**Abstract**

The proposed work introduces a unique data security protocol for validating digital e-certificates through achievement of the four milestones of data security- authentication, confidentiality, integrity and non-repudiation. To resolve authority issues and serve the aim of server authentication, copyright issuing authority signature is kept both at the server database and the client end(e-certificate). Client authentication is ensured by collecting client’s biometric information i.e. fingerprint and storing it at the server and client side. Using the client’s biometric data, the server generates a unique pair of public and private share of the fingerprint. The client’s public share is stored at the server database. For automation of manual financial e-certificates, the certificate is divided into four regions. Each region is partitioned into four segments. The server conceals the ownership signature and the client’s private share in the first and second segments of each region of e-certificate. In the third and fourth regions of each segment, the client’s signature and the nominee’s fingerprint are embedded secretly. All these signature fabrication processes are done through secure hash computations.In this aspect, the signature fabrication in each segment is based on two hash values determining the block interval of signature data embedding and the authentic circular orientation of signature fragments. Significantly, these hash values are derived through operations on e-certificate number, its date of maturity, money transaction amount and name of the client. Finally, all these parameters are validated at the time of verification by the server through utilizing those identical hash operations on the e-certificate data. The scheme also strengthens robustness and imperceptibility by variably encoding secret data on different transformed pixel bytes and maintaining multi-copy signature embedding. Overall the proposed approach confirms superiority over the existing approaches with exhaustive simulation results.

Index Terms: Dynamic Authentication, e-document Validation, Multi-Signature Fabrication, Variable Encoding.

**Table of Contents**

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  | Title Page | i |
|  | Declaration of the Students | iii |
|  | Bonafide Certificate | iv |
|  | Abstract | v |
|  | Acknowledgement | vi |
|  | List of Figures | vii |
|  | List of Tables (optional) | viii |
|  |  |  |
|  |  |  |
| **1.** | **INTRODUCTION** | **1** |
|  | * 1. Problem Definition   2. Project Overview/Specifications   3. Hardware Specification   4. Software Specification   1.3.1  1.3.2 | 1  2  3  4  4 |
|  |  |  |
| **2.** | **LITERATURE SURVEY** | **5** |
|  | 2.1 Existing System  2.2 Proposed System  2.3 Feasibility Study | 5  6  7 |
|  |  |  |
| **3.** | **SYSTEM ANALYSIS & DESIGN** |  |
|  | 3.1 Requirement Specification  3.2 Flowcharts / DFDs / ERDs  3.3 Design and Test Steps / Criteria  3.3 Algorithms and Pseudo Code  3.3.1  3.3.2 | 9  10  12  16  18  19 |
|  |  |  |
| **5.** | **CONCLUSIONS AND SCOPE FOR FUTURE WORK** | **47** |
| **6.** | **REFERENCES** | **49** |

