
# Cryptographic-Paging Mechanism with Secure Coprocessor to Enhance Digital Signature Integrity for Cyber Hygiene Standard

**Akhigbe-mudu, T.E**

African Institute of Science Administration and Commercial Studies, Lome – Togo

**Abstract**

Digital Signature is a mathematical algorithm to validate the authenticity and integrity of an electronic message. Modern browsers enable a wide range of sensitive operations over SSL/TLS connections. Cryptographic protocols (SSL/TLS) are used to imbue web communications with intgrity, security, and resilience against unauthorized tampering. It is interesting to note that PKI (Public Key Infrastructure) uses the TLS protocol to establish secure connections between clients and servers over the internet. Thereby ensuring that the information relayed is encrypted and can not be read by an external third party. Great scholars over the last decades have proved that a great number of internet users either tend to ignore or do not understand browser security indicators. Hence there are flaws and vulnerabilities in the certificates and websites. This study proposes a technique hereby referred as cryptographic paging mechanism with secure coprocessor to enhance the integrity of digital Signature. As a result of the encryption and integrity check,data security on the outgoing page is maintained.

**Key Words:** Encryption, Cryptographic Algorithms, Secure Coprocessor, Flaws and Vulnerabilities

*Corresponding author's e-mail: akhigbe-mudut@iaec-university.tg* 

## INTRODUCTION

Digital Signature (DS) is a mathematical technique to validate the authenticity and integrity of a message, software or digital document. It can be described as the digital equivalent of a hand-written signature or stamped, seal but it offers offering more internet security It is intended to solve the problem of tampering and impersonation in digital communications. Digital signature can be used to provide evidence of originality, identity and status of electronics documents, transaction or digital messages. Signers can also use it to acknowledge informed consent [1]. Digital signatures are mathematical algorithms used to validatethe authenticity and integrity of an electronic message. These could be email messages, a credit card transaction, a macro, or a digital document. Digital signatures create a virtual "fingerprint" that iscompletely unique to a person, or an entity. Hence, DS can be used to protect the contents of messages, and to ensure that they were written by 'who' they claim to have been. If we are interested in looking at digital signature from a deeper level, then you need to know one thing that -- digital signatures work by applyinga hash function to a message. In most cases, a user's private key will be used to create a "hash," which is afixed-length string of numbers and letters. And this string (hash) is totally unique to the message being hashed, or document being signed; this is depicted in figure 1 (Block Diagram of a Digital signature). In this diagram: each person adopting this scheme has a public-private key pair in cryptography. The pairs of key used for encryption or decryption are different for every signature processes. The private key used for signing is referred to as the signature key and the public key as the verification key as shown in figure 1. Then, the signer feeds data to the hash function and generates a hash of data strings of that message. The data string value and signature key are fed into the signature algorithm which produces the digital signature. And the signature is appended to the data and both are sent to the verifier to secure the message. The verifier feeds the digital signature and the verification key into the verification algorithm. Thus, the verification algorithm gives some value as output which is hereby referred as a ciphertext. It is the verifier that runs the hash function to generate hash values. During this process, the verification, the signature, this hash value, and output of verification algorithm are compared with each variable. Based on the comparison result, the verifier decides whether the digital signature is valid or invalid [9]. Therefore, the digital signature is generated by the 'private' key of the signer and no one else can have this key to secure

the data,the signer cannot repudiate signing the data in the future to secure that data by the cryptography.

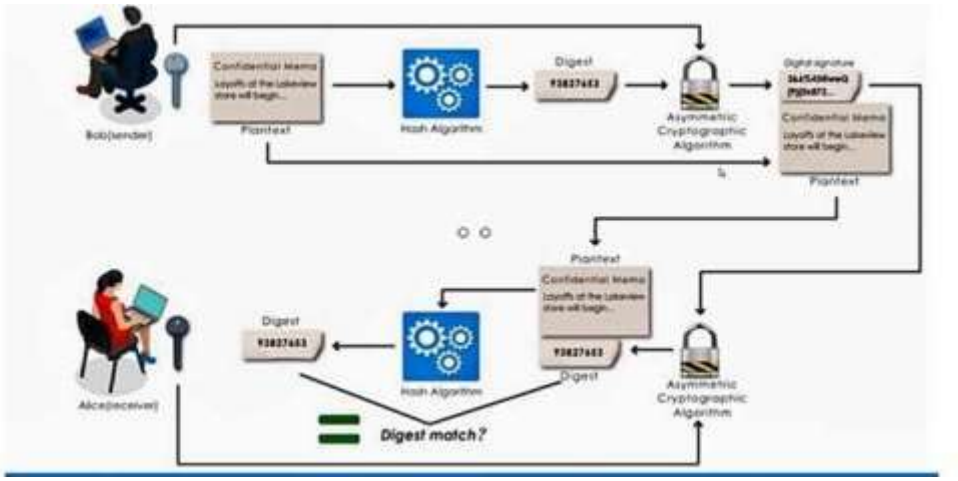



**Figure 1: Block Diagram of a Digital Signature**

## The Importance of Digital Signature

### (i) Authentication

Since in most cases, digital signatures are created using the sender's private encryption key, it is possible for you to verify the identity of the message source or sender of the message.

### (ii) Data Integrity

Remember, hash functions produce a digital signature by looking at the ENTIRITY of a particular message.It means that--if any part of the message changes, so does the hash function; you would immediately knowthis. It also means that if a message is intercepted in-transit and changed before it reaches you, when you will verify the digital certification, it will fail. You would be sure to know that the data or message has lostits integrity.

