and data encryption must not be excessive.

*C] In Users Absence/presence system should secure/restore respectively.*

When the user departs, the device must quickly secure itself so as to avoid the attack, to physically extract any information, by an unauthorized user. Conversely, when a user walks back to use the device, the token should regain wireless contact while she is still some meters away. This gives the system several seconds to restore the device's state thereby avoiding the attackers attempt to extract sensitive data.

*D] Always Ensure Authorized User's Consent.*

The device must not attempt to perform any critical action without the authorized user's consent. Transient Authentication must ensure that only the respective token is capable to carry out the authentication process with the corresponding devices only with her knowledge. To limit the consequences of mobile phone loss, users must authenticate themselves to their token daily.

Armed with these authentication principles, laptop protects data when the user departs by encrypting it. Cryptographic file systems secure data in persistent storage, but the unique characteristics of laptops make protecting data in other memory locations critical as well. Batteries and wireless network links allow devices to continue running while traveling and in public places. This is precisely where they are most vulnerable to loss or theft. Some processes can safely continue while the user is absent, either because they do not handle sensitive data or because they secure their secrets themselves.

## III CONNECTION ESTABLISHMENT

*A] Communication module:*

The communication module consists of a token (mobile phone) and computer system (laptop) which is implemented using User Datagram Packets. Each datagram packet in data field is simply the text inputted. The module establishes a typical single slave Bluetooth Piconet scenario; it opens up a Bluetooth port in both laptop and mobile phone for receiving communications as shown in Fig. 1.1. Once Laptop system receives the packet, it attempts to decrypt that packet using the key currently received from mobile phone and thereby allows the user to access the sensitive data in its original form.

*B] Connection establishment at laptop side:*

The laptop acts as client in the Piconets, the communication is achieved in following sequential manner:

1. Initializing the Bluetooth stack which involves setting the device name, security settings and/or turning the Bluetooth radio on/off.
2. Searching the respective mobile phone that is in proximity.
3. Opening, closing and initiating connections.
4. Perform security Input and Output messages.

These above mentioned steps are carried out by Bluetooth control centre, which typically is a set of control panels that serves as the central authority for local Bluetooth device settings. Before creating the connection the application retrieves local device information, which is used for creating the respective connection. The Bluetooth connection is established using the logical link control and adaptation layer (L2CAP) of the Bluetooth protocol stack. L2CAP does a simple Ns lookup and gets the address of the mobile phone (server) and tries to establish a logical connection with the L2CAP of the server (mobile phone) through the host controller interface (HCI) layer below. After creating the connection the application performs the security I/O messages. This is explained in Fig 1.

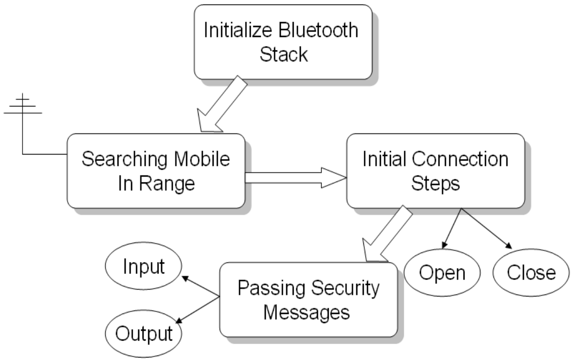
Initialize Bluetooth Stack
Searching Mobile In Range
Initial Connection Steps
Input
Passing Security Messages
Open
Close
Output

Fig 1 Connection Establishment at Laptop Side (Client Side)

*C] Connection establishment at mobile phone side:*

The mobile phone acts as server in the Piconet, it performs the following steps:

1. creates a server connection using the L2CAP
2. Waits for accepting connection and then opens up the connection with the client (laptop)
3. Performs security application I/O messages.

Before creating the connection the application gets, the information about local device and discovers it in the proximity. Meanwhile the client (laptop) establishes the connection to it. When mobile phone receives a L2CAP connection request, it immediately accepts and opens up the connection, then starts performing security I/O messages and accordingly manages the connection.

