International Journal of Systems and Software Security and Protection
Volume 14 • Issue 1

A Light Weight Temper Resistance Client File in an External Memory for Remote User Authentication and Access Control

Bello Alhaji Buhari, Usmanu Danfodiyo University, Sokoto, Nigeria*
https://orcid.org/0000-0001-7333-1386

Afolayan Ayodedele Obiniyi, Ahmadu Bello University, Zaria, Nigeria

Sahalu B. Junaidu, Ahmadu Bello University, Zaria, Nigeria

Armand F. Donfack Kana, Ahmadu Bello University, Zaria, Nigeria

ABSTRACT

This research proposes a lightweight tamper resistant client file in an external memory as an alternative to smart card for remote user authentication and access control. The benefit of using this special client file is portability and ease of acquirement, especially in school online portals, online resources portals, and e-commerce portals. The characteristics and design considerations that make smart card tamper resistant are reviewed. Techniques and characteristics to make a client file in an external memory to exhibit a lightweight tamper resistant property has been formulated. The Kumari et al.'s scheme, which is the latest research that uses external memory for remote user authentication, has been reviewed. The basic system design and software design of the proposed client file is presented and modeled. This will enable implementation of the proposed system using any prepared programming or scripting language of one's choice. The proposed scheme and reviewed scheme are also evaluated for efficiency, tamper resistance, and impersonation attack.

KEYWORDS

Access Control, Authentication, Client File, External Memory, Remote User Authentication, Smart Card, Tamper Resistance

1. INTRODUCTION

Due to increased demand of security and fast development in communication, networking, computer software, web and mobility, there are enormous demand in better user authentication and personalization techniques. Many of the authentication systems are not very reliable specifically in ad-hoc network where two or more devices or nodes or terminals with wireless communications and networking capability communicate with each other without the aid of any centralized administrator.

DOI: 10.4018/IJSSSP.318342 *Corresponding Author

They can be broken, stolen or forgotten. Similarly, the attackers can control access to secured locations for passwords. For these problems, one of the most effective ways is using biometrics to avoid password to be stolen or forgotten (Srivastava *et al.*, 2013).

Smart cards offer consistent authentication by executing challenge-response protocols without revealing implanted secret keys. This tamper resistant device is used in many applications where control over the execution of an algorithm or secrecy protection is required (Boneh *et al.*, 1999). These are also physical security boundaries implemented to restrict the scope of physical attacks (Ferres *et al.*, 2018). Further, reasons for its first choice are small physical size, the portability, the ease of nonvolatile memory, and the security guaranteed by a single chip computer embedded in a plastic card (Buhari *et al.*, 2022).

But usage of smart card authentication scheme in remote environment may be difficult to users due to cost of acquiring and implementing smart card facilities. That is, installation of the necessary infrastructure for smart cards, together with the technique of uploading diverse secure access modules (SAMs) into card readers.

These problems motivate the use of external memory instead of smart card (Rhee *et al.*, 2009; Chen *et al.*, 2012; Jiang *et al.*, 2013; He *et al.*, 2013; Kumari *et al.*, 2014). But the problem of non-tamper resistance property associated with external memory limited researches in that direction (Buhari *et al.*, 2022).

As such, client file that exhibit light weight tamper resistance property stored in external memory can be used. The advantage of using this special client file is portability and ease of acquirement especially in schools online portals, online resources portal and e-commerce portals. A technique to make a client file in an external memory to exhibit a light weight tamper resistance property is proposed. The characteristics and design considerations that make smart card tamper resistance is reviewed. Characteristics or features that will make a client file to exhibit light weight tamper resistance property are formulated. The basic system design and software design of the proposed client file is presented and modeled. This will enable implementation of the proposed system using any prepared programming or scripting language of one's choice. The Kumari *et al.*, (2014) scheme which is the latest research that uses external memory for remote user authentication has been reviewed. The proposed scheme and reviewed scheme are also evaluated for efficiency, tamper resistance and impersonation attack.

The contributions of this research are as follows:

1. The techniques that will enable client file residing in an external memory to exhibit light weight tamper resistance property are formulated. Followed strictly this feature during the design and implementation of the client file system will make it to be light weight tamperproof.
2. The design and modeling of the proposed client file system is presented. This will enable its implementation and deployment using any programming language or scripting language of one's choice.

The rest of this research is presented as follows: section 2 is literature review, section 3 is review of Kumari *et al.*, (2014) scheme, section 4 is smart design review, section 5 is proposed light weight tamper resistance client file design presentation, section 6 is evaluation of the proposed system and section 7 is conclusion.

2. RELATED WORKS

Different researches have been conducted on tamper resistance. These include Chong *et al.* (2004) that proposed a license protection technique based on a tamper-resistant hardware token and a key tree. The key tree ensures flexibility and the hardware token ensures tamper resistance. They use

their license protection scheme to LicenseScript licenses. They analyze the protection technique in terms of security with reference to some common security assumptions. They also make a formal protocol verification using CoProVe.

More so, Kursawe *et al.* (2009) characterized a new primitive, the reconfigurable PUF (rPUF) which is a PUF with an instrument to change it into a new PUF with a new unpredictable and out of control challenge-response manners. This works even if the challenge response manner of the unusual PUF is already known. They present two practical instantiations of a reconfigurable PUF first is a new alternative of the optical PUF, and the other is based on stage change memory. They also demonstrate how an rPUF can be applied to defend non-volatile storage against invasive physical attacks.

A practical and secure user authentication system that can enable the usage of a external memory and maintains all the benefits of smart card-based systems was proposed by Rhee *et al.* in 2009. Its security is based on the discrete logarithm problem with Diffie-Hellman keys, hashing, and time stamps. Even when a user uses an unsafe device, it is secure against off-line dictionary attacks, user and server impersonation attempts, and other threats. Tan (2009) provides a security analysis of the Fan *et al.*, (2005) and Rhee *et al.*, (2009) password authentication systems. They discovered that Rhee et al strategy is susceptible to middle man and impersonation assaults. Therefore, a hacker may log in and access the remote server by pretending to be a genuine user.

In addition, Akram *et al.* (2011) analyse the justification for a general-purpose cross-platform user centric tamper-resistant device based on the smart card architecture, its applications in different computing environments, along with the ownership management framework. They kept the design as generic so it can easily be integrated with the existing architecture of dissimilar computing platforms.

A secure password-based remote user authentication method without smart cards was developed by Chen *et al.* (2012). It addresses the issue of user impersonation attacks by incorporating a blind factor into the authentication data saved on a user's local memory device. Based on the computational Diffie-Hellman problem, blind factor, hash function, and time-stamp, the system is secure. Mutual authentication is guaranteed by their suggested system, which also fends off offline dictionary, replay, forgery, and impersonation threats. It keeps every benefit from the Rhee *et al.*, 2009 strategy. Compared to earlier approaches, the total message length is less and the computational cost is cheaper.

The Chen *et al.* (2012) system is also subject to cryptanalysis by He *et al.* (2013), who discovered that it is susceptible to insider privilege attacks and attacks on stolen devices. Additionally, it does not provide key control or perfect forward secrecy. They therefore suggested a better plan to address these issues and keep the advantages of the original plan. However, Chen et al(2012) approach still performs better than theirs. Their system's security is based on the discrete logarithm problem and hash function proposed by Diffie-Hellman.

An enhanced password-based remote user authentication system without the need of smart cards was suggested by Jiang *et al.* (2013) after analyzing Chen *et al.* (2012)'s scheme. They noted that Chen et al strategy is vulnerable to dictionary assaults conducted off-line. The hash function and computational Diffie-Hellman problem serve as the foundation for the scheme's security. They showed that their method achieves mutual authentication between the user and the server and can withstand a variety of assaults. Both in terms of computing and communication costs, it is more effective.

