

# A Secure Cloud Based Digital Signature Application for IoT

Sahar A. El-Rahman, Benha University, Cairo, Egypt & Princess Nourah Bint Abdulrahman University, Riyadh, Saudi Arabia  
Daniyah Aldawsari, Princess Nourah Bint Abdulrahman University, Riyadh, Saudi Arabia  
Mona Aldosari, Princess Nourah Bint Abdulrahman University, Riyadh, Saudi Arabia  
Omairah Alrashed, Princess Nourah Bint Abdulrahman University, Riyadh, Saudi Arabia  
Ghadeer Alsubaie, Princess Nourah Bint Abdulrahman University, Riyadh, Saudi Arabia

## ABSTRACT

IoT (Internet of Things) is regarded as a diversified science and utilization with uncommon risks and opportunities of business. So, in this article, a digital signature mobile application (SignOn) is presented where, it provides a cloud based digital signature with a high security to sustain with the growth of IoT and the speed of the life. Different algorithms were utilized to accomplish the integrity of the documents, authenticate users with their unique signatures, and encrypt their documents in order to provide the best adopted solution for cloud-based signature in the field of IoT. Where, ECDSA (Elliptic Curve Digital Signature Algorithm) is utilized to ensure the message source, Hash function (SHA-512) is used to detect all information variations, and AES (Advanced Encryption Standard) is utilized for more security. SignOn is considered as a legal obligated way of signing contracts and documents, keeping the data in electronic form in a secure cloud environment and shortens the duration of the signing process. Whereas, it allows the user to sign electronic documents and then, the verifier can validate the produced signature.

## KEYWORDS

Cryptosystem, Data Integrity, Data Privacy, Digital Signature, Elliptic Curve Digital Signature, Elliptic Curves, Encryption, Hash Function, Hashing, Information Security, IoT, Public Key Encryption, Security

## 1. INTRODUCTION

The vast evolution of the communication and networks in the last two decades changed the way our world works drastically (Srivastava, 2013). Using signature in paper transactions has been spread over the last years as a way of gaining assurance of the identity and authority of the parties involved in a contract (Sumroy and Sherrard, 2012). Electronic communication methods such as Email and the Internet make it potential to execute contracts. The most important formal requirements are the signature; therefore, a contract requiring a signature need to be able to do so electronically (Laborde, 2010). Digital Signature that is defined as an electronic data that is logically associated with or

DOI: 10.4018/IJESMA.2018070103

Copyright © 2018, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.

attached to other data in electronic form and that is considered as an authentication method (United Nations Publication, 2009). Also, a digital signature is an arithmetical model used to validate the authenticity and integrity of a digital document, message, or software (Martoni and Palmirani, 2013). Including assurance as to its authenticity even if later the verifying party or signer tries to deny (Chiranth and Shashikala, 2012). Because of the Internet's security protection holes, several of the digital community members explored how to allocate risk if a security breach were to occur. From this investigation, three main issues were identified as security risks: integrity, nonrepudiation, and authentication (Stern, 2001).

Digital signature should provide three main services: Authentication which is linking the creator of the information, Integrity, which means easily detecting any variations to the information provided and Non-repudiation for making sure of the satisfaction (from a legal perspective) about where the electronic signature is coming from (Wang, 2014). Digital signature laws had three generations, which started appearing since 1995 (Blythe, 2007; Embrogno, 2012). Moreover, mobile signatures are predicted to get a great future. To grant signatory mobility apart from a fixed, secure desktop workstation that has a trusted, personal signing tool (Rossnagel and Royer, 2005), and because of how smart devices are taking over PC's a need for signature applications emerge and many applications started to hit the different markets (Google Play, App Store and windows store).

## **2. CLOUD COMPUTING**

Now that almost all companies that used to rely on paperwork are moving to cloud services due to the powerful benefits those services provide this issue is becoming even more critical (Srinivasan, 2014). Whereas, cloud computing becomes a novel paradigm to offer computing as a utility. It is considered as a scheme to enable convenient, ubiquitous, on demand access to the network to share a configurable compute resources (Kinastowski, 2013; Mella and Grance, 2011). So, when two people are making a deal and they form an agreement to sign a document to accomplish the deal and to prove that neither party can oppose the deal later. Now what if one party used a fake signature and is planning to fool the other party? Digital signature should provide the non-repudiation needed (Azizi, 2011). A cloud based digital signature is considered as a paradigm for proper, reliable, secure infrastructure, with flexible access to the network that implements digital signature cryptographic processes. The main variation between a cloud based digital signature model and a standard one is the first comprises a network data interchange between signing-enabled cloud environment and the signer, but the second works in the closed environment of a plugged-in a dedicated device (card reader and microchip card) and PC (Kinastowski, 2013; Shakil et al., 2017). The most notable feature of cloud computing technology that related to the IoT is the storage over internet, so the document availability becomes 24/7 over the cloud and the users can digitally sign documents and request the digital signature of other individuals (AlZain et al., 2015; Stergiou et al., 2018).

## **3. INTERNET OF THINGS (IOT)**

Recently, IoT has been given a significant research awareness. IoT is considered the networking future. Where, IoT is a novel paradigm that involves everyday physical world entities by enabling exchange among them (Koppula and Muthukuru, 2016). IoT utilization in various applications is predicted to rise quickly in the forthcoming years. IoT permits billions of services, peoples, and devices to connect each other and interchange the information. So, the increasing of IoT objects utilization, the networks of IoT are subject to different security attacks. The privacy protocols, and efficient security deployment in IoT networks is highly required to ensure authentication, integrity, access control, and confidentiality, among others (AbdurRazzaq et al., 2017; Elmisery et al., 2017; Singh and Singh, 2015). The combination of IoT with cloud computing will provide the convenience of the proposed application (Stergiou et al., 2018).

#### 4. SIGNATURE PATTERNS

A signature pattern is composed of triple probabilistic polynomial time approaches (*Gen*, *Sign*, *Verify*); where *Gen* for key generation, *Sign* for a signature creation, and *Verify* for verification of the signature. The signature scheme should fulfill the following (Ali, 2015):

- The key generation approach takes no input and creates a key pair ( $pk$ ,  $sk$ ) as an output, where  $pk$  is the public key and  $sk$  is the matching private key, respectively. The operation of executing this approach can be written as:  $Gen \leftarrow (pk, sk)$ .
- The message  $M$  and  $sk$  are the inputs to signature creation approach and  $\sigma$  (a signature) will be produced as an output. The signature creation algorithm may be stateful, randomized or even both. If the algorithm generates its output randomly, there might be several correct signatures mapped to one message. When the algorithm is stateful, for example the signature algorithm makes the accessing use as a global variable, counter, maintained by the sender to create the signature. The output  $\sigma$  is written as:  $\sigma \leftarrow Sign(sk, M)$  or  $\sigma \leftarrow Sign_{sk}(M)$ .
- $pk$ ,  $M$ , and  $\sigma$  are the inputs to the signature verification approach and return a bit  $b$  as an output, where the signature is valid if  $b = 1$  and  $b = 0$  means the signature is not valid. It is written as  $b \leftarrow Verify_{pk}(M, \sigma)$  or  $b \leftarrow Verify(pk, M, \sigma)$  (Bellare and Rogaway, 2005).

#### 5. ECDSA SIGNATURE CREATION ALGORITHM

The signature algorithms are called *ecdsa-Fp* (ECDSA based on a group  $E(Fp)$ ) and *ecdsa-F<sub>2</sub><sup>m</sup>* (ECDSA based on a group  $E(F2^m)$ ). The security algorithms based on the computing difficulty of EC discrete logarithms (ETSI, 2011; Khan, 2015).