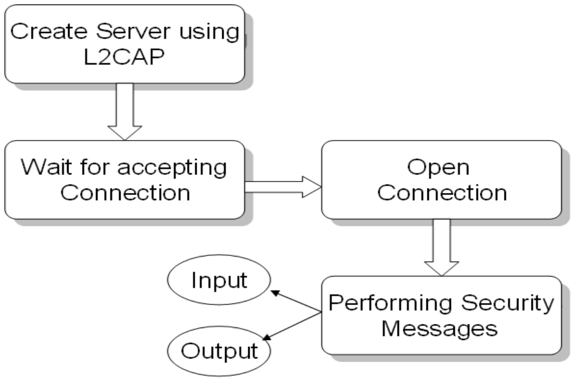
Create Server using L2CAP
Wait for accepting Connection
Open Connection
Input
Performing Security Messages
Output

Fig. 2 Connection Establishment at mobile phone side (Server Side)

*D]Authentication System*

*Initial authentication process:*

In initial authentication process the system performs an operation based on challenge-response messages between the laptop (client) and mobile phone (server) in order to authenticate each other based on immutable Universal Unique Identification system. This system uses UUID which represents a 128-bit value.

*User authentication process:*

As mentioned earlier User authenticates his/her mobile phone infrequently as well as persistently, when the mobile phone requests user for authentication then positive results of this authentication will be valid for a day , if failed to do so user cannot access his/her data for further use. User has to re-authenticate once in 24 hours to access the data as per persistent authentication.

### IV AUTHENTICATION AND ENCRYPTION-DECRYPTION KEY CREATION

Authentication key is used to authenticate the user to the laptop once in 24 hours.

Once the authentication process is complete then the user is requested to select the encryption-decryption (E-D) key to be used for those 24 hours, here the user need not perform a burdensome job of remembering the E-D key. If the process is completed successfully then the encryption-decryption process commences to perform the operation of encrypting the data in absence of the user and decrypting the data in presence of the user, using the same E-D key.

### V DISCONNECTION AND RECONNECTION

The laptop system periodically sends nonce to mobile phone which ensures the laptop system whether the authorized user is present or absent in the proximity. If the user is present then the sensitive data will be accessible. But if the user is absent then the system will secure itself immediately. But what if the short wireless link between the two devices drops the packet? In that case laptop will secure itself if the response is not received in expected round trip time. Since this is a single, uncontested network hop, this time is relatively stable. Then the Laptop system retries sending a request, if the response is achieved then data will be accessible otherwise it remains in secured state.Laptop checking for mobile phone presence.

### VI ENCRYPTION AND DECRYPTION PROCESS

In our system which we have implemented, we have used the Data Encryption Standard for the process of Encryption and Decryption. The reason for using this method is that since we have implemented our model using Java Technology, where the encryption and decryption function by default uses DES for encrypting and decrypting the data and also it is fast enough to run efficiently with limited memory resources and processing time. Mobile Phone sends the E-D key to for decrypting the data and Laptop uses this E-D key to decrypt the encrypted sensitive data.

### VII OVERALL AUTHENTICATION PROCESS

The over all processes of authentication system illustrated in Fig.

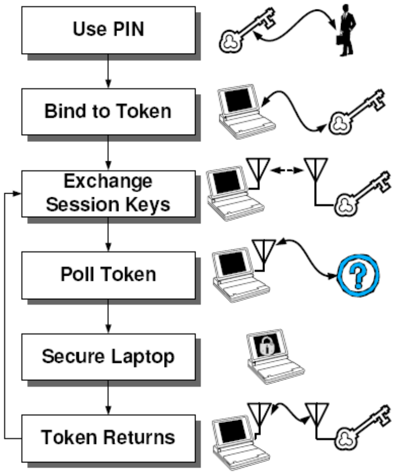

Fig.3 Authentication System

The steps followed in overall Authentication Process.

### VIII DATA FORTRESS IMPLEMENTATION (*ALGORITHMS)*

As we have mentioned earlier in that we have implemented the transient system using mobile phone and laptop. The Data fortress system uses mobile phone as token and Laptop as wireless device containing sensitive information. The languages used were J2SE and J2ME. The implementation algorithms developed by us are given below:
The vital Algorithms developed by us for implementing the Data Fortress applications are as follows:

*A]Algorithm for Connection*

Establishment between Laptop and Mobile phone:
BluetoothServiceDiscovery class is used to establish connection.