Jiang *et al.* (2013) and He *et al.* (2013) systems ignore a user's privacy, according to Kumari *et al.* (2014). They also noted that Jiang et al(2013) .'s approach lacks forward secrecy and is susceptible to insider assaults and denial of service attacks. Additionally, they discovered that the password-changing feature in He et al(2013) system is similar to registering, however it is inappropriate in Jiang et al(2013) .'s approach. Once more, neither scheme's login phase is able to stop users from entering the wrong password, which results in the calculation of an invalid login request. To address the found shortcomings, they therefore develop a new system that ensures user anonymity. Additionally, they provided a formal demonstration of the suggested scheme's security based on the reasoning put out by Burrows, Abadi, and Needham (BAN logic). It inherits a free password changing feature from

Jiang et al schemes, resistance to insider attack and denial of service attack from He et al (2013). Additionally, it safeguards user identities by granting anonymity to users.

Also, Khan and Sakamura (2015) presented an eTRON architecture setup with functions for mutual authentication, encrypted communication and access control that has the tamper-resistant eTRON chip. In addition to the security, the eTRON architecture also provides a wide variety of functionalities through a logical set of application programming interfaces (API) leveraging tamper-resistance. They also talk about various features of the eTRON architecture, and present two representative eTRON-based applications in order to evaluate its efficiency by comparing it with other existing applications.

Small scale defenses against power analysis attacks for a lightweight block cipher was proposed by Shibagaki *et al.* (2018). The key element of the suggested method's countermeasure is noise. In particular, the suggested method operates a random number generator to produce power consumption as a noise component (RNG). The noise then cancels the correlation, increasing the tamper resistance against power analysis attacks. A RNG is also unsuitable for cryptographic hardware because it is needed for the seed creation of secret keys, among other things. Noise component is generated by operating a RNG which is power consumption and tamper resistance against power analysis attacks is improved.

Furthermore, tamper-resistant technology based on blockchain for data in online and offline environments has been proposed by Kim *et al.* (2021). The suggested algorithm projected a new data recording instrument that operates in low-level hardware of digital tachographs for tamper-resistance in light blockchains and on/offline situations. With the exception of a random hash, the proposed light blockchain follows the same design as the current blockchain. The data of all blocks must be recalculated in order to determine the hash value of the current block if the data of the already-formed block is altered using the hash value of a prior block. This procedure makes tampering nearly impossible and requires authentication such as Proof of Work or Proof of Stake. The projected light blockchain algorithm took an average time of 1.85 ms/Mb for encoding and 1.65 ms/Mb for decoding. The statistical result shows that the average execution time was anticipated a performance index of light blockchain software. The estimation error in the execution time results in file units twisted out to be about three times greater.

Lastly, Lu *et al.* (2022) that offer a physical security system that can defend data from unauthorized access when the computer chassis is opened or tampered with. They used sensor switches to monitor the chassis status at all times and upload event logs to a cloud server for remote monitoring. Six modules are used in the development of this system: SPS IDSWeb, PSPS IDSServer, PSPS IntrusionManager, PSPS Defense, PSPS Synchronize, and PSPS RecoveryManage. They used three programming languages namely Visual C++ 2019, Java 1.8.0 and HTML, CSS, JavaScript. They performed two test cases to validate the system operation and show how to monitor the system state with PSPS IDSWeb. Also, they present a comparison with two well-known IDS systems: HIDSOSSEC and NIDS-SNORT.

3. REVIEW OF KUMARI ET AL. (2014) SCHEME

Kumari *et al.* (2014) is the latest research using external memory that propose a more secure scheme with a user anonymity feature. Three phases namely: registration, login and authentication will be discuss in details in the next sub-sections of this part.

3.1 Registration Phase

For any person to become a valid user he/she has to get registered at S through this phase. The steps to be followed by user U_i and S for registration are as follows:

1. U_i Chooses his identity Id_i in the specific format, password Pw_i and a random number $r_i \in Z_q^*$. Then submit $\left\{Id_i, h\left(Pw_i, r_i\right)\right\}$ to S via secure channel.
2. On receiving the information S computes:

$$Y_i = h(Id_i \,||\, x) \oplus h\left(Pw_i, r_i\right) \tag{1}$$

$$W_i = h(h\left(Pw_i, r_i\right) || \, Id_i) \tag{2}$$

Encrypt $EId_i = E_x\left(Id_i, T_{Ri}\right)$ (3)

where:

- T_{Ri} is the registration timestamp;
- S provides the information $\left\{Y_i, EId_i, W_i, h\left(\bullet\right), p, q, g\right\}$ to U_i through the secure channel.

Upon receiving the information, U_i computes:

$$Z_i = \left(Id_i \otimes Pw_i\right) \oplus r_i \tag{4}$$

and saves it along with his receiving information.

Now, U_i's USB contains $\left\{Y_i, EId_i, W_i, h\left(\bullet\right), Z_i, p, q, g\right\}$.

This can be shown in Fig. 1.

3.2 Login Phase

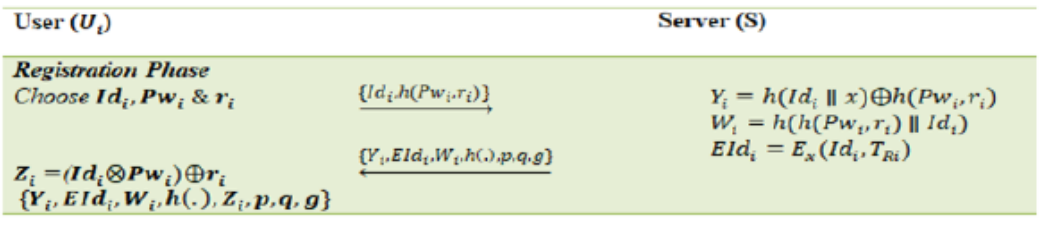
To login U_i obtains the stored information from his USB stick and computes the required values to compile the login request as follows:

Retrieves $r_i = Z_i \oplus \left(Id_i \oplus Pw_i\right)$ (5)

and computes:

$$W_i^* = h(h\left(Pw_i, r_i\right) || \, Id_i) \tag{6}$$

Figure 1. Registration Phase (Kumari *et al.*, 2014)



Compares W_i^* with W_i. If $W_i^* = W_i$ he proceeds further; otherwise he discard the session U_i then:

Computes $Y_i^{'} = Y_i \oplus h\left(Pw_i, r_i\right)$ (7)

Choose $\alpha \in Z_q^*$.

Acquires timestamp T_i and computes:

$$C_i = g^{\alpha} \bmod p \tag{8}$$

$$V_i = h\left(Id_i, Y_i^{'}, C_i, T_i\right) \tag{9}$$

Sends the login request $\left\{EId_i, C_i, V_i, T_i\right\}$ to S via public channel.

This can be shown in Fig. 2.