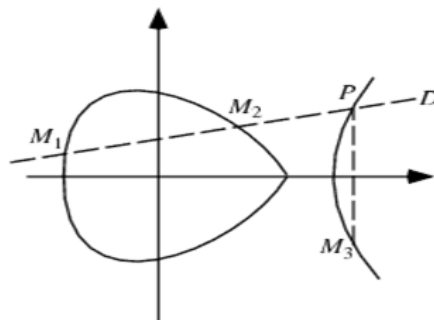
##### 5.1. EC Operations Over Finite Fields

The main two elliptic curve operations to achieve the main operation which are:

- **The Addition of Points:** It is adding process of  $M_1$  and  $M_2$  points on EC to get the result  $M_3$  on the same EC as indicated in Figure 1. Consider two points  $M_1$  and  $M_2$  on EC and  $M_2 \neq -M_1$  hence a the line drawn between  $M_1$  and  $M_2$  exactly intersects EC at one more point and let's call it P. The point P reflection relates to x-axis represents  $M_3$  that is the result of addition. So, on EC the Equation (1) is:

$$M_3 = M_1 + M_2 \quad (1)$$

Figure 1. Point Addition (Jain, 2012)



From Equation (1), the line over  $M_2 = -M_1$  point will cross the infinity  $O$ . So,  $M_1 + (-M_1) = O$ , where  $(-M_1)$  is the reflection of that point related to the x-axis (Brow, 2010; Khalique et al., 2010).

- **Doubling of the Point:** The point  $J$  is doubling on EC to get another  $L$  point on the selfsame EC, i.e. to find  $L$  as shown in Equation (2):

$$L = 2J \quad (2)$$

Consider  $J$  on EC as indicated in Figure 2. Where if the y component of  $J$  isn't 0, then the tangent line at  $J$  will exactly cross EC at another  $-L$  point. The reflection of this point  $-L$  related to x-axis represents  $L$ , that is the result of doubling, i.e.,  $2J = L$ . If the y component of  $J$  is 0, then the tangent line at  $J$  will cross infinity  $O$ . So, if  $y_J = 0$  then  $O = 2J$ . Figure 2 indicates the doubling of point (Botes and Penzhorn, 1994; Koppula and Muthukuru, 2016; Jain, 2012).

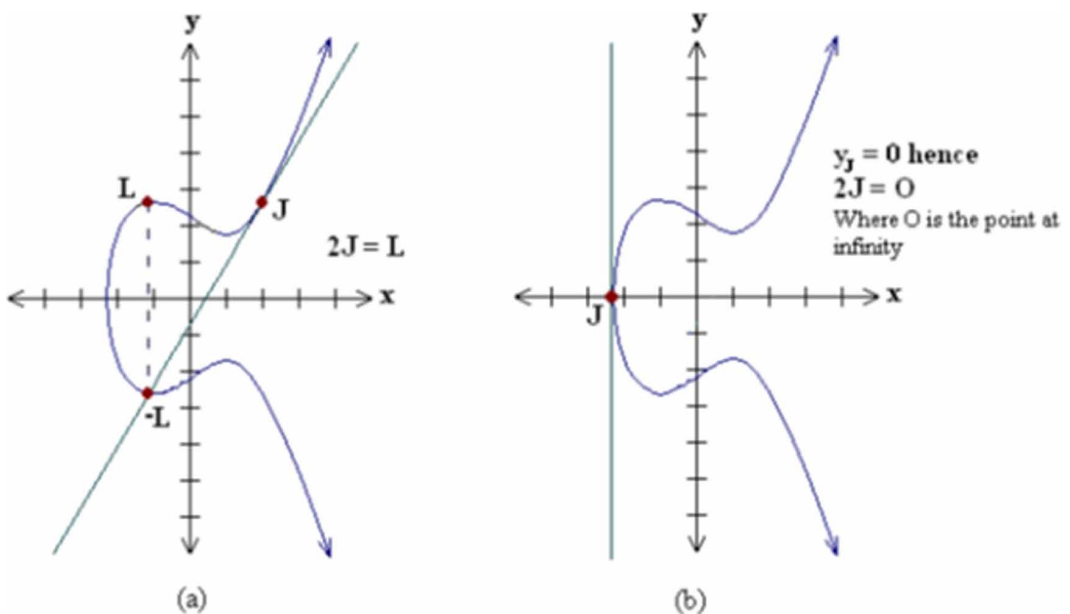
## 5.2. Domain Parameters of ECDSA

ECDSA needs the key pairs that utilized for the verification of digital signature and generation be done related to a specific EC domain parameter set. Whereas, these parameters might be common to the public and to a group of users and can stay fixed for an expanded time interval (FIPS 186-3, 2010; Tewari and Gupta, 2017).

A key pair of digital signature  $(d, Q)$  that represent (*private, Public*), is created with a specific domain parameter set. From given a parameter set  $(q, FR, a, b, SEED, G, n, h)$ , an EC key pair  $(d, Q)$  can be generated. Two algorithms are implemented to create the EC key pair  $(d, Q)$ . Before creating EC key pair, the parameters authentic copy  $(q, FR, a, b, SEED, G, n, h)$  will be generated. ECDSA Domain parameters are (FIPS 186-3, 2010; Koppula and Muthukuru, 2016):

- $q$  is the implied area size.
- $FR$  is a indicator of field representative, with the value of *NULL*.

Figure 2. Point Doubling (Jain, 2012)



- $a$  and  $b$  are the EC parameters.
- *SEED*: utilized to create parameter  $b$ .
- $G = (xG, yG)$  is the base point that lies on the curve.
- $n$  is the order of base point  $G$ .
- $h$  is the EC group order divided by  $n$  of  $G$ .

#### 5.2.1. EC Key Pair Generation based on Extra Random Bits (FIPS 186-3, 2010; Koppula and Muthukuru, 2016)

In this approach, the required bits (64 bits) are more than are required for  $d$ , hence  $bi$  is generated by the modular reduction in step 6 is neglected.

- **Input:**  $(q, FR, a, b, SEED, G, n, h)$ .
- **Output:**
  1. **Status:** the returned status from the key pair generator procedures are *ERROR* or *SUCCESS*.
  2.  **$(d, Q)$ :** the created key pairs. If an error is faced with the generator procedure, the pair represented by (invalid  $d$ , invalid  $Q$ ), must be returned. Where  $Q$  is an EC point and the created  $d$  is as an integer in  $[1, n - 1]$ .
- **Procedures:**
  1. Check that  $N$  is valid, where  $N = len(n)$ , and  $N = 256$  or  $N = 384$  (the only Suite B valid lengths).
  2. (*Invalid  $d$ , Invalid  $Q$* ) and *ERROR* must be returned, If  $N$  is invalid.
  3. Put the required strength of security to be related to  $N$ , that is 192 if utilizing  $P-384$  or 128 if utilizing  $P-256$ .
  4. Get a returned string ( $N + 64$  bits) from an *RBG* with a required strength of security or more. *ERROR* and (*Invalid  $d$ , Invalid  $Q$* ) will be returned, if *RGB* return an error indicator.
  5. The returned bits are converted to  $c$  that is a non-negative integer.
  6.  $d = (c \bmod (n - 1)) + 1$  is set.
  7. Calculate  $Q = dG$  utilizing the multiplication of EC scalar.
  8. Return *SUCCESS* and  $(d, Q)$ .