### (iii) Non-repudiation

You should also know that one is advised never to share one's PRIVATE KEY with anybody or any entity.This is the foundation of non-repudiation that it cannot be duplicated by anybody else, the sender's identityis verified and fully-established...it makes the document or message totally permissible as a proof, in any court of law. In most parts of the world, digital signatures are considered legally binding and hold the samevalue as traditional document signatures. However, it's important to recognize that newly-trained security analysts often overlook the importance and centrality of digital signatures to cyber

security. Though technologies like Virtual Private Network (VPN) with real-time encryption may ensure the security of data when it is in-transit, but they cannot authenticate the identity of the creator of a message in the same way that a digital signature can. Other security technologies have their own place in the scheme of things, but never forget that digital signatures still cover a lot of ground.

**Statement of the Problem**

Modern mobile browsers enable a range of sensitive operations over SSL/TLS connections. Mobile SSL and its successor, TLS, are encryption-based security protocols that enable businesses to ensure integrity, privacy and security of data in internet communications. SSL certificates help prevent a range of cyber- attacks including eavesdropping, spoofing and man-in-the-middle attacks, among others. SSL/ TLS certificates, however, do not guarantee the trustworthiness of the website [2]. There are flaws and vulnerabilities in the certificates and the websites that erode the security posture. Yet browsers are increasingly being relied upon to perform security sensitive operation. Studies over the last decade have repeatedly shown that average users either ignore or do not understand browser security indicators [3]. It'spossible that an application might use SSL incorrectly such that malicious entities may be able to interceptan app's data over the network. One study of technically skilled Android users, found that half of the internetusers could not tell, from the indicators provided by Android's browser, whether Secure Socket Layer (SSL) was in effect or not. This has significantly changed the use and consistency of the security indicators and certificate information that alert users of site identity and the presence of strong cryptographic algorithms. One of the arguable reasons for this is that the Hamming Distance (HD) between the eye-level SSL indicators and the no SSL indicators is as small as shown in the magnified view of figure 2. To help ensure that this does not happen to your app, this article proposes a cryptographic –paging mechanism with securecoprocessor to enhance the integrity of the Digital Signature. When an SSL certificate is issued, what's really happening is, one is sending an unsigned certificate to a trusted Certificate Authority (CA), then validates the information contained in the certificate and applies its digital signature using one of its roots'private keys [10].


**Figure 2: Near-Invisible SSL Indicators**

**Related Work**

With all companies having a strong online presence and rising technology evolution, security is a major issue. Companies are dealing with sensitive information every day on the internet. This may be with the employees, clients, authorities, and practically all other stakeholders. But is it safe to do so? The answer isa clear no. Safety protocols are essential for maintaining the security of the information. A digital signature is one of such technique to ensure a company's safety in today's world [4]. It is a method for validating the authenticity and integrity of any type of file. This is similar to handwrittensignature or stamped seals but with more security features. It came in the frame with an aim to solve tampering and identity-related problems. It is a solid proof to authenticate origin, identity, transactions, and more. The senders and recipients, both benefit from this to

certificates, however, do not guarantee the trustworthiness of the website. There are flaws ensure their safety and security. Invisible to the end-user, a process called the "TLS/SSL handshake" creates a protected connection between your web server and web browser nearly instantaneously every time you visit a website. Websites secured by a TLS/SSL certificate will display HTTPS and the small padlock icon in the browser address bar. TLS/SSL certificates are used to protect both the end users' information while in transit, and to authenticate the website's organization identity to ensure users are interacting with legitimate website [11].

SSL and its successor, TLS, are encryption-based security protocols that enable businesses to ensure integrity, privacy and security of data in internet communications. SSL certificates help prevent a range ofcyber-attacks including eavesdropping, spoofing and man-in-the-middle attacks, among others. SSL/ TLS

and vulnerabilities in the certificates and the websites that erode the security posture. By

implementing the latest SSL certificate best practices, businesses can improve SSL security and overall web security (see figure 3).


**Figure 3: Certificate Signing Request and SSL Overview**

How do we build distributed systems that are secure? Cryptographic techniques can be used to secure the communications between physically separated systems, but this is not enough: it must be able to guarantee the privacy of the cryptographic keys and the integrity of the cryptographic functions, in addition to the integrity of the security kernel and access control databases on the machines [12]. Physical security is a central assumption upon which secure distributed systems are built; without this foundation even the best cryptosystem or the most secure kernel will crumble [5]. This work, addresses the distributed security problem by proposing the addition of a small, physically secure hardware module, a secure coprocessor. The axiom here is that secure coprocessors are able to maintain the privacy of the data they process.

**How do digital signatures work?**

Digital signatures, at the most fundamental level, are mathematical algorithms used to validate the authenticity and integrity of an electronic message. This "message" could be an email, a credit card transaction, a macro, or a digital document. These digital signatures create a virtual "fingerprint" that is completely unique to a person, or an

entity. That's why these can therefore be used not just to protect the contents of messages, but also to ensure that they were written by 'who' they claim to have been. If we are interested in looking at digital signature from a deeper level, then you need to know one thing that -- digital signatures work by applying a hash function to a message. In most cases, a user's private key will be used to create a "hash," which is a fixed-length string of numbers and letters. And this string (hash) is totally unique to the message being hashed, or document being signed as shown in figure 5. What is PKI (Public Key Infrastructure) used for? In a nutshell, PKI is responsible for making online interactions more secure,and it does this by: Establishing the identity of endpoints on a network. Encrypting the flow of data via the network's communication channels. It does this by using private keys and public keys for encryption and decryption respectively, which are facilitated in turn by digital certificates.