3.3 Authentication Phase

S and U_i perform the following steps to authenticate each other:

1. On receiving $\left\{EId_i, C_i, V_i, T_i\right\}$ S:
   a. Acquires T_s to verify whether $\left(T_s - T_i\right) \leq \Delta T$. If so:
   b. It decrypts EId_i to obtain a user's identity:

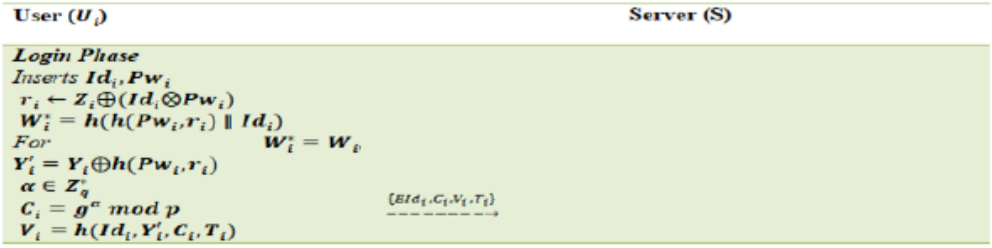
$$\left(Id_i, T_{Ri}\right) = D_x\left(E_x\left(Id_i, T_{Ri}\right)\right) \tag{10}$$

   c. Checks the format of Id_i, if Id_i is valid and timestamp is fresh, then goes to step b; otherwise dump the login request and ends the session.
2. Computes:

$$Y_i^{''} = h(Id_i \,||\, x) \tag{11}$$

$$V_i^* = h\left(Id_i, Y_i^{''}, C_i, T_i\right) \tag{12}$$

Figure 2. Login Phase (Kumari *et al.*, 2014)



Verifies if V_i^* and V_i are equal, if they are equal it goes to step c; otherwise dump the login request and ends the session.

3. Generates a random number $\beta \in Z_q^*$ and computes $E_i = g^{\beta} \bmod p$. Then, it acquires another current timestamp T_{ss} and computes:

$$V_s = h\left(Id_i, Y_i^{''}, T_{ss}\right) \tag{13}$$

$$EId_i^{'} = E_x\left(Id_i, T_{ss}\right) \tag{14}$$

$$W_s = (EId_i^{'} \,||\, E_i) \oplus h\left(Y_i^{''}\right) \tag{15}$$

Then it sends $\left\{W_s, V_s, T_{ss}\right\}$ to U_i through public channel.

4. After receiving the message form S, the user checks T_{ss} for freshness. For fresh T_{ss} the user proceed further:

Computes $(EId_i^{''} \,||\, E_i) = W_s \oplus h\left(Y_i^{''}\right)$ (16)

a. Compare $EId_i^{''}$ and previously stored EId_i. If both are different he proceeds; otherwise he discards the session:

Computes $V_s^* = h\left(Id_i, Y_i^{''}, T_{ss}\right)$ (17)

b. Verifies whether the computed V_s^* and received V_s are equal. If they are equal mutual authentication finished successfully; otherwise, S is not authenticated, and U_i disrupts the session.
c. Then U_i replaced EId_i in his USB sticks with $EId_i^{''}$
d. Once U_i and S authenticate each other, they compute the session key:

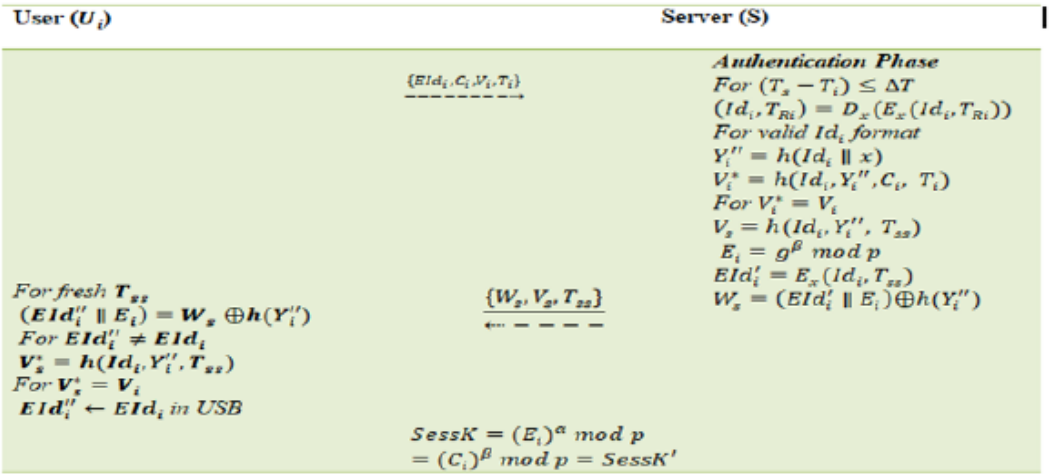
$$SessK = (E_i)^{\alpha} \bmod p \tag{18}$$

and:

$$SessK' = (C_i)^{\beta} \bmod p \tag{19}$$

This can be shown in Fig. 3.

Figure 3. Authentication Phase (Kumari *et al.*, 2014)

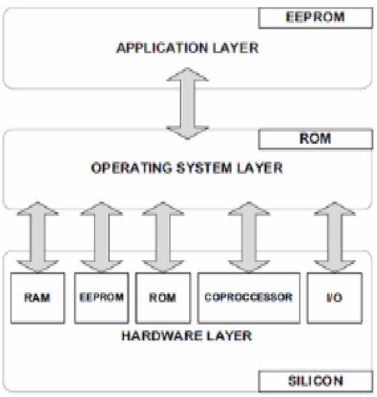


4. SMART CARD DESIGN

A Smart card has a unique identifier, partake in an automated electronic transaction, used primarily to include security, not easily forged or copied, store data securely and host/run a variety of security algorithms and functions (Mayes and Markantonakis, 2017). It is a defense token that contains an implanted chip and has encoded information within the microchip (Sharma and Dixit, 2018). It is a tiny personal computer without a screen and keyboard (Adavalli, 2017). The microchip on the smart card can either be a microcontroller or an implanted memory chip. They can be prepared out of metal or plastic. The design descriptions of a normal smart card are an 8-bit or 16-bit CPU, 16 to 64 Kbytes of ROM, 4 to 64 Kbytes of EEPROM and 256 bytes to 1Kbytes of RAM (Hassler, 2002). The data storage on the smart card is managed by RAM, EEPROM and ROM because of these limited resources offered (Adavalli, 2017). The smart card basic system can be shown in fig. 4. They communicate only via a reliable terminal like ATM, EFTPOS or any PC connected with a card reader.

There are two categories of cards namely: memory card and intelligent card (Adavalli, 2017). A memory card can store data but cannot process it and this card can be modified or duplicated easily while an intelligent card has microprocessor to execute instructions on the data available in its memory

Figure 4. Smart Card basic system (Selimis *et al.*, 2009)



resources. There are five parties involved in the life cycle. Semiconductor manufacturers are responsible for chip design and mass production. Smart card manufacturers implant issuers' requirements. Card issuers usually have more business/behavioral considerations while deploying and managing smart card-based solutions. Service providers design and implement value-added services and Users gain from those services (Deville et L., 2003) as shown in fig. 5.

Normally a smart card is prepared from three elements. The plastic card is the most basic one which is of 85.60 mm × 53.98 mm × 0.80 mm dimension but may have the smaller size of a GSM subscriber identification module recognized as SIM. A printed circuit and an integrated circuit (IC) are implanted on the card (Taponen, 2000).

Smart card reader is used for reading or writing and sending or receiving information to and from smart card. This includes electrical contacts that allow the card to communicate with other devices, and a microcontroller with a RAM memory to execute the application program stored in smart card (Martínez-Peláez *et al.*, 2008). It consists of one microcontroller to execute the application program and identify or react against physical attack, a secure coprocessor to identify or react against physical attacks and perform cryptographic operations such as encryption or decryption of sensitive data, a graphical LCD to display messages, a key pad to enter sensitive data such as PIN number, X-ray sensor to identify the exposure of radiation in order to stop imprinting of the RAM memory, tamperature sensor to identify the extreme variation of tamperature (lower than -20ºC) in order also to stop imprinting the RAM memory, barrier substrate which is the first line of protection generating and sending a signal when it is compromised and Lithium battery that allows to store keys in the RAM memory (Martínez-Peláez *et al.*, 2008). This can be shown in fig. 6.