### 5.3. Security of ECDSA

The ECDSA security goal is to be existing unforgeable versus selected message attacks. The objective of an enemy who dispatches this type of attacks on a known entity  $A$  is to get a working signature of one message  $m$ , after acquiring  $A$ 's signature on a messages combination (without  $m$ ) of the attacker's choice (Menezes et al., 2001). Some advancements on verifying ECDSA security, albeit in robust theoretical models. Minor variants of DSA and ECDSA (but not ECDSA itself) verified to be existing enduring versus selected message attacks (Pointcheval and Stern, 1996), under the suspicions that the DL issue is difficult and the hash function used is as a randomly created function. Particularly, their findings indicate that each effective attack that doesn't utilize the frame of the known hash function in the pattern has to cause breach ECDLP (*Elliptic Curve Discrete Logarithm Problem*). ECDSA has newly been proved as a secure by Brown (2000) that was assumed that the implied group is a general one and the hash function used is a collision resistance. Explicitly, Brown's findings indicate that any effective attack must detect impacts in the hash function and this attack doesn't utilize the frame of the EC group (Menezes et al., 2001).

### 5.4. ECDSA Signature Generation (Stadick, 2006; Subramaniam et al., 2012)

The parameters that will be utilized to create the signature and will be shared with both sender and receiver are indicated in Table 1.

**Table 1. EC Parameters**

Definition	Parameter
EC field and equation utilized	<i>Curve</i>
A prime order base point on the curve	<i>G</i>
EC base point ( <i>G</i> ) order	<i>n</i>

A sender BOB utilizes  $d_A$  (his private key) to sign  $M$  and send it to ALICE. The steps are:

1. Compute the hash for the message without a signature and let's call it  $h$

$h = \text{HASH}(\text{message})$ , where *HASH* is *SHA-512*.

2. Select from the interval  $[1, n-1]$  a random integer  $k$ .
3. Calculate  $r = x1 \pmod n$ , If  $r = 0$ , then go to step 2, where  $(x1, y1) = k * G$ .
4. Compute  $s = k^{-1}(h + d_A r) \pmod n$ , If  $s = 0$ , then go to step 2
5. Then the signature will be  $(r, s)$ .

### 5.5. ECDSA Signature Verification (Koppula and Muthukuru, 2016; Stadick, 2006; Subramaniam et al., 2012)

ALICE receives message  $m$  and authenticates and utilizing BOB's  $Q_A$  to verify the signature, the steps are:

1. Test if  $s$  and  $r$  are in  $[1, n-1]$  as integers, then the signature is valid, else it is *invalid*.
2. Compute  $h = \text{HASH}(m)$ , where *HASH* is the identical hashing algorithm utilized during signature creation.
3. Compute  $w = s^{-1} \pmod n$ .
4. Calculate  $u1 = hw \pmod n$  and  $u2 = rw \pmod n$ .
5. Calculate  $(x1, y1) = u1G + u2Q_A$
6. The signature is valid, if  $x1 = r \pmod n$ , otherwise it is *invalid*.

### 5.6. Comparison Between Signature Creation Algorithms

The signature algorithms comparison is indicated in Table 2 (EESSISG, 2001; Kumar and Koul, 2011; Zongqu, 2010).

## 6. AES ENCRYPTION ALGORITHM

AES is considered as a symmetric fixed cipher which was proposed by J. Daemen & V. Rijmen to the US NIST (*National Institute of Standards and Technology*) and it was chosen as *US federal standard* (Berent, 2003; FIPS PUB 197, 2001). Where, in the late 1990s, NIST managed a context for developing a replacing algorithm for DES. In 2001, the winner was announced to be the Rijndael algorithm, intended to be the novel AES. Rijndael combines the SPN model by adding the operations of Galois field to any round. AES is not utilizing a Feistel network, in contrast to its predecessor. The strength and design of all AES key lengths (256, 192 and 128) are adequate to save confidential data up to the SECRET level. Topmost SECRET data will request utilizing either 256 or 192 key lengths (FIPS PUB 197, 2001; Stallings, 2011).

**Table 2. The Signature Algorithms Comparison (EESSISG, 2001; Kumar and Koul, 2011; Zongqu, 2010)**

Factors	RSA	DSA	ECDSA
Security	High	High	High
Complexity	Integer Factorization	Discrete Logarithm	Discrete Logarithm
Domain	PC, Laptops, Super computers	Light-weighted devices	Light-weighted devices
Key Length (80-bits)	1024	1024	160
Key Generator	Fast	Fast	Faster
Execution Time	Slow	Slow	Fast
Verification Time	Fast	Slow	Slow
Signature Time	Fast	Fast	Fast
Data Size	Smaller	Large	Large

AES is utilized to save critical unclassified data utilized by federal agencies and departments. Also, this standard may be utilized and adopted by commercial and private organizations (non-federal government organizations) to provide the required security (FIPS PUB 197, 2001). AES is an iterated algorithm which means multiple rounds (10, 12, 14) consisting of 4 numbers of procedures are carried out respectively. Add to that key expansion operation which is done in the algorithm start. Performing AES encryption is done by the integration of four functions which uses two main cipher concepts permutation and transportation these functions are: add a round key, shift row, a byte sub, and mix column and it is called in almost each round (see Figure 3) (Berent, 2003; Stallings, 2011).

### 6.1. AES, DES and 3DES Comparison

All of AES, DES and 3DES are widely used ciphers. Choosing any one of them is dependent on the needs. The difference between them in terms of performance and security is indicated in Table 3 (Alanazi et al., 2010; Chiranth and Shashikala, 2012; Siwik and Mozgowoj, 2015).

## 7. HASH FUNCTIONS

Hash function according to (FIPS publications) is a function that maps an arbitrary length bit string to a constant length string. It should fulfill the characteristics:

- **One Way:** Where it is preventing any way to discover the input which is mapped to any new pre-defined output.
- **Collision Resistance:** It is preventing any way to get two different inputs which are mapped to the output itself (Locke, 2009).
- **Hash Functions:** Are usually utilized with other approaches of cryptographic, like a digital signature approach. Secure hash algorithms: *SHA-1*, *RIPEMD-160*, and *SHA-2* family are iterative. After processing the message, the hash function produces a “message digest”. Message digests help in finding out the message integrity. Different message digests result is usually because of the change to the message will. So, this characteristic is beneficial in generating and verifying of the digital signature (Bryson, 2015).

In Figure 4, the message enters the hash function as an input to produce the distinct message digest in the sender side. The message and its digest transmit to the receiver. Where, the receiving

Figure 3. AES Algorithm. (a) AES Block Diagram; (b) AES Structure (Stallings, 2011)