1. Creating the object of DiscoveryAgent.
2. A StartInquiry method of DiscoveryAgent object is called for searching the device in the proximity.
3. After completion of the Search for devices, a lock is applied until the user selects the respective device for the purpose of binding. Lock is applied for halting the processes.
4. An application service is searched on the selected device from step (C) using a SearchService method.
5. After application service is searched on the token device then L2CAP connection is established using Connector.open(connectionURL) where connectionURL is a string.
6. After connection is established Nounces are send/receive to validate the
7. presence of the token in the proximity.

B] Algorithm for monitoring and maintaining the connection

LaptopThread class is used for checking the status of connection and accordingly performing encryption and decryption.

1. The lock is applied. This lock helps in keeping the track of the connection.
2. This lock is notified when disconnection/reconnection occurs.
3. The status of the connection is checked and respective action is taken, that is,

If disconnection occurs :

1. Encrypt the sensitive data
2. Stop the accessibility of the access rights of these files

If Reconnection occurs:

1. Allow the access rights to be accessible
2. Decrypt the data
3. Go to step A.

C] Algorithm for Locking the Access Rights of sensitive files

Locker class was developed to block the access rights for the purpose of unauthorized access. To achieve this we developed two functions.

Lock() function is called when disconnection occurs.

Store the path of the files.

1. File channel uses pointer to access the Read-Write rights of the files.
2. Lock is applied.

Unlock() function is called when reconnection occurs.

1. Release the lock applied in Lock().
2. Close the file channel to allow the user to access the file.

D] Algorithm for protecting the sensitive files.

Protection, encryptfile and decryptfile classes are used to provide the protection to the sensitive files. For encryption and decryption Advanced Encryption Standard Algorithm is used.

For encrypting the file,

1. Create the object of Encryptor class and the file.
2. Store the file in the FileInputStream, which reads the input file in bytes.
3. The E-D key and data is passed to the encrypt() of the Encryptor class.
4. Obtain the encrypted file by using the FileOutputStream.

For decrypting the file,

1. Create the object of Encryptor class and the file.
2. Store the encrypted file in the FileInputStream, which reads the input file in bytes.
3. The E-D key and encrypted data is passed to the decrypt() of the Encryptor class.
4. Obtain the decrypted file by using the FileOutputStream.
5.

## XI FUTURE WORK

The threat of losing the token is very serious. Several future research topics could mitigate this problem. The first is to require biometric feedback to keep the token functioning.

Although a fingerprint may work for an initial authentication, continuous monitoring of heartbeat or body temperature could detect if the user removes the token. If the pulse or heat source is lost, the token must be revived with a password.

If the mobile system can provide a small amount of physically secure hardware, there are simpler ways to construct this system. For instance, this hardware would store the key-encrypting keys, which can only be unlocked by a wireless token.

This token need only supply the correct pseudo-random sequence—similar to a SecureID. This requires trusting the physical security of the hardware, and in exchange the token can be constructed as a one way transmitter with significantly less computational power. Such a system still requires the use of mechanisms to deal with lost authentication, binding, and key-management.

## X RESULTS

System declares user absent after three tries to connect to mobile phone without response. Figures below show the time required to Encrypt and Decrypt data by using Encryption and Decryption algorithms like AES, Blowfish and RSA. By analyzing time we can conclude that AES required less time for encryption and decryption process.

**System security:** There are two concepts in system security:
1. User's mobile phone cannot provide authentication services to other user's laptops.