To operate the smart card, the reader needs to implement the following four functionalities (Lassus, 1997):

1. Power on/off the smart card
2. Reset the smart card
3. Read data from the smart card
4. Write data to the smart card

Figure 5. Smart Card Life Cycles

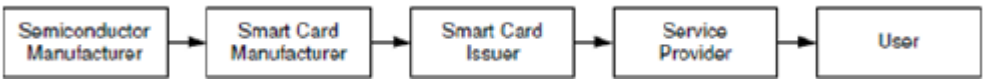
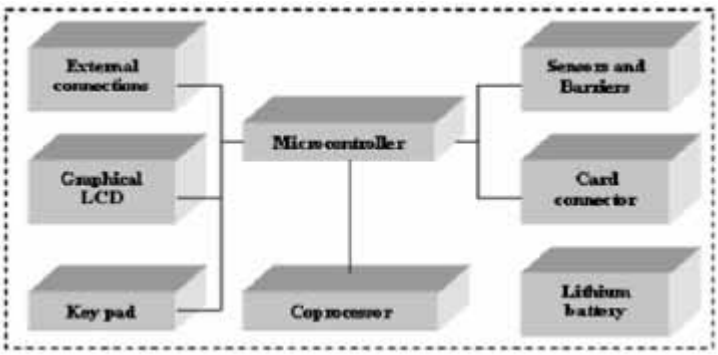


Figure 6. Block diagram of a smart card reader (Martínez-Peláez *et al.*, 2008)



The physical size of a smart card is described in ISO 7810. The dimensions of a smart card are 85.6 mm by 53.98 mm, with a corner radius of 3.18mm and a thickness of 0.76mm. Smartcard chip placement was defined in ISO 7816-2, which was developed in 1988 (Selimis *et al.*, 2009).

True open Smart cards will have the following characteristics (Mohammed *et al.*, 2004):

1. They will run a non-proprietary operating system commonly implemented and supported.
2. No single vendor will specify the standards for the operating system and the card's use.
3. The cards will support a high-level application programming language (e.g., Java, C++) so issuers can supply and support their own applications as well as applications from many other vendors.
4. Applications can be written and will operate on different vendor's multi-application smart cards with the same API (Application Programming Interface).

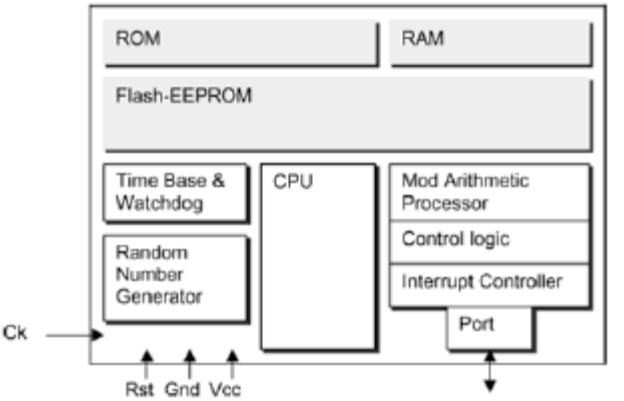
Four applications of smart card arise in examining its potential IT uses as follows (Jurgensen and Guthery, 2002):

1. ID Badge
2. Token for building and office door access
3. Token for computer and network access
4. Token for cash financial transactions

4.1 Smart Card Microcontroller

The microcontroller employed in Smart card applications consists of a central processing unit (CPU) of 8, 16 and 32 bits words and blocks of memory (Bolchini *et al.*, 2003). These include RAM of 256 bytes to 1 kilobyte, ROM of up to 32 kilobytes, and reprogrammable nonvolatile memory (NVM) of 256 bytes to 64 kilobytes. RAM functions to store executing programs and data temporarily, while ROM functions to store the operating system, fixed data, standard routines, and lookup tables. The reprogrammable nonvolatile memory functions to store information that has to be preserved when power is switch off. They must also be alterable to hold data specific to individual cards or any changes feasible above their lifetime. It has minimum of 100,000 write/erase cycles. This is shown in fig. 7.

Figure 7. A schematic representation of a smart card microcontroller (Bolchini *et al.*, 2003)



4.2 Smart Card Coprocessor

The CPUs used in smart cards are not very fast but very reliable. Some of the requirements specific to smart cards cannot be fully fulfilled using software running on the CPU. Therefore, the need for supplementary hardware to meets these demands (Leng, 2009). Since security is the spotlight of this research, we will express the coprocessors directly associated to the general goal of security.

4.2.1 Coprocessors for Cryptographic Algorithms

It has been a long time that the smart card has implemented the coprocessors to calculate DES (Data Encryption Standard) being employed first as the standard cryptographic algorithm for financial systems and telecommunications applications. It has been replaced by AES (Advanced Encryption Standard) since it is developed with the considerations on the smart card implementation (Leng, 2009). In majority of applications, smart card needs to store the certificates and generate/verify signatures. This consists of the calculations in the sphere of public-key algorithms, such as RSA and elliptic-curve algorithms. To assist these algorithms, there are personally developed arithmetic units on the silicon, which are personally design to achieve several basic calculations that are essential for these types of algorithms. These are exponentiation and modulo computations using large numbers (Leng, 2009).

4.2.2 Random Number Generator

Random numbers are regularly needed in smart cards for keys generation and authentication protocols with which smart cards and terminals authenticate each other's uniqueness. The random number generated must be real random numbers rather than pseudo–random numbers universally created by software-based random-number generators (Leng, 2009). Different approach is applied because it s very difficult to implement in silicon. The random-number generator takes a variety of logic states from the microcontroller, like the clock signal and the contents of the memory, and uses them to a linear feedback shift register (LFSR) clocked by a signal that is also generated using numerous dissimilar parameters.

4.3 Smart Card Software

There are basically two types of smart card software: Host software is also referred to as reader-side software and card software referred to as card-side software (Guo, 2002). Majority of smart card software is host software designed for personal computers and workstation servers which accesses existing smart cards and integrate these cards into larger systems. It usually include end-user application software, system-level software that supports the attachment of smart card readers to the host platform, and system-level software that supports the deployment of the particular smart cards needed to support the end-user application. They also comprise application and utility software essential to support the administration of the smart card infrastructure. It is typically designed using one of the high-level programming languages found on personal computers and workstations like C, C++, Java, BASIC, COBOL, Pascal, or FORTRAN and associated with commercially existing libraries and device drivers to access smart card readers and smart cards mounted into them.

Card software on the other hand is also frequently classified as operating system, utility, and application software as with host software. It is normally used to modify an existing smart card for a particular application and involve moving some functionality from host application software onto the card itself. It is designed using a low-level machine language for a particular smart card chip and is used to expand or restore basic functions on the smart card.

The operating system allows the microprocessor to manage and control card memory. One of the main tasks of operating system is to offer standard way to transmit data between the card, card reader, and/or applications. It is also accountable for access control, authentication, and information security. Multi-Functional Card (MFC) operating system is one of the first smart card operating systems that introduced by IBM in 1990. Others are: CardOS, STARCOS, JCOP, TCOS, Cyberflex, and Payflex.