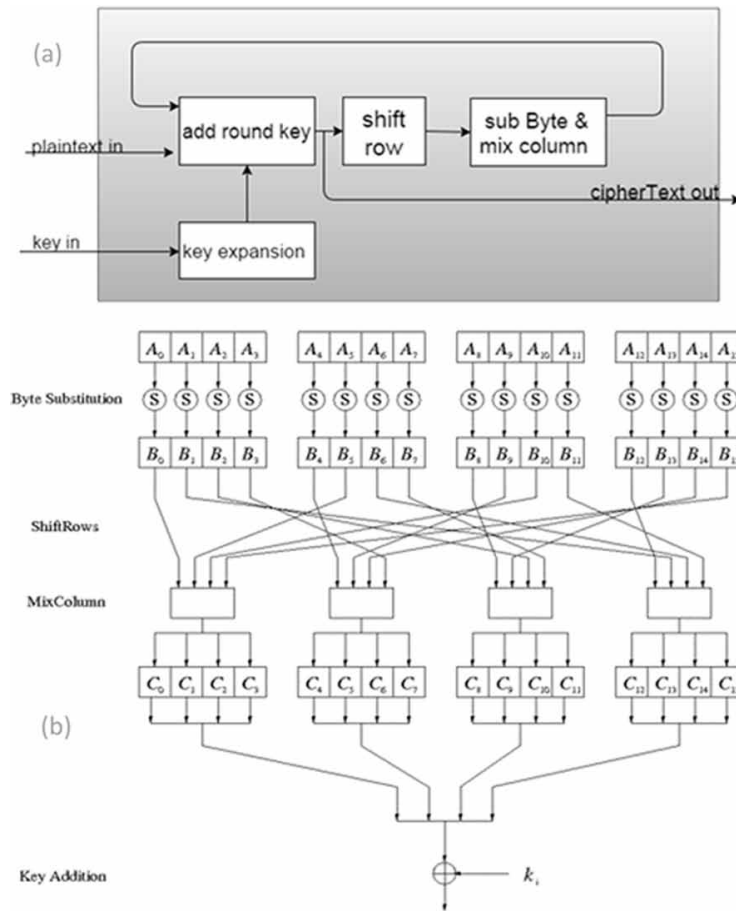
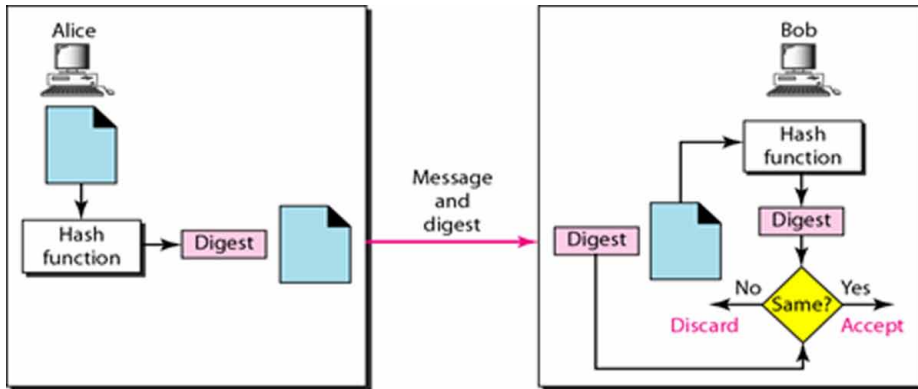


Table 3. Comparison between Encryption Algorithm (Alanazi et al., 2010; Chiranth and Shashikala, 2012)

3DES	DES	AES	Factors
(k1, k2 and k3) 168 bits (k1 and k2 is same) 112 bits	56 bits	128,192, or 256 bits	<b>Key length</b>
Symmetric block cipher	Symmetric block cipher	Symmetric block cipher	<b>Cipher Type</b>
64 bits	64 bits	128,192,256 bits	<b>Block size</b>
1978	1977	2000	<b>Developed</b>
Vulnerable to differential, attack of brute force	Vulnerable to linear cryptanalysis and differential, tables of weak substitution	Strong versus truncated differential, differential, linear, interpolation and square attacks	<b>Cryptanalysis resistance</b>
$2^{112}$ or $2^{168}$	$2^{56}$	$2^{128}$ , $2^{192}$ or $2^{256}$	<b>Possible keys</b>
$95^{14}$ or $95^{21}$	$95^7$	$95^{16}$ , $95^{24}$ or $95^{32}$	<b>Possible ASCII printable character keys</b>
For a 112-bit key: the days are 800	For a 56-bit key: the days are 400	For a 128-bit key: the years are $5 \times 10^{21}$	<b>Possible ASCII printable character keys</b>



Figure 4. Hash Function to Check the Integrity (Forouzan, 2007)



message will enter to hash function to create a new message digest. The system compares the new message digest with the previous one. If they are the same, the message will be accepted. If not, the message will be discarded.

## 7.1. SHA-2

In 2001, new hashing approaches were introduced. This newly introduced collection of hashing approaches denoted by Secure Hash Algorithms 2 (*SHA-2*), utilize bigger digest messages, that make them further resistant to potential strikes and granting them to be utilized with big data blocks, up to 2128 bits, e.g. as *SHA512*. The *SHA-2* hashing approach is similar to the *SHA512*, *SHA384*, *SHA256*, and *SHA224*, the variation is only in the left and right operands size, the size of the final digest message, the initialization vectors (Chaves et al., 2006).

### 7.1.1. SHA-256

*SHA-256* generates a 256 bit message digest, which depend on the input, formulated from many 512 bit blocks. The input message to be hashed is first (SHA256-384-512, 2001):

1. The message is padded in a manner which the output length is a multiple of 512 bits.
2. Parsing in the message of 512 bit blocks  $M(1)$ ,  $M(2)$ , ...,  $M(N)$ .

### 7.1.2. SHA-512

It is a secure hash algorithm that have up to  $2^{128}$  a message size as an input and it is separated into blocks each of 1024 bits to generate a message digest of 512-bits. Hash function checks the integration of the message, so it supplies the protection against accidental message modification. It is working with the digital signature to secure the documents.

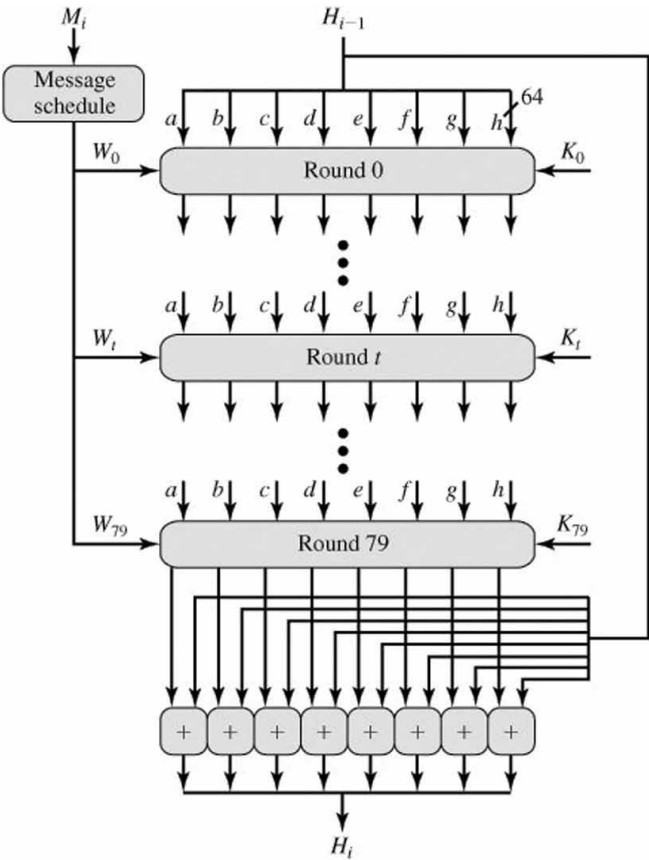
Its calculation is identical to that of *SHA256*, the variation is in the operand size, which is nearly 64 bits instead of 32 bits as *SHA256*. The message digest size, that has double the size being composed of 512-bits (see Figure 5) (Chaves et al., 2006; Stallings, 2011).

**Input:** The document to be signed.

**Output:** The document message digest that is a unique for the distinct message.

**Procedure:** The message is padded in a manner which the output length is a multiple of 1024 bits. Then, parsing in the message of 1024-bit blocks  $M(1)$ ,  $M(2)$ , ...,  $M(N)$  (SHA256-384-512, 2001). Detailed discussions about *SHA-512* found in (FIPS 180-4 (Bryson, 2015)).

Figure 5. SHA-512 Processing on Each Block (Stallings, 2011)



7.2. SHA-1, RIPEMD, SHA-256, SHA-512 and SHA-512/256 Comparison

The comparison between hash functions is indicated in Table 4 (Chaves et al., 2006; Mendel et al., 2006; SHA256-384-512, 2001).