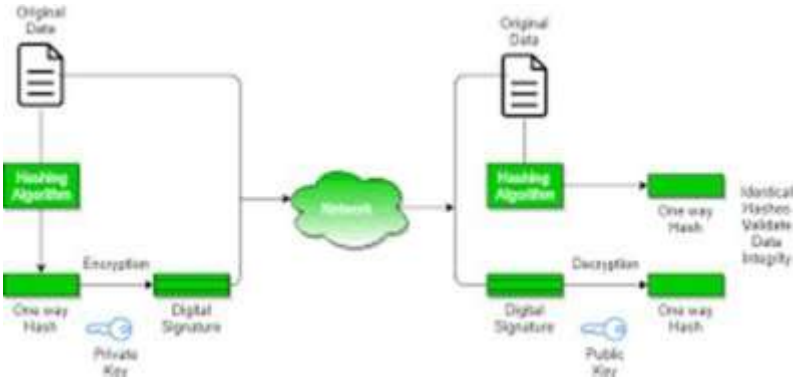



**Figure 4: Processes Hash Function**

A method and System for maintaining integrity and confidentiality of pages paged to an external Storage unit from a physically secure environment. An outgoing page is selected to be exported from a physically secure environment to an insecure environment [13]. An integrity check value is generated and Stored for the outgoing page. In one aspect, it takes the one-way hash of the page using a well-known one-way hash function. The in/out pages are encrypted using a cryptographically Strong algorithm. The encrypted page is exported to the external Storage. Sequel to the encryption and integrity check, data security on the outgoing page is maintained [6].

## Methodology

### The Signature Verification Environment

The signature verification process Shown in figure 5, is integrated with the signature process mechanism. The separation of the two processes is justified by the fact that those who assume responsibilities are fewer than those who can verify their fulfillment. This can be likened to a dictatorial organization, where fewer people are entitled to issue orders, and many are entitled to verify and carry out those orders. This is equally applicable to signature verifiers, therefore it is not recommended to separate the two types of operations.

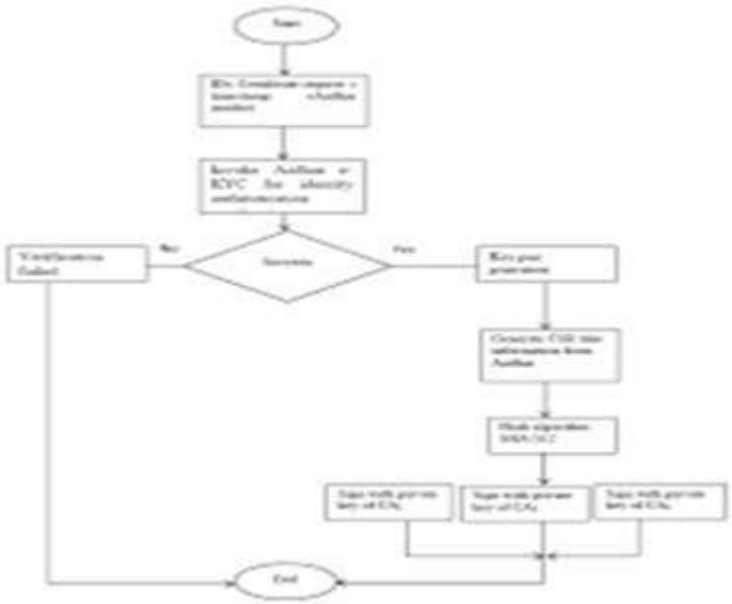


**Figure 5: Digital Signature Algorithm Flow Chart**

### Digital Signature Mechanism-RSA

The RSA (short of Rivest, Shamir, Adleman) used modulo concept in arithmetic to perform a digital signature [2]. It provides message recovery. The RSA public-key encryption scheme has the message M and the ciphertext. Key Generation process in RSA public key crypto-systems are as:

– Both sender and Receiver create primes p and q that are two large distinguished random numbers.

Computes $n = p \cdot q$ (1)

and $\varphi(n) = (p-1)\cdot(q-1)$ . (2)

Selects an integer number e as random such that $1 < e < \varphi(n)$ ,

Where $\gcd(e,\varphi(n)) = 1$. (3)

Computes integer d as a unique number such that $1 < d < \varphi(n)$ ,

where $ed \equiv 1 \bmod \varphi(n)$ . (4)

Thus, sender has the public key (n, e) and private key is d.

Signature Generation processes:

- A message $m \in M$, Sender defines m with a number $m \in Z$ through a map $R: M \rightarrow Z$
- Sender computes the signature $s = \overline{m}d(\bmod .n)$ (5)

**Verification process of Alice Signature is as the following:**

- Bob chooses the public key (e, n) of Alice.
- Bob computes $\overline{m} = s.e(\bmod .n)$ (6)
- Bob verifies that $\overline{m} \in \overline{M}$ , where $\overline{M}$ denotes the set of images of R . The signature rejects, if m does not hold else recovers the message as $m = R^{-1}((\overline{m})$ .