4.4 Smart Card Authentication Process

According to Jurgensen and Guthery (2002) there are three entities involved in smart card authentication. These include: the cardholder, the smart card token and PC system. Authenticating these entities require three separate actions. The cardholder must authenticate himself to the smart card token. This secure against the lost of token which will enables some unknown person to be able to impersonate the legitimate cardholder. Once the smart card is convinced about the identity of cardholder it then authenticates its identity or identity of its cardholder to the PC system. And, the PC system now authenticates itself to the smart card token. Now the transaction can proceed with each party confident in dealing with legitimate identity.

Since the smart card reader and smart card cannot store a certificate revocation list (CRL), the smart card have to verify the status of the certificates online by using the online certificate status protocol (OCSP) responder (Martínez-Peláez *et al.*, 2008) as shown in fig. 8. Fig. 9 shows the common action to be performed when a physical attack is detected in a tamper resistance device.

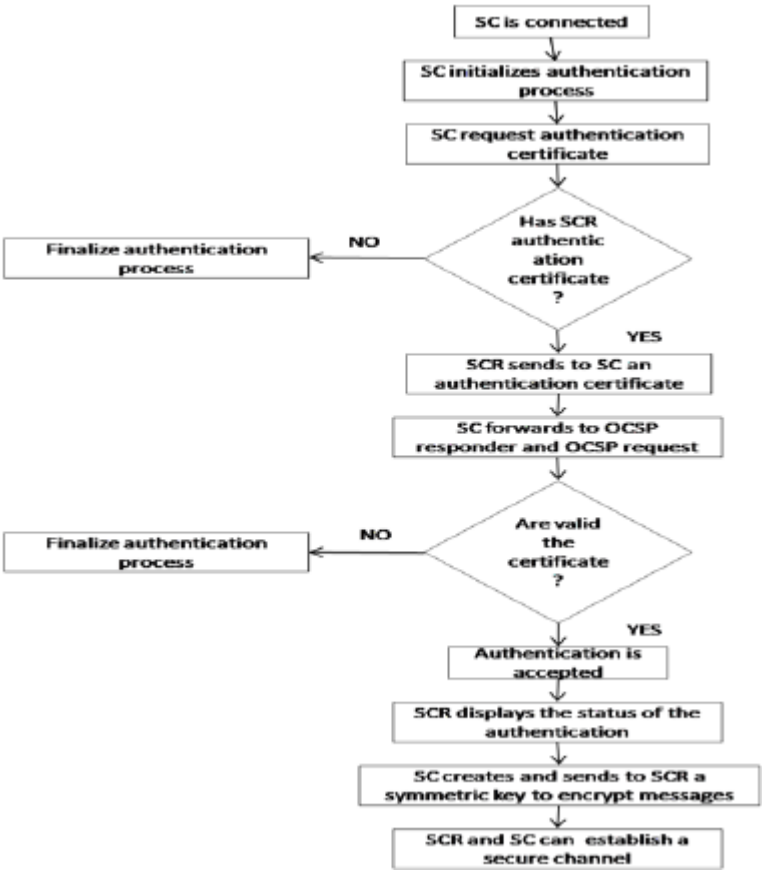
5. PROPOSED LIGHT WEIGHT TAMPER RESISTANCE CLIENT FILE IN AN EXTERNAL MEMORY

A file is a contiguous logical address space, mapped by the operating system onto physical devices. It is a named location on an external memory to store related information. There are two types of files namely: text and binary file. A text file is a term used to describe a file that is consists a sequence of character codes in human readable form. While binary file content is in a binary format consisting of a series of sequential bytes not readable by humans.

Special client file store in an external memory can be used for remote user authentication due to cost of infrastructure requirements and mobility of smart card by web users. The advantage of using this special client file is portability and ease of acquirement especially in schools' online portals, online resources portal and e-commerce portals (Buhari *et al.*, 2022). Techniques to make a client file on external memory to exhibit a light weight tamper-resistance property are proposed.

Figure 8. Flowchart of tamper resistant protection in progress (Martínez-Peláez *et al.*, 2008)

Figure 9. Flowchart of authentication algorithm (Martínez-Peláez *et al.*, 2008)



The distinguished features of the proposed light weight tamper resistance client file in an external memory are as follows:

1. It must be a binary file.
2. It should be password protected automatically on creation using cryptographic key generated from user's biometric and other key generation parameters.
3. It must always reside on the client side not server side.
4. Programs written to manipulate it should be strictly client side. Possible access to the client file on sever should be prevented.
5. Sensitive information for login authentication and access control should be stored in the client not at the server.
6. Login session should be timestamp and be updated at regular interval. So, the login session will expire if there is no activity within the timestamp and therefore logout automatically.
7. On failing to login for a specified number of times the client file should be deleted. The legitimate user will now have to re-register to create the client file again.

5.1 Proposed Light Weight Tamper Resistance Client File Basic System

The proposed light weight tamper resistance client file consists of three main components namely: client file, client and server. The client file is a binary file. This means they can only be read or

written by the program that created it. It stores such information as: biometric, cryptographic key, cryptographic algorithm, user identity and timestamp. The biometric is the unique user template like fingerprint, iris, etc, cryptographic key is the secrete key generated from the user's biometric, the cryptographic algorithm is the cryptographic method used in the generation of the cryptographic key – so, any cryptographic method of one's choice can be used, the user identify like username or email address and timestamp for take rid of login session expiry. This can be shown in fig. 10. This shows that server has no direct access to the client file. So, the client computer acts as an interface between the client file and the server.

5.2 Proposed Light Weight Tamper Resistance Client File Software System

Client file software system is a system written using any of web-based programming language or scripting language. It also, contains other devices that work together for the successful operation of the system. It enables reading or writing and sending and receiving information to and from the client file. It comprises of biometric reader, display, keyboard, client computer, server computer, external memory and clock timer.

Biometric reader attached to client computer or inbuilt allow biometric template of user to be read and transmitted for generation of cryptographic key in the system. Display unit for the display of messages or information to the user. Client computer which transmitting of the biometric from biometric reader and other key generation parameters from keyboard for the generation of cryptographic key, automatically password the created client file store in an external memory with the biometric, authenticate user by verifying the cryptographic key generated from the user's biometric and client file password with the biometric thereby allow the user access to the server, update the timestamp of the client file regularly using the clock timer to automatically logout the user when there is no activity within the timestamp to control server access and delete the client file when fail attempts are exhorted. This can be shown in fig. 11.

5.3 Proposed Light Weight Tamper Resistance Client File Authentication Process

There are four elements involved in client file authentication. These include: user, client file, client computer and server computer. Client file authentication requires three phases namely: registration, login authentication and access control. These are discussed in the sub-sections of this section.