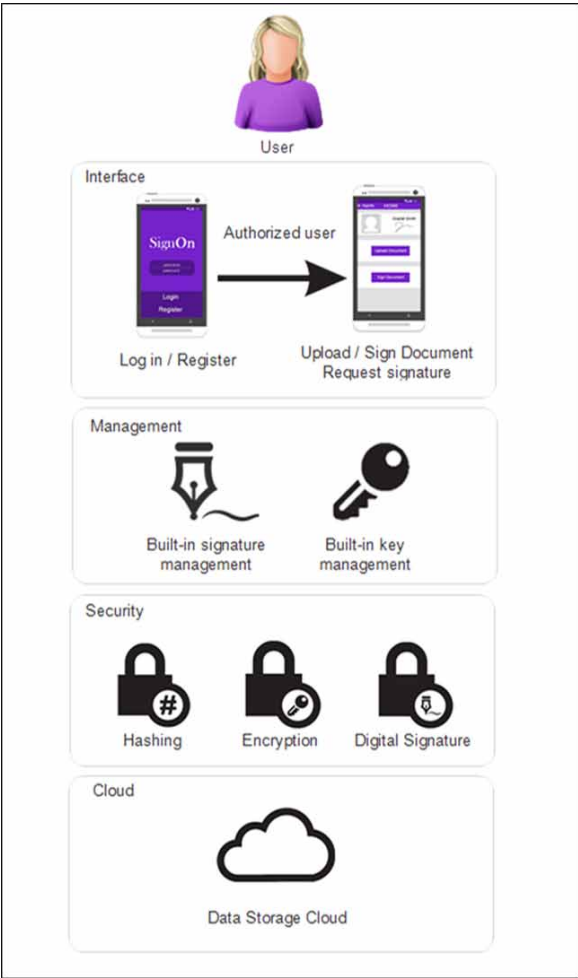
8. SIGNON APPLICATION METHODOLOGY

SignOn application has an architecture of layered style (see Figure 6). The first layer is for the operations that deal with the access of the user: login and register, and the operations that the

Table 4. Comparison between Hash Functions (Chaves et al., 2006; Mendel et al., 2006; SHA256-384-512, 2001)

Hash Function	Message Size	Message Blocks Size	Message Digest Size
SHA-1	$< 2^{64}$	512-bit	160-bit
RIPEMD-160	$< 2^{64}$	512-bit	160-bit
SHA-256	$< 2^{128}$	512-bit	256-bit
SHA-512	$< 2^{128}$	1024-bit	512-bit
SHA-512/256	$< 2^{128}$	1024-bit	256-bit

Figure 6. SignOn Application Architecture

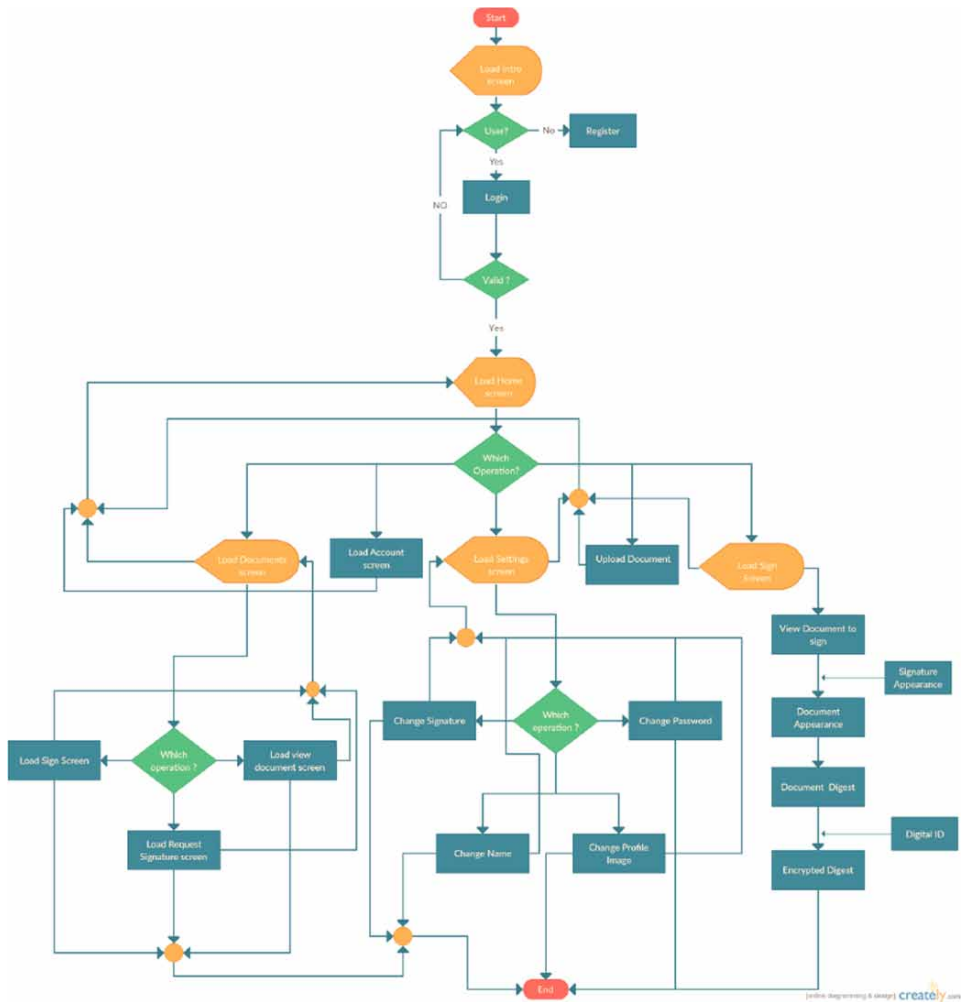


authorized user can do: upload, sign, and view documents, request signature to another user, change the setting of the account, etc. The second layer is for the security where the hashing document, and secure the message digest with the digital signature keys and encrypt message. The final layer is in the cloud for data storage where the document and the database saved. Whereas, SignOn is designed as explained in Figure 7.

The proposed application was implemented on windows 8 as OS using android studio 1.5.1 and JDK 7 Whereas, it provides a cloud based digital signature with a high security to follow the growth of IoT and the speed of the life. The main concepts to accomplish this service are integrity, authentication, and encryption. ECDSA is utilized to ensure about Authentication, SHA-512 is utilized to fulfill integrity, that easily detecting the change of the information provided from and to resist against forgery attack and AES is utilized for more security.

The proposed application uses a combination of FTP server and Firebase cloud to retrieve and store the user's data and documents. Whereas, Firebase cloud does not support the document upload capability, so this problem was solved with the proposed application by uploading the documents to FTP server then storing the link to that document on the cloud. Also, when developing for Android