**DSA Signature Scheme**

DSA(short of Digital signature algorithm) that use different domain parameters such as x is the private key,k is per message secret key number, signed the data, and the hash function [7]. Digital signature algorithm checked by y that is the public key, checked the data and also the same hash function that is used in creating the signature. So, the parameters implemented are as follows:

− A prime modulus p

− A prime divisor for ( p −1) is q .

− A generator for the sub group for order q (mod p) is g .

− The private key that is a random integer selected in the range [1, q −1] is x .

− The public-key is y. It is acquired by $y = g^{x}(\mathrm{mod}.p)$ (7)

− Message has k as secret key.

The message M has the signature consists of both numbers r and s implemented by using:

$$r = \left(g^{k} \ \mathrm{mod}.p\right)\mathrm{mod}.q \quad (8)$$

z = the farthest to the left min (bits) for Hash (M ) .

s = (k −1(z + x.r ))mod q ; (9)
(r, s) is the signature created.

Alice transmits message M , and the signature (r, s) to Bob. To verify the signature, Bob implements the following steps: He will verify which $0 < s' < q$ and $0 < r' < q$ ; the signature will reject; if any one of the condition is violated. If the conditions are not violated in the first phase, Bob calculates

$$w = \left(s'\right)^{-1} \mathrm{mod}.q \quad (10)$$

z is the farthest to the left min (N,outlen) bits for H ( $\overline{M}$ )

$$U_1 = (z, w)\mathrm{mod}.q$$

$$U_2 = \left((r')w\right).\mathrm{mod}.q \quad (11)$$

$$V = \left((g)^{u_1}(y)^{u_2}\right)\mathrm{mod}.p\mathrm{mod}.q \quad (12)$$

If $V = r'$ then the signature is accepted. (13)

### Hash function and Digital Signature

A digital signature is created when the sender of data creates a "hash" value for the data, encrypts the hashwith his Private Key, and then transmits the data (plus encrypted hash). The receiver has access to the senders Public Key and decrypts the hash, comparing the hash value he calculates from the data, which should come to the same value [8]. When this happens the receiver has verified the sender's identity and also the integrity of the data sent. SSL uses asymmetric encryption for digital signatures and (in some cases)encryption of symmetric key (see figure 6)

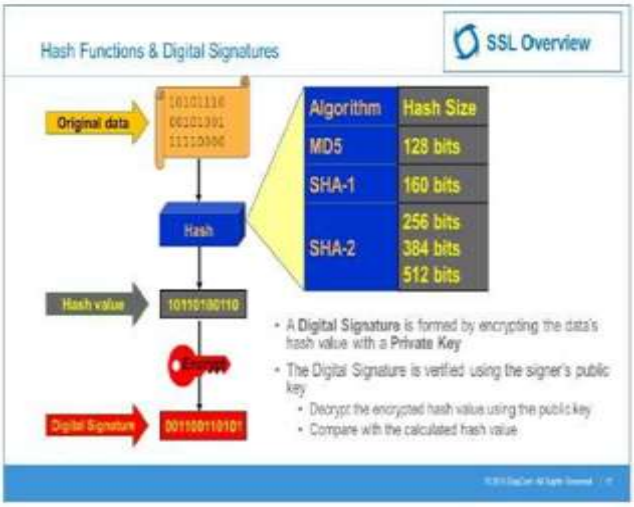



**Figure 6: Hash function and Digital Signature**

A cryptographic hash function is a mathematical algorithm that maps data of arbitrary size (often called the"message") to a bit string of a fixed size (the "hash value", "hash", or "message digest") and is a one-way function [14], that is, a function which is practically infeasible to invert. A digital signature is a mathematical scheme for verifying the authenticity of digital messages or documents. The signature is created using the signer's private key.