2. Mobile phone cannot send authentication messages over wireless link in clear text form

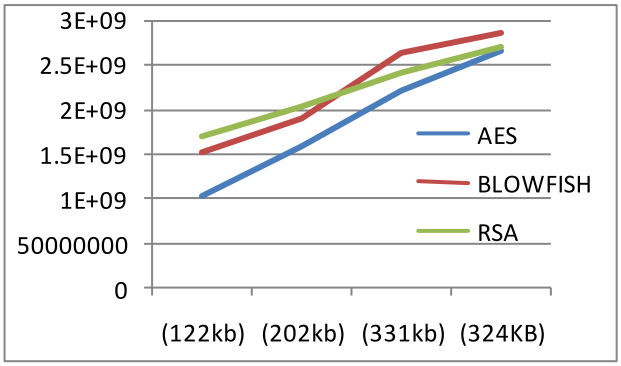

Fig. 4 Time required for Encryption

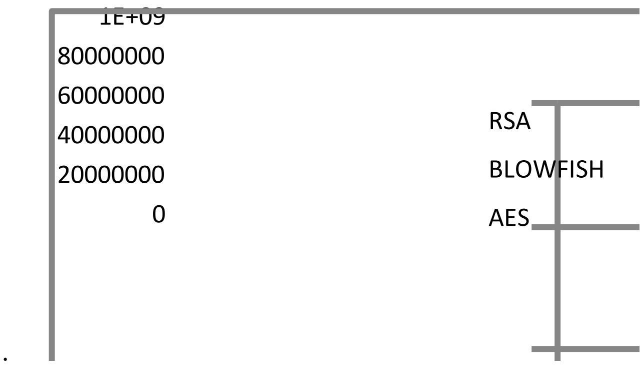

Fig.5 Time required for Decryption

## CONCLUSION

Now-a-days, information in wireless devices is indispensable for the users of the respective devices. This information may be present in laptops, desktop computers etc. which is vulnerable to theft. We provide a secure system which protects the sensitive data of the user in these devices.

In our system, we are using cell phone as a token which will authenticate the system and the client machine, which can be a laptop or a desktop computer. These two systems are connected to each other via a Bluetooth.

Once, the devices are authenticated and connected then our application will ask the user to declare the sensitive files and folders on laptop or desktop machine.

When the user along with his cell phone is in the range of the laptop or desktop computer, the sensitive data will be available for access and as soon as the user is outside the range then the data will be inaccessible to others.

Hence, our application provides security to the sensitive data in the laptop or desktop machine. The advantage of our application is that the user doesn't have to authenticate him/her time and again to the system, as authenticity is taken care by our application. We have developed the pioneer version of Data Fortress application we strictly feel that this application has various areas in which it can further be extended. We the developers conclude that our security makes system more efficient and also assures high level of reliability to the users of Data Fortress.

## REFERENCES

[1] Rania Abdelhameed, Sabira Khatun, Abdul Ramlise, "Application of Cell-Phonein Laptop Security" Journal of Applied Sciences 5 (2) : pp.215-219 Feb 2005.

[2] Brian D. Noble, Mark D. Corner "The Case For Transient Authentication".

[3] Mark D. Corner, "Transient Authentication For Mobile Devices" 2003.

[4] Anthony Nicholson, Mark D. Corner, Brian D. Noble "Mobile Device Security Using Transient Authentication, "IEEE Transaction on Mobile Computing, vol. 5,
No. 11, pp. 1489-1501, Nov. 2006.

[5]C. E. Landwehr. "Protecting unattended computers without software.", In Proceedings of the 13th Annual Computer Security Applications Conference, Pp 274–283, December 1997.

[6] S. Brostoff and M. A. Sasse. "Are passfaces more usable than passwords? a field Trial Investigation" In Proceedings of HCI 2000, pp 405–424, Sunderland, UK, 2000.

[7] D. Davis. "Compliance defects in public-key cryptography". In Proceedings of the 6th Usenix Security Symposium, pp 171–178, San Jose, CA, 1996.

[8] A. Adams and M. A. Sasse. "Users are not the enemy: Why users compromise security mechanisms and how to take remedial measures." Communications of the
ACM, 42(12): pp 40–46, December 1999.

[9] A. Whitten and J.D. Tygar. "Why johnny can't encrypt: A usability evaluation ofnPGP 5.0", In Proceedings of the 8th USENIX Security Symposium, Washington D.C., August 1999

[10] R. Morris and K. Thompson. "Password security: A case history. Communications" ACM, 22(11):pp 594–597, November 1979.

[11] D. V. Klein. "Foiling the cracker: A survey of, and improvements to, password security. ", In Proceedings of the USENIX Security Workshop, Summer 1990.

[12] U. Manber. "A simple scheme to make passwords based on one-way functions muchharder to crack. Computers and Security, 15(2):171–176, 1996.

[13] L. Lamport."Password authentication with insecure communication. Communications " ACM, 24(11):pp 770–72, November 1981.

[14] F. Monrose, M. K. Reiter, Q. Li, and S. Wetzel." Cryptographic key generationnfrom voice. " In Proceedings of the IEEE Conference on Security and Privacy, Oakland, CA, May 2001.

[15] C. E. Landwehr. " Protecting unattended computers without software." In Proceedings of the 13th Annual Computer Security Applications Conference,
pp 274–283, San Diego, CA, December 19