Figure 7. SignOn Flowchart



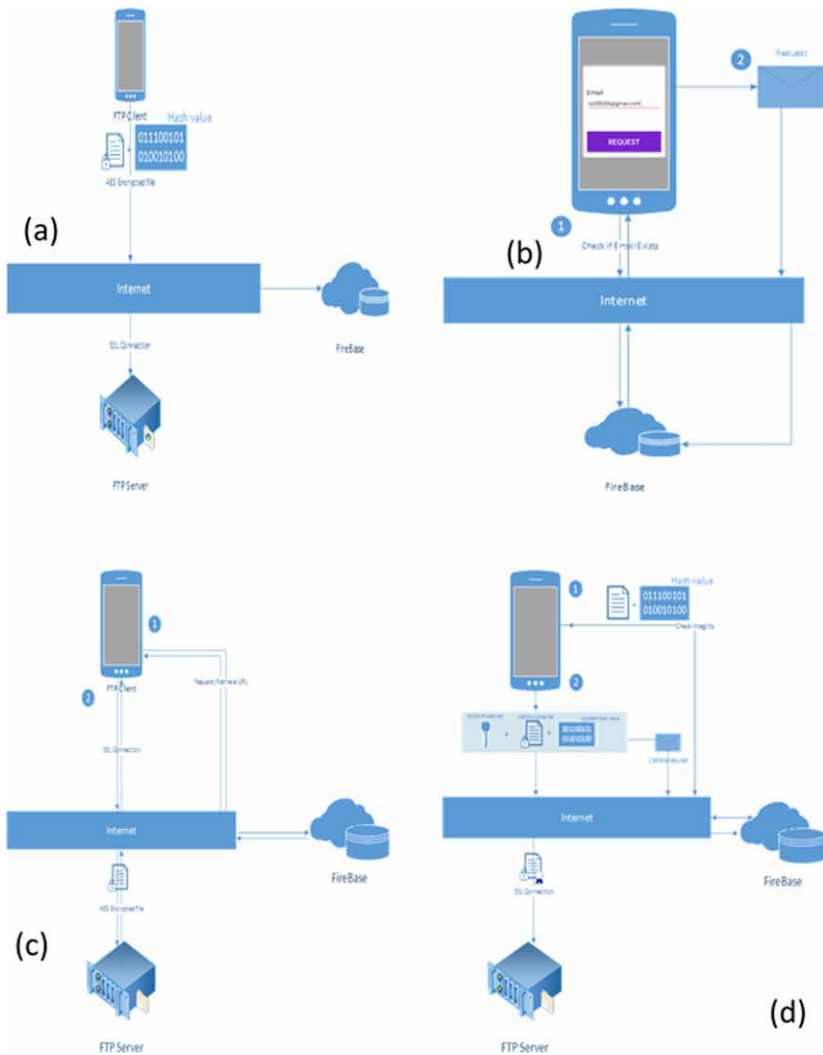
API level under 21, no build-in PDF rendering is provided so, this problem was solved by setting the minimum API of the application to 21.

Documents is hashed by SHA512, then on the client side, they are encrypted via AES encryption as indicated in Figure 8a. Signature requests can be sent by providing the signers' Emails. Users can view all their requests in the pending list as indicated in Figure 8b. Documents are downloaded from the server after obtaining the URL from the cloud, afterward on the client side, they're decrypted as indicated in Figure 8c. During the signing operation, the hash values are checked, then the ECDSA signature is generated and the hash value is updated. The updated document is uploaded to the server as indicated in Figure 8d.

### 8.1. SignOn I/O Screens

The Proposed Application I/O Screens as indicated in Figure 9, when the application is launched, a login screen will appear, then the user can login or register. After the login screen, the home screen will appear. So, the user can Upload document, capture signature, or Navigate through the application by using the navigation bar. When, the user selects "documents", lists of documents will appear: my

Figure 8. The proposed SignOn application

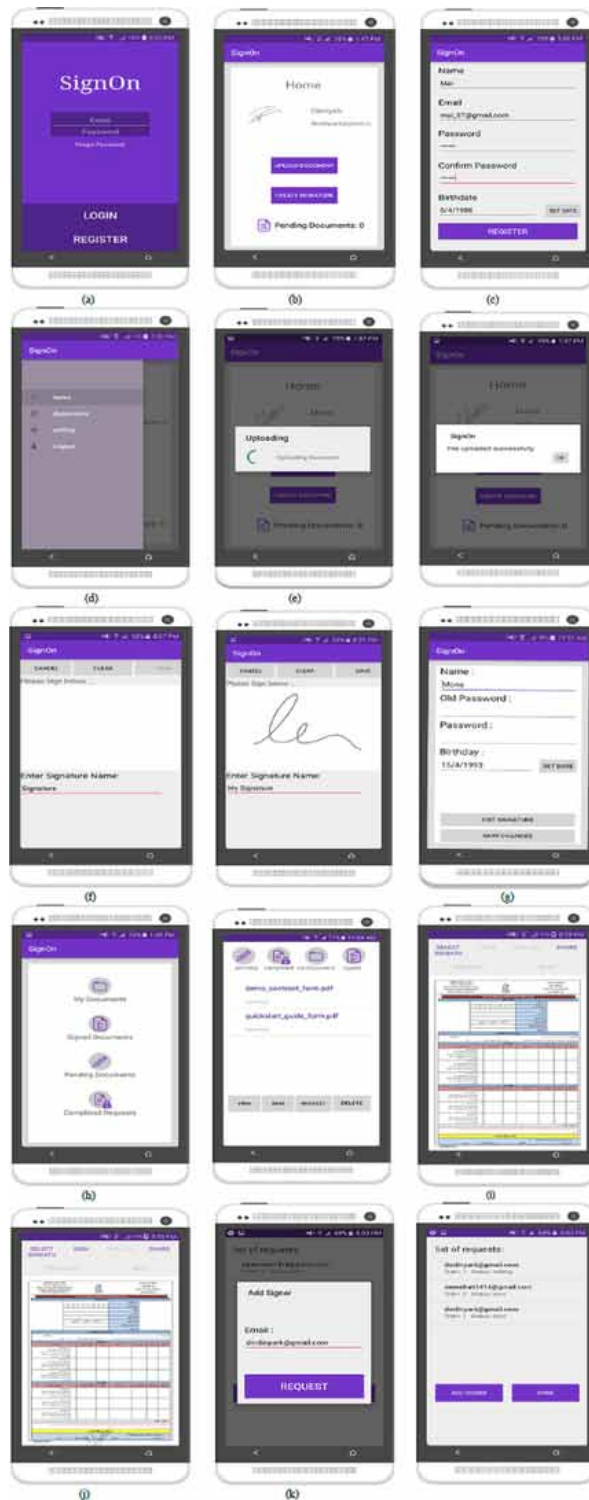


documents, signed documents, pending documents and completed request. After that, the user can select a document from my documents list he can View, Sign, Request, or Delete.

## 9. RESULTS AND DISCUSSIONS

SignOn is Online Signature mobile application that enables any authorized user to sign documents. Any an organization administrators, employees, students, suppliers, or clients need to authenticate invoices, contracts, tax forms, and other documents. For security, every user has a username and password to login and subsequently, the user can send and receive documents to be signed, create their own digital signature and store these documents in a secure cloud. So, the main functional capabilities of SignOn application are tested in all phases of the testing cycle. The results of these tests were a good way for ensures that all operations in this application is working as expected. All the tests were used to check the reliability of the application and fix any configuration, component parameters for reliable performance. Although, the testing of the mobile application is not the same

Figure 9. SignOn I/O Screen: (a) User Login Screen; (b) User home screen; (c) User Register Screen; (d) Navigation bar screen; (e) Upload document screen; (f) Capture signature screen; (g) User setting Screen; (h) User Documents Screen; (i) Before user signs the document screen; (j) After signing the document screen; (k) User request screen


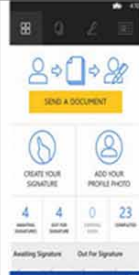




and more complicated than the classic web application testing. SignOn application and three of the most competitor applications comparison as explained in Table 5, to get a clear vision of the domain and do utilize of the applications market.