**Simulation**

**DIGITAL SIGNATURE DOCUMENT DEVELOPMENT PHP SCRIPT**

```
<?php
define("MULTIPART_BOUNDARY", " ---WebKitFormBoundary7MA4YWxkTrZu0gW");
class HSLPackageCreation
{
        //Define api key and api url
        private $url = "https://app.hellosign.com/sign/ed82ce87dda6ec0316abf80e256a4bff5d44f500";
        private $key = "86444a15a739beff87471f25fb7d8b50ec01f5ef52e95ab9d31b00ed6b236c46";
        private $packageAppend = "/packages/";
        private $tokenAppend = "/signerAuthenticationTokens/";
        //Constructor method for HSLPackageCreation
        public function__construct()
        {
        }
        //Create package
        public function buildPackage($firstName, $lastName, $email)
        {
                $build = array(
                        'type' => 'PACKAGE',
                        'status' => 'DRAFT',
                        'roles' => array(
```

```
'id' => 'Signer1', 'type' => 'SIGNER',
                    'signers' => array(
                        array(
                            'email' => $email,
                            'firstName' => $firstName,
                            'lastName' => $lastName,
                            'id' => 'Signer1',
                        )
                    ) ,
                ) ,
                array(
                    'id' => 'Sender1',
                    'type' => 'SIGNER',
                    'signers' => array(
                        array(
                            'email' => 'mail32@mailinator.com',
                            'firstName' => 'John',
                            'lastName' => 'Smith',
                            'id' => 'Sender1',
                        )
                    ) ,
```

```
                    ) ,
                ) ,
                'name' => 'PHP Application Example',
            );
            $packageJSON = json_encode($build);
            $packageId = json_decode($this->sendRequest($this->packageAppend, $packageJSON, NULL, NULL), true);
            return $packageId;
        }
        //Upload document
        public function buildDocument($packageId, $firstName, $lastName, $residentialAddress, $city, $state, $sex, $permanentAddress, $phoneNumber, $emailAddress, $staeofOrigin, $nextofKin, $carType, $DLN, $carRegnumber, $chasis, $highestQual, $institutionAttended, $phoneNumber2, $emailAddress2, $nextofKin2, $yes, $no, $med1, $med2, $illness, $details, $BVN, $bankaccountNumber, $bankaccountName, $bankName, $declaration, $signature)
        {
            $build = array(
                'fields' => array(
                    array(
                        'value' => $firstName,
                        'name' => 'first_name',
                    ) ,
                    array(
                        'value' => $lastName,
```

```
            'name' => 'last_name',
        ) ,
        array(
            'value' => $address,
            'name' => 'permanentAddress',
        ) ,
        array(
            'value' => $city,
            'name' => 'city',
        ) ,
        array(
            'value' => $state,
            'name' => 'stateofOrigin',
        ) ,
        array(
            'value' => $sex,
            'name' => 'sex',
        ) ,
        array(
            'value' => $residentialAddress,
            'name' => 'residentialAddress',
        ) ,
```

```
array(
        'value' => $phoneNumber,
        'name' => 'phone_number',
) ,
array(
        'value' => $emailAddress,
        'name' => 'email',
) ,
array(
        'value' => $nextofKin,
        'name' => 'nextofKin',
) ,
array(
        'value' => $carType,
        'name' => 'carType',
) ,
array(
        'value' => $DLN,
        'name' => 'DLN',
) ,
array(
        'value' => $carRegnumber,
```

```
            'name' => 'carRegnumber',
        ) ,
        array(
            'value' => $chasis,
            'name' => 'chasis',
        ) ,
        array(
            'value' => $highestQual,
            'name' => 'highestQual',
        ) ,
        array(
            'value' => $institutionAttended,
            'name' => 'institutionAttended',
        ) ,
        array(
            'value' => $phoneNumber2,
            'name' => 'phoneNumber2',
        ) ,
        array(
            'value' => $emailAddress2,
            'name' => 'emailAddress2',
        ) ,
```

```
array(
        'value' => $nextofKin2,
        'name' => 'nextofKin2',
) ,
array(
        'value' => $yes,
        'name' => 'yes',
) ,
array(
        'value' => $no,
        'name' => 'no',
) ,
array(
        'value' => $med1,
        'name' => 'med1',
) ,
array(
        'value' => $med2,
        'name' => 'med2',
) ,
array(
        'value' => $illness,
```

```
                'name' => 'illness',
        ) ,
        array(
                'value' => $details,
                'name' => 'details',
        ) ,
        array(
                'value' => $BVN,
                'name' => 'BVN',
        ) ,
        array(
                'value' => $bankaccountNumber,
                'name' => 'bankaccountNumber',
        ) ,
        array(
                'value' => $bankaccountName,
                'name' => 'bankaccountName',
        ) ,
        array(
                'value' => $bankName,
                'name' => 'bankName',
        ) ,
```

```
                array(
                    'value' => $declaration,
                    'name' => 'declaration',
                ) ,
                array(
                    'value' => $signature,
                    'name' => 'signature',
                ) ,
            ) ,
            'extract' => true,
            'name' => 'Driver Application',
            'id' => 'Driver Application'
        );
        $documentJSON = json_encode($build);
        $postdata = "--" . MULTIPART_BOUNDARY . "\r\n";
    $postdata .= "Content-Disposition: form-data; name=\"file\"; filename=\"driver_application2.pdf\"\r\n";
    $postdata .= "Content-Type: application/pdf" . "\r\n\r\n";
    $postdata .= file_get_contents("documents\driver_application2.pdf");
    $postdata .= "\r\n\r\n";
    $postdata .= "--" . MULTIPART_BOUNDARY . "\r\n";
    $postdata .= "Content-Disposition: form-data; name=\"payload\"\r\n\r\n";
    $postdata .= $documentJSON;
```

```
        $postdata .= "\r\n\r\n";

        $postdata .= "--" . MULTIPART_BOUNDARY . "--\r\n";

            $status = $this->sendRequest($this->packageAppend . $packageId['id'] . '/documents', $documentJSON, $postdata, 'multipart/form-data; boundary=' . MULTIPART_BOUNDARY);

            return $status;

    }

    //Send Package

    public function buildSend($packageId)

    {

            $build = array(

                'status' => 'SENT'

            );

            $sendJSON = json_encode($build);

            $this->sendRequest($this->packageAppend . $packageId['id'], $sendJSON, NULL, 'application/json');

            return NULL;

    }

    //Sender signs consent and contract documents

    public function buildSign($packageId)

    {

            $build = array(

                'documents' => array(

                    array(
```

```
                    'id' => 'default-consent',
                    'name' => 'Electronic Disclosures and Signatures Consent'
                ) ,
                array(
                    'id' => 'contract',
                    'name' => 'Driver Application'
                )
            )
        );
        $signJSON = json_encode($build);
        $signatureAppend = $this->packageAppend . $packageId['id'] .
'/documents/signed_documents';
        $this->sendRequest($signatureAppend, $signJSON, NULL, 'application/json');
        return NULL;
    }
    //Get a session token
    public function buildToken($packageId)
    {
        $build = array(
            'packageId' => $packageId['id'],
            'signerId' => 'Signer1'
        );
        $tokenJSON = json_encode($build);
```