Figure 10. Proposed Client File Basic System

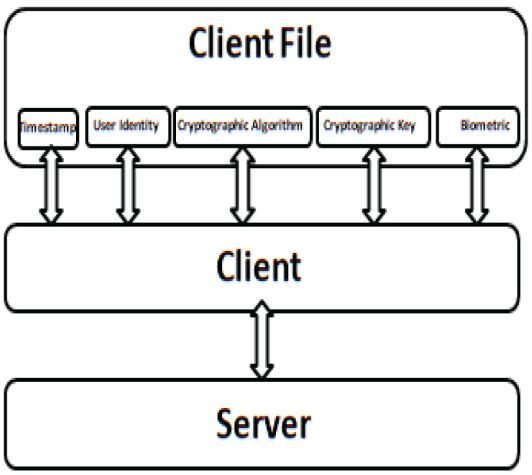
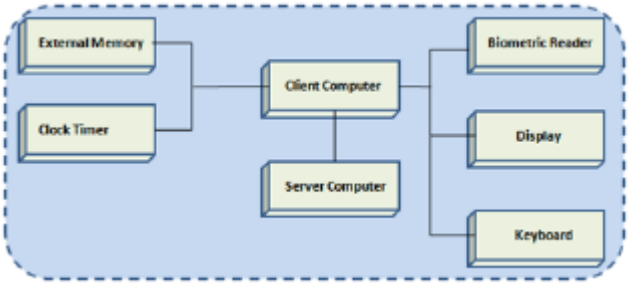


Figure 11. Block diagram of the Proposed Light Weight Tamper Resistance Client File Software System



5.3.1 Registration Phase

This is when the user registered with server as a legitimate user. The client file is initially created in this phase. The phase steps are as follows:

1. The server S_i selects the cryptographic algorithm $Cryp(\bullet)$ to be used.
2. User U_i provides his/her identity ID_i and biometric B_i.
3. The client Computer CL_i generates a cryptographic key K_i from the user U_i's identity ID_i and biometric B_i:

$$K_i = Cryp(ID_i, B_i) \tag{20}$$

4. The client Computer creates a client file CF_i and automatically password it using the biometric B_i.

   That is:

$$TRCF = PWD_{CF_i}(B_i) \tag{21}$$

5. The client Computer gets the current timestamp T_i.
6. The client Computer CL_i store $\{B_i, K_i, Cryp(\bullet), ID_i, T_i\}$ on the client file CF_i.
7. The client Computer CL_i store $\{ID_i\}$ on the server S_i.

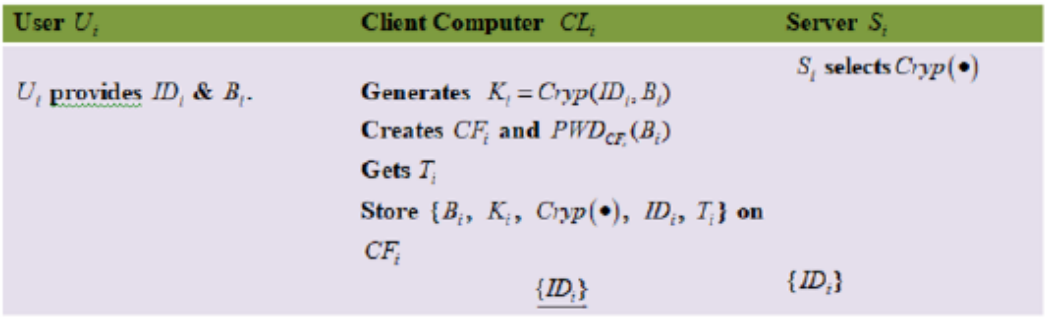
   This can be shown in fig. 12.

5.3.2 Login Authentication Phase

This allows user to login to the server as legitimate user. The user is authenticated and login session is created which enable the user access to the server resources throughout the session lifetime. This phase steps are as follows:

1. User U_i provides his/her identity ID_i and biometric B_i.
2. The client Computer CL_i opens the client file CF_i using biometric B_i provided as the password. If the password is valid go to the next step, otherwise the login requested is rejected.

Figure 12. Registration Phase



3. The client Computer CL_i again generates a cryptographic key $K_i^{'}$ using the cryptographic algorithm $Cryp(\bullet)$ from the user U_i's identity ID_i and biometric B_i provided. If the client file CF_i key $K_i = K_i^{'}$ go to the next step, other the login request is rejected. If fail attempts FA_i are reached the client file CF_i is deleted.
4. Now both the client Computer CL_i and server S_i computes the session key $sk(ID_i)_{CL_i} = sk(ID_i)_{S_i}$.

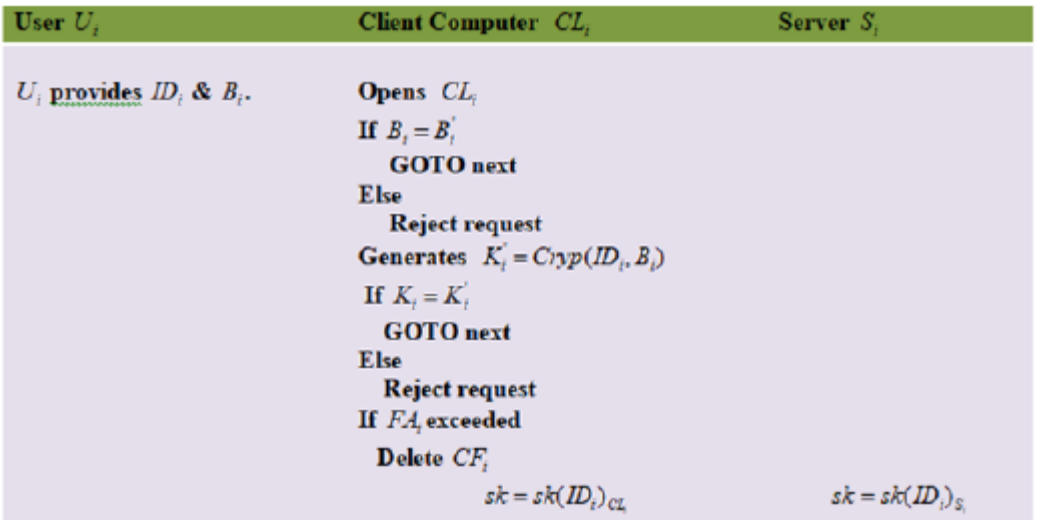
This can be shown in fig. 13.

5.3.3 Access Control Phase

This allows continual regulation of who are accessing the server resources. The user access is monitored at regular interval to prevent impersonation attack. The phase steps are as follows:

1. One the session has been established, the client Computer CL_i get the current timestamp $T_i^{'}$ and update the client file CF_i timestamp T_i.

Figure 13. Login Authentication Phase



2. Immediately there is no activity for a specified time t_i or timestamp expired, the login request is reset and login authentication phase starts again.

   This can be shown in fig. 14.

6. EVALUATION OF THE PROPOSED LIGHT WEIGHT TAMPER RESISTANCE CLIENT FILE

The proposed light weight tamper resistance client file in an external memory has been presented. Its basic system design, software design and modeling have been discussed. The proposed design will be evaluated based on efficiency, tamper resistance and impersonation attack. These are discussed in the next sub-sections of this section.

6.1 Efficiency

Now, the proposed scheme's relative computational cost is contrasted with that of Kumari *et al.* (2014). This analysis of the target protocols specifically separated the operations of the protocols into crypto-operations. A one-way hashing operation, symmetric encryption/decryption operation and modular exponentiation operation respectively, have computing times of 0.00032 s, 0.0056 s and 0.0192 s, and the computational expenses of XOR, timestamp, and random number generation are typically disregarded because they are significantly less expensive than one-way hash computations (Lee *et al.*, 2013).

In the registration phase, the proposed protocol uses symmetric encryption/decryption operation. The Kumari *et al.*'s scheme uses four one-way hashing operations and one symmetric encryption/decryption operation.

In the login and authentication phase, the proposed scheme uses two symmetric encryption/decryption operations. The Kumari *et al.*'s scheme uses ten one-way hash function operations, two symmetric encryption/decryption operations and three modular exponentiation operations.

Therefore, the proposed is lower in computational cost than Kumari *et al.*'s scheme by slight difference in registration phase but lower in computational cost that Kumari el al.'s scheme by high difference in login and authentication as shown in fig. 15 and table 1.