## 10. CONCLUSION

In this paper, a proposed cloud based digital signature application (SignOn) aims to provide a legal binding way of signing contract and documents while keeping the user's data electronic and secure. Whereas, the different documents can be stored and shared electronically, so, the signing procedure is fastened and the paperwork amount is decreased. Whereas, the data complexity validation is uniquely and exclusively assigned to somebody, making it possible to identify that person, its objective is making access secure. Whereas, SignOn is an Android mobile application with cloud and FTP storage. It enables the users to sign documents via

Table 5. SignOn and Similar Applications Comparison

	SignOn	DocuSign (www.docusign.com)	Signing Hub (www.signinghub.com)	Right Signature (www.rightsignature.com)
<b>Problem solved</b>	The need of signing on paper which might lead to delays in the workflow - Lack of security - Lack of authenticity - Lack of integrity	The need of signing on paper which might lead to delays in the workflow - Lack of security - Lack of authenticity - Lack of integrity	The need of signing on paper which might lead to delays in the workflow - Lack of security - Lack of authenticity - Lack of integrity	The need of signing on paper which might lead to delays in the workflow - Lack of security - Lack of authenticity - Lack of integrity
<b>User authentication</b>	- Email-based authentication. - Firebase authentication.	- Email-Based Authentication - Access Code Authentication - SMS Authentication ...etc.	- Email Authentication. - External Identity Providers (OAuth /OpenID/ social media) ...etc.	- Biometric Authentication - Webcam Photo Authentication - Multi-Factor Authentication
<b>Encryption Algorithm</b>	AES (256-bit)	AES (128-bit)	AES (256 bit)	No encryption
<b>Advanced Electronic Signatures</b>	Yes	Yes	Yes	No
<b>Software &amp; hardware</b>	Smartphone android	PC, Android, IOS, Web-based, Windows	PC, Android, IOS, Web-based.	PC, Android, IOS, Web-based.
<b>User features</b>	- Signing sequencing - Upload document from cloud - Document statuses viewing (my documents / signed /pending)	- Upload document form cloud - Contact list - Signing sequencing - The user selects the type of authentication (none /access code /phone / knowledge based)	- Set workflow (parallel / sequential/ individual) - Contact list - The user selects the recipients' permission - Reminder setter	- Signing sequencing - Document statuses viewing (viewed / signed /expired)
<b>Limitations</b>	Max file size 10MB (FTP subscription limitation)	Maximum file size 25 MB-	Limits are defined separately in each user service plan	-Maximum file size 20 MB
<b>Supported file types</b>	PDF, TXT, PNG, JPG, JPEG, ICO, BMP	.AS, .ASL, .ASP, .DOC, .HTM, .HTML, .PDF, .PDX, .RTF, .TXT, .WPD, .WPS.	.PDF, .PTX, .PPT, .DOC, .DOCX .XLS .XLSX .	TXT, PDF, DOC, .DOCX.
<b>Interface</b>				

mobile devices, and the documents are saved in a secure cloud in order to provide trust between the user and his/her business partner. ECDSA algorithm was used in encrypting the signature and verify it. SignOn meets the usability criteria and cross-platform requirements. It was designed to consider the security requirements. Likewise, the proposed application concentrates mainly on signer cloud communication and can exploit cloud benefits. Moreover, the proposed application is efficient and secure when compared with the existing applications. In the future, some features will be added to SignOn as a contact list, adding a comment when a signature is requested, the ability to sign with Initials, and developed as Web/iOS/Windows application.



## REFERENCES

- Alanazi, H., Zaidan, B., Zaidan, A., Jalab, H., Shabbir, M., & Al-Nabhani, Y. (2010). New Comparative Study Between DES, 3DES and AES within Nine Factors. *Journal of Computers*, 2(3), 152–157.
- Ali, A. (2015). Comparison and Evaluation of Digital Signature Schemes Employed in NDN Network. *International Journal of Embedded systems and Applications*, 5(2), 15-29.
- AlZain, M. A., Li, A. S., Soh, B., & Pardede, E. (2015). Multi-Cloud Data Management using Shamir's Secret Sharing and Quantum Byzantine Agreement Schemes. *International Journal of Cloud Applications and Computing*, 5(3), 35–52. doi:10.4018/IJCAC.2015070103
- Azizi, F. (2011). Advanced Electronic Signature [M.Sc. Thesis]. Department of Telematics, Norwegian University of Science and Technology, Trondheim, Norwegian.
- Bellare, M., & Rogaway, P. (2005). *Introduction to Modern Cryptography. Lecture Notes for CSE207*. Department of Computer Science and Engineering, University of California, San Diego. Retrieved from <http://cseweb.ucsd.edu/~mihir/cse207/>
- Berent, A. (2003). AES (Advanced Encryption Standard) Simplified. ABI software development. Retrieved from <http://index-of.es/Tutorials/AstalaVista/AESSimplified.pdf>
- Blythe, S. (2007). China's New Electronic Signature Law and Certification Authority Regulations: A Catalyst for Dramatic Future Growth of E-Commerce. *Chicago-Kent Journal of Intellectual Property*, 7(1), 1–32.
- Botes, J., & Penzhorn, W. (1994). An implementation of an elliptic curve cryptosystem. In *Proceedings of COMSIG '94 - South African Symposium on Communications and Signal Processing*. IEEE. doi:10.1109/COMSIG.1994.512441
- Brow, E. (2010). Elliptic Curve Cryptography, Math 189A: Algebraic Geometry. Retrieved from <https://www.math.hmc.edu/~ursula/teaching/math189/finalpapers/elaine.pdf>
- Brown, D. (2000). *The exact security of ECDSA. Technical report CORR 2000-54*. University of Waterloo.
- Bryson, J. (2015). *NIST: FIPS PUB 180-4: Secure Hash Standard (SHS)*. National Institute of Standards and Technology.
- Chaves, R., Kuzmano, G., Sousa, L., & Vassiliadis, S. (2006). Improving SHA-2 Hardware Implementations. In L. Goubin & M. Matsui (Eds.), *International Workshop on Cryptographic Hardware and Embedded Systems, LNCS (Vol. 4249, pp. 298–310)*. International Association for Cryptologic Research.
- Chiranth, B., & Shashikala, B. (2012). Survey of Performance Comparison of DES, 3DES and AES Algorithms. *International Journal of Data & Network Security*, 1(3), 47–50.
- Chiranth, B., & Shashikala, B. (2012). Survey of Performance Comparison of DES, 3DES and AES Algorithms. *International Journal of Data & Network Security*, 1(3), 47–50.
- Descriptions of SHA-256, SHA-384, and SHA-512. (2001). Retrieved from <http://www.iwar.org.uk/comsec/resources/cipher/sha256-384-512.pdf>
- European Electronic Signature Standardization Initiative Steering Group (EESSIG). (2001). Algorithms and Parameters for Secure Electronic Signatures (V.1.44 DRAFT). Retrieved from <http://docplayer.net/9961872-Algorithms-and-parameters-for-secure-electronic-signatures-v-1-44-draft-may-4-th-2001.html>
- Elmisery, A. M., Sertovic, M., & Gupta, B. B. (2017). Cognitive Privacy Middleware for Deep Learning Mashup in Environmental IoT. *IEEE Access: Practical Innovations, Open Solutions*. doi:10.1109/ACCESS.2017.2787422
- Embrogno, T. (2012). Electronic Signatures: How to Navigate the Last Mile for Your Firm. *Technology Tools For Today*, X(3), 1–8.
- ETSI. (2011). Electronic Signatures and Infrastructures (ESI); Algorithms and Parameters for Secure Electronic Signatures. Retrieved from <http://docplayer.net/13398437-Etsi-ts-102-176-1-v2-0-0-2007-11-technical-specification.html>