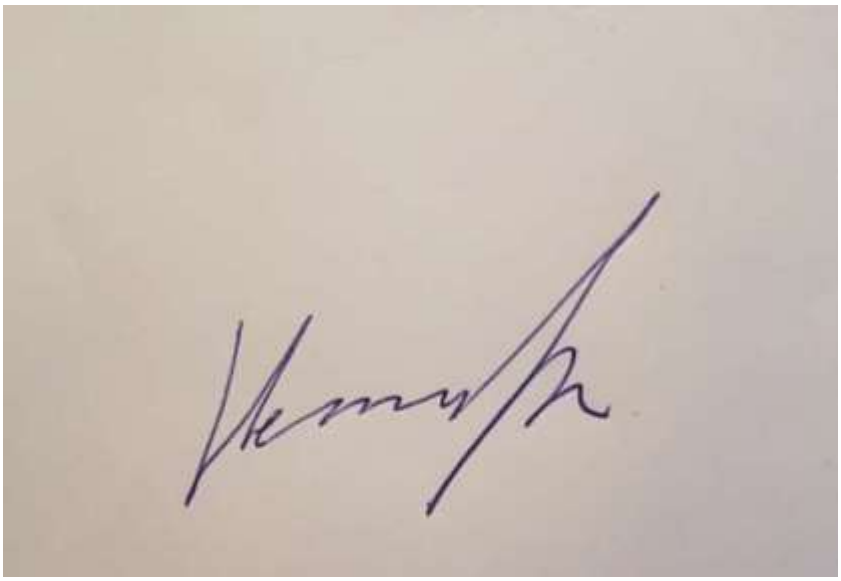
```
        $token = json_decode($this->sendRequest($this->tokenAppend, $tokenJSON, NULL,
'application/json'), true);
        return $token;
    }
    //cURL function to send requests to eSignLive
    private function sendRequest($type, $json, $document, $contentType)
    {
        if (is_null($document) && is_null($contentType))
        {
            $postfields = array(
                "payload" => $json
            );
        }
        else if (is_null($document) && !is_null($contentType))
        {
            $postfields = $json;
        }
        else
        {
            $postfields = $document;
        }
        $headerOptions = array(
            'Authorization: Basic ' . $this->key,
```

```
            'Accept:
application/json,application/zip,text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q
=0.8'

        );

        if (!is_null($contentType))

        {

            $headerOptions[] = "Content-Type: $contentType";

        }

        $ch = curl_init();

        curl_setopt($ch, CURLOPT_URL, $this->url . $type);

        curl_setopt($ch, CURLOPT_POST, true);

        curl_setopt($ch, CURLOPT_FAILONERROR, true);

        curl_setopt($ch, CURLOPT_FOLLOWLOCATION, true);

        curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);

        curl_setopt($ch, CURLOPT_SSL_VERIFYHOST, false);

        curl_setopt($ch, CURLOPT_SSL_VERIFYPEER, false);

        if (!is_null($postfields))

        {

            curl_setopt($ch, CURLOPT_POSTFIELDS, $postfields);

            if (!is_array($postfields))

            {

                $headerOptions[] = 'Content-Length: ' . strlen($postfields);

            }
```

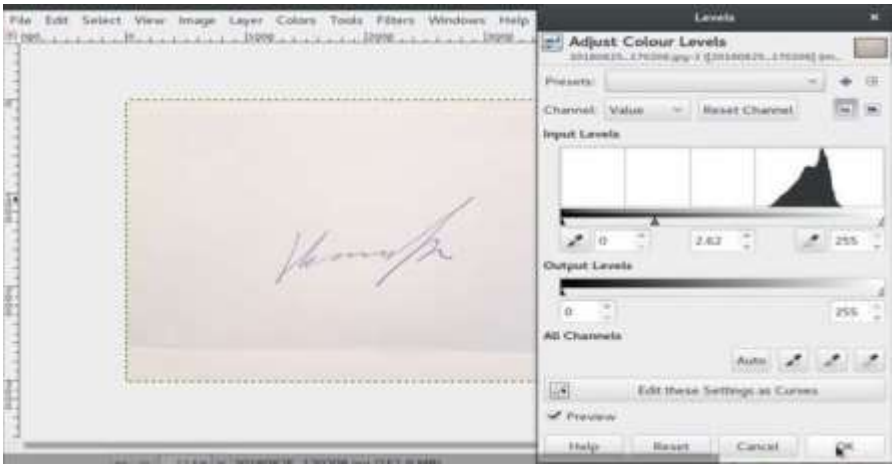
```
                }
                curl_setopt($ch, CURLOPT_HTTPHEADER, $headerOptions);
                $response = curl_exec($ch);
                $err = curl_error($ch);
                curl_close($ch);
                if ($err)
                {
                        return $err;
                }
                else
                {
                        return $response;
                };
        }
} ?>
```

**Adding a signature on a Digital document Like PDF or WORD File without Printing and ScanningStep 1: put your signature on a white paper.**