6.2 Tamper Resistance

Tamper-resistance reduces the risk of unauthorized access and corruption of information. This is absent in non-tamper resistance devices like external memory used by Kumari *et al.*'s scheme. Therefore there is possibility of an adversary U_a to access, alter or damage the information $\left\{Y_i, EId_i, W_i, h(\bullet), p, q, g\right\}$ stores in U_i's external memory. This can lead to legitimate user U_i

Figure 14. Access Control Phase

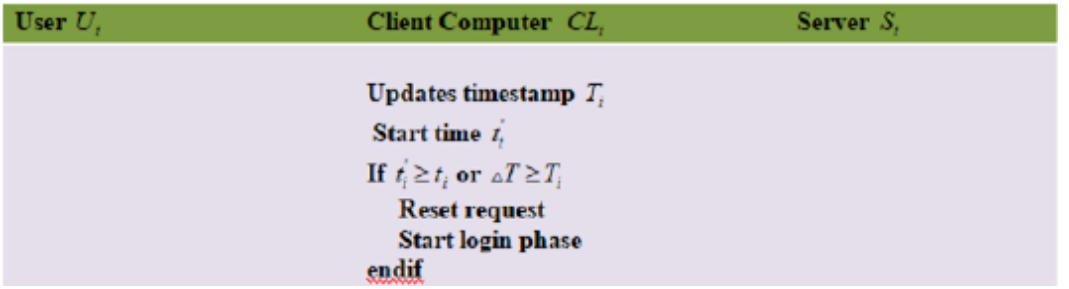
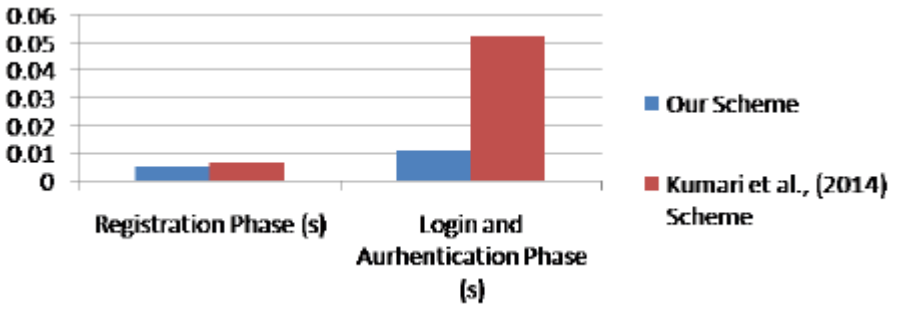


Table 1. Computational cost of our proposed scheme and Kumari *et al.*, (2014) scheme. T(Hash) is one-way hashing operation time, T(Sym) is symmetric encryption/decryption operation time and T(Modular) is modular exponentiation operation time

	Registration Phase	Login and Authentication Phase
Our Scheme	1T(Sym) = 0.0056 s	2T(Sym) = 0.0112 s
Kumari *et al.*, (2014)	4T(Hash) + 1T(Sym) = 0.00688	10T(Hash) + 2T(Sym) + 4T(Modular) = 0.0528

Figure 15. Computational cost of our proposed scheme and Kumari *et al.*, (2014) scheme



computed W_i^* not to be equal to the stored W_i. As such, he will not be authenticated because his login session will be discarded. This is also another form of denial of service (DoS) attack. But the feature formulated by our proposed scheme when strictly followed will enable the client file to exhibit light weight tamper resistance property. The creation of the client file CF_i as a binary file, such that only the client software that creates it can read or write to it, makes it light weight tamper resistance. And also, automatic password of the client file CF_i, that is $PWD_{CF_i}\left(B_i\right)$, restrict reading and writing to the file by only legitimate user.

6.3 Impersonation Attack

Kumari *et al.*'s scheme resist user impersonation attack because if an adversary U_a could obtain the stored information $Y_i = h(Id_i \,||\, x) \oplus h\left(Pw_i, r_i\right)$ from U_i's external memory, he still cannot retrieved $h(Id_i \,||\, x)$ because it is XORed with $h\left(Pw_i, r_i\right)$ that prevent the retrieval of $h\left(Pw_i, r_i\right)$ by any one accept U_i. But, if the U_i's credentials and external memory are stolen by adversary U_a, U_a can:

$$\text{Retrieve } r_a = Z_i \oplus \left(Id_i \oplus Pw_i\right) \tag{22}$$

$$\text{Computes } W_a^* = h(h\left(Pw_a, r_a\right) \,||\, Id_a) \tag{23}$$

Compare W_a^* with W_i stored in the external memory.

Since Id_a and Pw_a are the actual credentials of U_i therefore $W_a^* = W_i$, hence the login session will not be discarded. So, the adversary impersonates the user U_i and continues with authentication.

But in our proposed scheme to impersonate user U_i, the attacker must generate valid cryptographic key $K_i = Cryp\left(ID_i, B_i\right)$. Since the user U_i's biometric B_i is unique, possession based and cannot be redistribute the login request will be rejected and when it login attempts reached maximum, the client file CF_i will be deleted. Hence, the proposed system resists user impersonation attack.

7. CONCLUSION

Client file that exhibits light weight tamper resistance property stored in external memory is proposed as an alternative to smart card. The advantage of using this special client file is portability and ease of acquirement especially in schools online portals, online resources portal and e-commerce portals. Smart card design is thoroughly revised to identify the feature or characteristics that make the smart to have tamper resistance property. A technique to make a client file in an external memory to exhibit a light weight tamper resistance property is proposed. Characteristics or features that will make a client file to exhibit light weight tamper resistance property are formulated. The Kumari *et al.*'s scheme which is the latest research that uses external memory for remote user authentication has been reviewed. The basic system design and software design of the proposed client file is presented and modeled. This will enable implementation of the proposed system using any prepared programming or scripting language of one's choice. The proposed scheme and reviewed scheme are also evaluated for efficiency, tamper resistance and impersonation attack.

REFERENCES

Adavalli, S. R. (n.d.). *Smart Card Solution: Highly secured Java Card Technology.* https://www.cs.auckland.ac.nz/courses/compsci725s2c/archive/termpapers/725adavalli.pdf

Akram, R. N., Markantonakis, K., & Mayes, K. (2011, October). User centric security model for tamper-resistant devices. In *2011 IEEE 8th International Conference on e-Business Engineering* (pp. 168-177). IEEE. doi:10.1109/ICEBE.2011.69

Bolchini, C., Salice, F., Schreiber, F. A., & Tanca, L. (2003). Logical and physical design issues for smart card databases. *ACM Transactions on Information Systems*, *21*(3), 254–285. doi:10.1145/858476.858478

Boneh, D., Lie, D., Lincoln, P., Mitchell, J., & Mitchell, M. (1999). Hardware support for tamper-resistant and copy-resistant software. *Nov, 14*, 1-13.

Buhari, B. A., Obiniyi, A. A., Junaidu, S. B., & Kana, A. F. (2022). Trends in Remote User Authentication Based on Smart Card and External Memory. *International Journal of Security and Privacy in Pervasive Computing*, *14*(1), 1–10. doi:10.4018/IJSPPC.307148

Chen, B. L., Kuo, W. C., & Wuu, L. C. (2012). A secure password-based remote user authentication scheme without smart cards. *Information Technology and Control*, *41*(1), 53–59. doi:10.5755/j01.itc.41.1.975

Chong, C. N., Ren, B., Doumen, J., Etalle, S., Hartel, P. H., & Corin, R. (2004, August). License protection with a tamper-resistant token. In *International Workshop on Information Security Applications* (pp. 223-237). Springer.

Deville, D., Galland, A., Grimaud, G., & Jean, S. (2003, September). Assessing the future of smart card operating systems. Conference E-SMART.