- FIPS 186-3. (2010). Suite B Implementer's Guide to FIPS 186-3 (ECDSA). Retrieved from <http://cits.eerx.ist.psu.edu/viewdoc/download?doi=10.1.1.182.4503&rep=rep1&type=pdf>
- Forouzan, B. (2007). *Data Communications and Networking* (4th ed.). McGraw-Hill Companies, Inc.
- Jain, G. (2012). Digital Signature Algorithm. *International Journal of Innovations in Computing*, 1(1), 1–6.
- Khalique, A., Singh, K., & Sood, S. (2010). Implementation of Elliptic Curve Digital Signature Algorithm. *International Journal of Computers and Applications*, 2(2), 21–27. doi:10.5120/631-876
- Khan, K. (2015). The Security of Elliptic Curve Cryptosystems - A Survey. *Global Journal of Computer Science and Technology: E Network. Web & Security*, 15(5), 7–9.
- Kinastowski, W. (2013). Digital Signature as a Cloud-based Service. In *Cloud Computing. The Fourth International Conference on Cloud Computing, GRIDs, and Virtualization* (pp. 68-72).
- Koppula, S., & Muthukuru, J. (2016). Secure Digital Signature Scheme Based on Elliptic Curves for Internet of Things. *Iranian Journal of Electrical and Computer Engineering*, 6(3), 1002–1010.
- Kumar, V., and Koul, P. (2011). Robust RSA for Digital Signature. *International Journal of Computer Science Issues*, 8(6-3), 359-362.
- Laborde, C. (2010). *Electronic Signatures in International Contracts*. Peter Lang GmbH. doi:10.3726/978-3-653-00124-2
- Locke, G. (2009). *NIST: FIPS PUB 186-3: Digital Signature Standard (DSS)*. National Institute of Standards and Technology.
- Martoni, M., & Palmirani, M. (2013). Remote Signatures for e-Government: The Case of Municipal Certification in Italy. In *13th European Conference on e-Government (ECEG)* (pp. 310-318).
- Mella, P., & Grance, T. (2011). *The NIST Definition of Cloud Computing. Special Publication 800-145, NIST Technical Series Publication*. Gaithersburg: Information Technology Laboratory, National Institute of Standards and Technology.
- Mendel, F., Pramstaller, N., Rechberger, C., & Rijmen, V. (2006). On the collision resistance of RIPEMD-160. In S. K. Katsikas, J. Lopez, M. Backes et al. (Eds.), *International Conference on Information Security, LNCS* (Vol. 4176, pp. 101–116).
- Menezes, A., Qu, M., Stinson, D., & Wang, Y. (2001). Evaluation of Security Level of Cryptography: ECDSA Signature Scheme. Certicom Research. Retrieved from [https://www.ipa.go.jp/security/enc/CRYPTREC/fy15/doc/1051\\_ecdsa.pdf](https://www.ipa.go.jp/security/enc/CRYPTREC/fy15/doc/1051_ecdsa.pdf)
- NIST. (2001). FIPS PUB 197: Advanced Encryption Standard (AES). Federal Information Processing Standards Publication, Technical Series Publication.
- Pointcheval, D., & Stern, J. (1996). Security Proofs for Signature Schemes. In U. Maurer (Eds.), *Advances in Cryptology - EUROCRYPT '96, LNCS* (Vol. 1070, pp. 387-398).
- United Nations Commission on International Trade Law. (2009). *Promoting confidence in electronic commerce: legal issues on international use of electronic authentication and signature methods*. United Nations Publication.
- Rossnagel, H., & Royer, D. (2005). Investing in Security Solutions - Can Qualified Electronic Signatures be Profitable for Mobile Operators. In *11th Americas Conference on Information Systems (AMCIS)*, Omaha, NE (pp. 3248-3257).
- Razaq, M. A., Gill, S. H., Qureshi, M. A., & Ullah, S. (2017). Security issues in the Internet of Things (IoT): A comprehensive study. *International Journal of Advanced Computer Science and Applications*, 8(6), 383–388.
- Shakil, K., Zareen, F., Alam, M., & Jabin, S. (2017). BAMHealthCloud: A biometric authentication and data management system for healthcare data in cloud. *Journal of King Saud University – Computer and Information Sciences*.
- Singh, S., & Singh, N. (2015). Internet of Things (IoT): Security Challenges. Business Opportunities & Reference Architecture for E-commerce. In *International Conference on Green Computing and Internet of Things (ICGCIoT)* (pp. 1577-1581). IEEE. doi:10.1109/ICGCIoT.2015.7380718

Siwik, L., and Mozgowski, L. (2015). Server-Side Encrypting and Digital Signature Platform with Biometric Authorization. I. *J. Computer Network and Information Security*, 7(4-1), 1-13.

Srinivasan, S. (2014). Security, Trust, and Regulatory Aspects of Cloud Computing in Business Environments. In *Information Science Reference* (Ch 5, pp. 92-114).

Srivastava, A. (2013). *Electronic Signatures for B2B Contracts: Evidence from Australia*. Springer India.

Stadick, R. (2006). A Java implementation of the elliptic curve digital signature algorithm using NIST curves over GF(p) [Honors Bachelor of Science Thesis]. Oregon State University, University Honors College.

Stallings, W. (2011). *Cryptography and Network Security Principles and Practice* (5th ed.). Prentice Hall.

Stergiou, C., Psannis, K. E., Kim, B. G., & Gupta, B. (2018). Secure integration of IoT and cloud computing. *Future Generation Computer Systems*, 78(Part 3), 964–975. doi:10.1016/j.future.2016.11.031

Stern, J. (2001). The Electronic Signatures in Global and National Commerce Act. *Berkeley Technology Law Journal*, 16(1), 391–414.

Subramaniam, A., Chaudhry, J., & Ahmad, M. (2012). A Study on Elliptic Curve Digital Signature Algorithm (ECDSA) for Reliable E-Commerce Applications. *Smart Computing Review*, 2(1), 71–78.

Sumroy, R. & Sherrard, B. (2012). *Electronic Signature: are we getting there?* Slaughter and May.

Tewari, A., & Gupta, B. B. (2017). A lightweight mutual authentication protocol based on elliptic curve cryptography for IoT devices. *International Journal of Advanced Intelligence Paradigms*, 9(2-3), 111–121. doi:10.1504/IJAIP.2017.082962

Wang, F. (2014). *Law of electronic commercial transactions: Contemporary issues in the EU. US and China*: Routledge.

Zongqu, Z. (2010). A Digital Signature System Based on Discrete Logarithm in Mobile Payment. In *3rd International Symposium on Electronic Commerce and Security Workshops (ISECS '10)*, Guangzhou, P. R. China (pp. 84-86).

Sahar A. El-Rahman has received her M.Sc. (2003) in an AI Technique Applied to Machine Aided Translation, and PhD (2008) in Reconstruction of High-Resolution Image from a Set of Low-Resolution Images, from the Faculty of Engineering- Shoubra, Benha University, Cairo, Egypt. She is currently Assistant Professor, College of Computer and Information System, Princess Nourah Bint Abdulrahman University (Saudi Arabia). Also, she is Assistant Professor from 2008 till now at Faculty of Engineering-Shoubra, Benha University, Cairo, Egypt. She has published many papers in national and international journals and conferences. Her research interests include Computer Vision, Image Processing, Signal Processing, Information Security, Human Computer Interaction, E-Health, Big Data and Cloud Computing. She is a member of IACSIT (International Association of Computer Science and Information Technology) since 2013. A member of Internet Society (ISOC) since 2015, a member of IAENG (International Association of Engineers) since 2011, and a member of the Egyptian Engineers' Syndicate since 1997.

Daniyah Abdullah Aldawsari is a software developer at Yesser (E-Government Transformation Program). Aldawsari is a computer science graduate from PNU. She previously trained at PwC as a technology consultant.

Mona Aldosari's current interests are in web development and database management but I am also growing my knowledge and skills in other areas.

Omaimah Alrashed is a Web developer at E-commerce Sea in Saudi Arabia. She graduated from Princess Nourah University with a degree in computer science.

Ghadeer Alsubaie graduated from Princess Norah University, computer science department. Alsubaie is now working as a system engineer in Tata consultancy services.