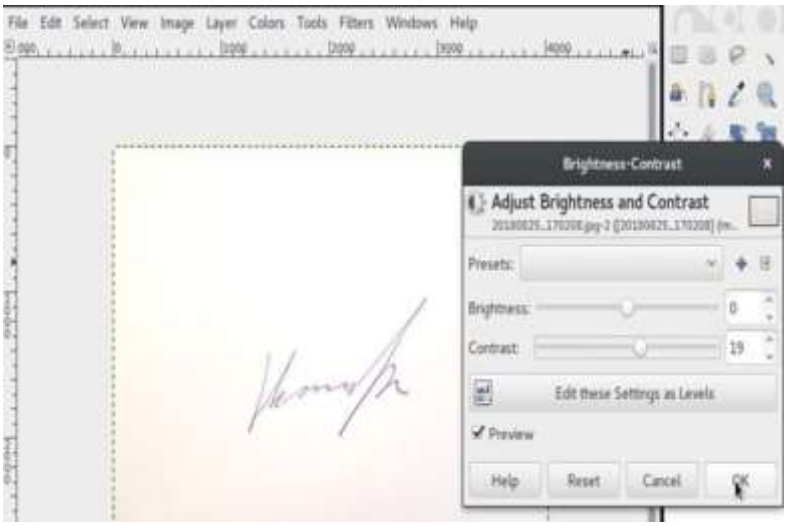
**Step 2: Take a Nice photo of your signature**



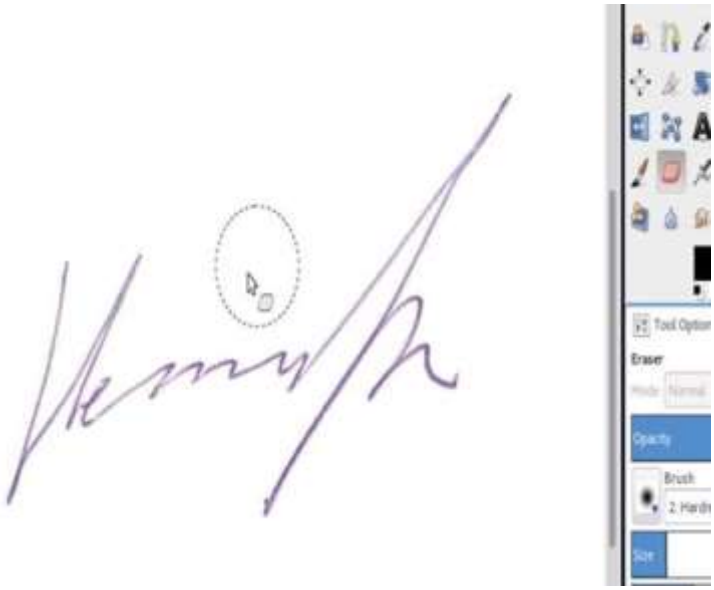
**Step 3: Open the photo with GIMP and adjust levels as shown in the image**



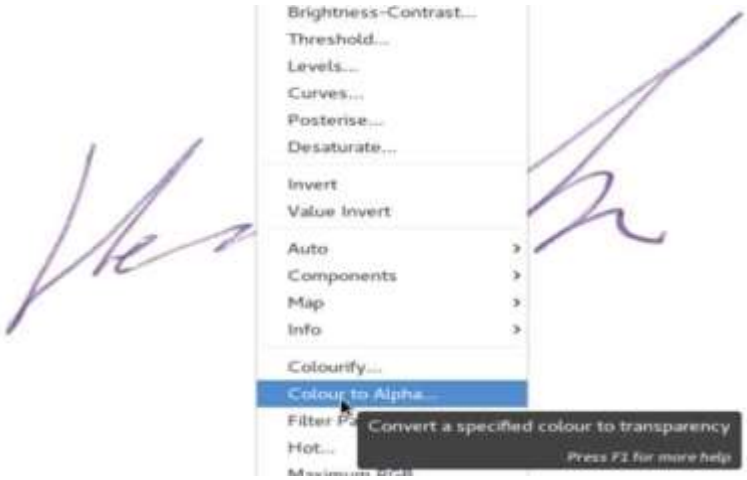

**Step 4: Adjust contrast as shown in the image**



**Step 5: Clean around your signature by using Eraser tool**




**Step 6: : Convert White Colour To Alpha**




Step 7: Save Image as PNG file

## Conclusion

Digital Signatures are the public-key primitives of messages authentication. Browsers are increasingly being relied upon to perform security sensitive operations because of the operations of SSL/TLS connections. A study of technically skilled android user, found that half of the internet users could not tell from the indictors provided by Android's browser, whether SSL/TLS was in effect or not [15]. This has significantly changed the use and consistency of the SSL/TLS indicators. This study proposes a technique to further enhance the integrity of Digital Signature by using a cryptographic mechanism coupled with secure coprocessor.

**Abbreviations:**

SSL Secure Socket Layers

TLS Transport Layer

Security

DSA Digital Signature

AlgorithmVPN virtual

Private Network HD

Hamming Distance

CA Certificate Authorithy

RAS Rivest-Shamir-

Adleman PKIPublic Key

Infrastructure

**Data Availability section**

Datasets related to this article can be found at[https://doi.org/10.1051/e3scorf/2021/244 /2023. TO BE LINKED TO DATA SET], hosted at [E3s web of conferences 244, 12023 (2021), EMMFT-2020. Anastasia Khrykova, Maria Bolsunovskaya, Svetlana Shirokova and Andrey Novopashenny] Datasets related to this article can be found at [https://doi.org/10.29099/ijar.v4i 1.171 [Untung Raharja, Sudaryono Sudrayono, Nuke Puji and Adam Faturahan (2020)].