Fan, C. I., Chan, Y. C., & Zhang, Z. K. (2005). Robust remote authentication scheme with smart cards. *Computers & Security*, *24*(8), 619–628. doi:10.1016/j.cose.2005.03.006

Ferres, E., Immler, V., Utz, A., Stanitzki, A., Lerch, R., & Kokozinski, R. (2018, October). *Capacitive multi-channel security sensor ic for tamper-resistant enclosures. In 2018 IEEE SENSORS*. IEEE.

Guo, H. (2002). Smart Cards and their Operating Systems. *Helsinki Univ. Technol. Softw. Multimed. Lab. Tech. Rep*, 1-15.

Hassler, V. (2002). *Java Card for e-payment Applications*. Artech House.

He, D., Wang, D., & Wu, S. (2013). Cryptanalysis and improvement of a password-based remote user authentication scheme without smart cards. *Information Technology and Control*, *42*(2), 105–112. doi:10.5755/j01.itc.42.2.2554

Jiang, Q., Ma, J., Li, G., & Ma, Z. (2013). An improved password-based remote user authentication protocol without smart cards. *Information Technology and Control*, *42*(2), 113–123. doi:10.5755/j01.itc.42.2.2079

Jurgensen, T. M., & Guthery, S. B. (2002). *Smart cards: the developer's toolkit*. Prentice Hall Professional.

Khan, M. F. F., & Sakamura, K. (2015, December). Tamper-resistant security for cyber-physical systems with eTRON architecture. In *2015 IEEE International Conference on Data Science and Data Intensive Systems* (pp. 196-203). IEEE. doi:10.1109/DSDIS.2015.98

Kim, Y., Back, J., & Kim, J. (2021). A Tamper-Resistant Algorithm Using Blockchain for the Digital Tachograph. *Electronics (Basel)*, *10*(5), 581. doi:10.3390/electronics10050581

Kumari, S., Khan, M. K., Li, X., & Wu, F. (2014). Design of a user anonymous password authentication scheme without smart card. *International Journal of Communication Systems*, *29*(3), 441–458. doi:10.1002/dac.2853

Kursawe, K., Sadeghi, A. R., Schellekens, D., Skoric, B., & Tuyls, P. (2009, July). Reconfigurable physical unclonable functions-enabling technology for tamper-resistant storage. In *2009 IEEE International Workshop on Hardware-Oriented Security and Trust* (pp. 22-29). IEEE. doi:10.1109/HST.2009.5225058

Lassus, M. (1997). *Smart-cards-a cost-effective solution against electronic fraud*. Academic Press.

Lee, C. C., Chen, C. T., Wu, P. H., & Chen, T. Y. (2013). Three-factor control protocol based on elliptic curve cryptosystem for universal serial bus mass storage devices. *IET Computers & Digital Techniques*, *7*(1), 48–55. doi:10.1049/iet-cdt.2012.0073

Leng, X. (2009). Smart card applications and security. *Information Security Technical Report, 14*(2), 36-45.

Lu, M. C., Huang, Q. X., Chiu, M. Y., Tsai, Y. C., & Sun, H. M. (2022). PSPS: A Step toward Tamper Resistance against Physical Computer Intrusion. *Sensors (Basel)*, *22*(5), 1882. doi:10.3390/s22051882 PMID:35271029

Martínez-Peláez, R., Rico-Novella, F., & Satizábal, C. (2008, April). Secure smart card reader design. In *2008 IEEE International Symposium on Consumer Electronics* (pp. 1-3). IEEE.

Mayes, K. E., & Markantonakis, K. (Eds.). (2017). *Smart cards, tokens, security and applications* (Vol. 1). Springer. doi:10.1007/978-3-319-50500-8

Mohammed, L. A., Ramli, A. R., Prakash, V., & Daud, M. B. (2004). Smart card technology: Past, present, and future. *International Journal of The Computer, the Internet and Management, 12*(1), 12-22.

Rhee, H. S., Kwon, J. O., & Lee, D. H. (2009). A remote user authentication scheme without using smart cards. *Computer Standards & Interfaces*, *31*(1), 6–13. doi:10.1016/j.csi.2007.11.017

Selimis, G., Fournaris, A., Kostopoulos, G., & Koufopavlou, O. (2009). Software and hardware issues in smart card technology. *IEEE Communications Surveys and Tutorials*, *11*(3), 143–152. doi:10.1109/SURV.2009.090310

Sharma, Y. K., & Dixit, S. (2018). Smart Card for Healthcare System. *International Journal of Electronics Engineering.*, *10*(1), 359–362.

Shibagaki, K., Nozaki, Y., & Yoshikawa, M. (2018, June). Tamper Resistance Evaluation of Noise Based Countermeasure for IoT Devices. In *2018 IEEE International Meeting for Future of Electron Devices, Kansai (IMFEDK)* (pp. 1-2). IEEE. doi:10.1109/IMFEDK.2018.8581975

Srivastava, P. C., Agrawal, A., Mishra, K. N., Ojha, P. K., & Garg, R. (2013, January). Fingerprints, Iris and DNA Features based Multimodal Systems: A Review. *I.J. Information Technology and Computer Science*, *02*(2), 88–111. doi:10.5815/ijitcs.2013.02.10

Tan, Z. (2009, June). Security analysis of two password authentication schemes. In *2009 Eighth International Conference on Mobile Business* (pp. 296-300). IEEE. doi:10.1109/ICMB.2009.57

Taponen, V. (2000). Tamper-resistant smart cards—Too much to ask for? *Proc. Helsinki University of Technology Seminar on Network Security*

Bello Alhaji Buhari obtained B.Sc. in Computer Science at Usmanu Danfodiyo UniversitySokoto – Nigeria and M.Sc. in Computer Science at Ahmadu Bello University Zaria –Nigeria. He is now pursuing Ph.D. in Computer Science at Ahmadu Bello University, Zaria – Nigeria. He is a Lecture in the Department of Computer Science, Usmanu Danfodiyo University Sokoto – Nigeria since 2004. His research interest include: Web Security and Cryptography.

A. A. Obiniyi received his Ph.D degree in Computer Science from Ahmadu Bello University (ABU), Zaria in Kaduna State of Nigeria in 2009. He is a Professor of Computer Science and a member of Nigeria Computer Society (NCS), Internet Society (ISOC), Academia in Information Technology Professionals (AITP), Institute of Electrical and Electronic Engineers (IEEE) and a Chartered member of Computer Professionals (Registration Council of Nigeria) [CPN]. He lectures in the Department of Computer Science of Ahmadu Bello University, Zaria – Kaduna State. Presently, he is co-supervising eight Ph. D. and thirteen Master of Computer Science students with many Ph.D. and Master of Computer Science scholars completed their studies. He also has many publications to his credit. His research interests include Computer Networking, Cyber Security and Database Development among others.

Sahalu Junaidu Balarabe is a Professor at the Ahmadu Bello University, Zaria Kaduna State Nigeria. He is a Life Member of Association of Computing Machinery (ACM); Member, Institute of Electrical and Electronic Engineers (IEEE); Member, Computer Professionals (Registration Council) of Nigeria; Member, Nigerian Computer Society. He is the Pioneer Head of Department, Department of Computer Science, Ahmadu Bello University Zaria from 2016 to Date.

Armand F. Donfack Kana received his B.Sc. degree in computer science from the University of Ilorin, Nigeria, M.Sc. and Ph.D. degrees in computer science from the University of Ibadan, Nigeria. He is currently a READER at the Department of Computer Science, Ahmadu Bello University, Zaria, Nigeria. His research interests include Knowledge Representation and Reasoning, Machine Learning, Formal Ontologies and Soft Computing.