[The 'Data Availability' section provides a convenient place to make the dataset citation, but authors may also choose to cite their data at any appropriate place within the main body of the manuscript]

**Acknowledgement**

This paper is dedicated to the memory of my mentor, professor Fatunla, who died, tragically in 1993. It is gratifying that I have been able to honour him with this work which substantially overlaps with his research interests. In addition, I am also indebted to Professor Akinwale of Federal University of Agriculture Abeokuta Nigeria, for numerous perceptive comments on various drafts of the manuscripts and for bringing to my attention gaps in my knowledge and holes in my logic.

Correspondence Author: Akhigbe-mudu

**FUNDING**

The author did not receive support from any organization for the submitted work. No funding was received to assist with the preparation of this manuscript. No funding was received for conducting this study. No funds, grants or other support was received.

Therefore, the author has no relevant financial interest to disclose. The author has no conflict of interests todeclare that are relevant to the content of its article.

**REFERENCES**

[1] **Alaa D. Alrehily, Asmaa F. Alotabi, Suzan B. Almutairy, Mashael S. Alqhtani, Jayarrakash Kar (2015);** Conventional and Improved Digital Signature scheme: A Comparative Study. Journal of Information Security. 2015, 6, 59–67. https://dx.doi.org/10.4236/jis.2015.61007

[2] **Anastasia Khrykova, Marina Bolsunovskaya, Svettlana Shirokova & Andrey Novopashenny (2021):** Implementation of Digital Signature technology to Improve the Interaction in Company. E3S Web of Conferences 244, 12023 (2021), EMMFT-2020. https://doi.org/10.1051/e3scorf/2021/244/2023

[3] **Dhanashree Toradmalle, jayabbhaskar Muthukuru, Sathyanarayana B.(2020):** Implementation of Provably Secure Digital Signature Scheme Based on Elliptic Curve. August 2020. Indian Journal of Computer Science and engineering 11(4):405-411. https://doi.org/10.21817.indjcse/2020/v11i4/201104299.

[4] **Fulvio Valenza, Matteo repetto, Stavos Shiaeles (2021):** Guest Editorial: special Issue on Novel Cuber-security Paradgins for Software Defined and Virtualized Systems. Journal of Computer networks volume 193, 5th July 2021,

108126.https://doi.org/10.1016/j.comnet.2021.108126.

[5] **Gamal E.I Selim, Ezz El-Din Hemdan, ahmed M. Shehatta, Nawal A. Elfish, El-Fishawy (2021)**: An Efficient Machine Learning Model for Maliciuos Activities Recognition in Water – Bsed Industrial Internet of Things. Journal of Security and Privacy, Volume 4, pp. 1-14, issue 3, May/june 2021.https://doi.org/10.1002/spy2.154

[6] **Gaurang Bansal, viray Chamola, Georges kaddoum, Md.Jalil Piran** (**2021**): Next generation Stock Exchange Recurrent Neural Learning Model for Distributed Ledger Transactions. Volume 193,5thJuly(2021) 107998. https://doi.org/10.1016/j.comnet.2021.107998

[7] Giannakoulias Michael Angelopoulous, Francosco Ramos (**2021);** SPEAR SIEM: A Security Information and Event Management System for the Smart Grid. Journal of Computer Networks. 193(2021) 108008. https://doi.org/10.1016/j.comnet.2021.108008 .

[8] **Kriszitian EGERSZEGI and Peter Erdosi** (**2003**): Problems in the Implementation of the Electronic Signature. Article in Periodica Polythnica Social & Management Sciences, November 2003, Volume 11, N0. 1, pp. 67-82.

[9] Monika Sharma (2020): AES (Advanced Encryption Standard) and RAS (Rivest-Shamir-adleman) Encryption on Digital Signature Document: A Literature Review. International Journal of Information technology and Business. Vol. 2, No.1, (2020) 26-29.

[10] **Panagiotis Radogou- Grammliks, Pangiotis Sariagiamidis, Eider Iturbe, Erkuden Rios (2021)**: Spear SIEM: A Security Information and Event Management for the Smart Grid. Journal of Computer Networks. Volume 193, July 2021, https;//doi.org/10.1016/j.comnet.2021.10800

[11] **Quzhao Zhou, Junging Yu, Dong Li (2021)**: A Dynamic and Lightweight Framework to Secure Source Addresses in the SDN-Based Networks. Journal of Computer Networks. Volume 193, 5th July 2021,108075. https://doi.org/10.1016/j.comnet.2021.108075.

[12] **Sandeep shalt, Pratyusa Manadhata and Loal Zomlot** (**2014);** The Operational role of security Information and event Management Systems. IEEE security and Privacy 12, 5(2014), 35- 41. https://dx.doi.org/10.1109/msp.2014.103

[13] **Untang Raharja, Sudaryona sudaryono, Nuke Puji & Adam Faturaham** (**2020**): COV-19: Digital Signature Impact on higher Education Motivation Performance. May 2020, International Journal of Artificial Intelligence Research 4(1). https://doi.org/10:29099/ijar.vol4i.171

[14] **Vielketh M. and pernul G**. (**2018):** A security Information and event Management Pattern.

12th Latin American Conference on Pattern Languages of Programs, November 2018, 12 pages.

[15] **Vanderson Martins Do Rosario, Maurico breternitz Jr., Edsun Borin (2021**): Efficiency and Scalability of Multi-Lane Capsule Networks (MLCN). Journal of Parallel and Distributed Computing.Volume 155,September 2021, Pages 63–73 https://doi.org/10.1016/j.jdc.2021.04.010