|  |  |
| --- | --- |
| **1.** | **INTRODUCTION** |
|  | **1.1** **Problem Definition**  **1.1.1** The idea is to address all the essential data security features like authentication, confidentiality, integrity and non-repudiation to certify both the ownership claims and critical data validations.  **1.1.2** Additionally, this approach also incorporates secure data hiding techniques for authentications and in this aspect hash value based variable data encoding concepts are also focused for strong validations.  **1.1.3** This is mainly focused to strengthen the existing authenticity mechanisms.  **1.2** **Project Overview/Specifications**  The proposed work introduces a unique data security protocol for validating digital e-certificates through achievement of the four milestones of data security- authentication, confidentiality, integrity and non-repudiation. To resolve authority issues and serve the aim of server authentication, copyright issuing authority signature is kept both at the server database and the client end(e-certificate). Client authentication is ensured by collecting client’s biometric information i.e. fingerprint and storing it at the server and client side. Using the client’s biometric data, the server generates a unique pair of public and private share of the fingerprint. The client’s public share is stored at the server database. For automation of manual financial e-certificates, the certificate is divided into four regions. Each region is partitioned into four segments. The server conceals the ownership signature and the client’s private share in the first and second segments of each region of e-certificate. In the third and fourth regions of each segment, the client’s signature and the nominee’s fingerprint are embedded secretly. All these signature fabrication processes are done through secure hash computations.In this aspect, the signature fabrication in each segment is based on two hash values determining the block interval of signature data embedding and the authentic circular orientation of signature fragments. Significantly, these hash values are derived through operations on e-certificate number, its date of maturity, money transaction amount and name of the client. Finally, all these parameters are validated at the time of verification by the server through utilizing those identical hash operations on the e-certificate data. The scheme also strengthens robustness and imperceptibility by variably encoding secret data on different transformed pixel bytes and maintaining multi-copy signature embedding. Overall the proposed approach confirms superiority over the existing approaches with exhaustive simulation results.  **1.3** **Hardware Specification**  Hardware performs a leading role to implement any project work. To implement this project, we have used-  1.3.1 A high speed computer processor (Intel Core i processor (5th generation))  1.3.2 A 4 GB RAM  1.3.3 Graphics and Multimedia supported preferable  **1.4** **Software Specification**  A Though this project is going on Linux environment and we are using C as programming language but it can also be implemented with the following components  1. 1 Linux or Windows operating system  1. 2 Compiler of any programming language  1. 3 JIMP  1. 4 Irfan View and  1. 5 MATLAB computer programming language as testing tool. |

**2. LITERATURE SURVEY**

**2.1** **Existing Related Works**

The traditional concept for digital document or image authentication involves secret fabrication of some copyright signature or watermarking within the concern image document, such that this embedded signature can only be revealed by the intended authorized party [4-14]. Also, such signatures should be fabricated in a robust manner and will not be easily removed by external signal or image processing attacks [4-10]. Vitally, these issues are also considered for e-document authentications with some additional data security and authenticity scenarios.

**2.1.1** **Existing Approaches on e-document Authentication**

On these works Anitha et al. [1] adopted biometric watermarking where iris image of the user is converted to a live template of bit coded signature and is periodically duplicated within the cover image. The bit encoding is done through Least Significant Bit (LSB) method and the model is capable of detecting large tampers. Then Mishra et al. [2] also embedded biometric details like fingerprints, retinal scan, facial characteristics, palatal patterns and DNA on the smart card and their different data combination categories will identify the owner. During validation, the comparison of the recaptured biometric details and the detected biometric combination data from the card will authenticate the owner. After that, Hasan et al. [3] has implemented segmented multi-watermarking onto the three regions of the different intensity components such as R, G and B through discrete wavelet transform (DWT). The sub bands of LH1, LL2, LL3 are used for watermark coding while merging of data coded intensities will form the watermarked image.

In view of the above, it can be said that if multiple identity signature of the owner is embedded in the cover image then the multi confirmation and recognition phases associated with it makes it difficult to access the digital document by unauthentic parties. Also, the dispersing of signature in different regions improves the robustness against several forms of attacks. Among several existing multi watermarking approaches Segmented watermarking provides better encryption and data security. Various existing multi-watermarking techniques are discussed below-

**2.1.2** **Concepts on Multi Signature Driven Authentication**

Interestingly, a significant amount of such works has now inclined towards multi-watermarking due to their trustable authentications and here the segmented or partitioned multi-watermarking concept serves good robustness. Among these works Nasir et al. [4] tried embedding of encrypted fragments of the binary watermark on different regions of the blue components through spatial encoding technique. Then Behnia et al. [5] has upgraded such ideas with chaotic map technique where different binary watermarks were embedded onto the concern Red, Green, and Blue channel. Here, both the embedding positions and encoded data bits were securely encrypted. Additionally, Sadh et al. [14] has shown coding of watermarks on green and blue components where four parts of the encrypted watermark were hidden onto the different regions of these color intensity planes through spatial coding.

Further, improvement of these works was then continued with frequency or transform domain-based data embedding techniques for better security and robustness. On such works, Bhatnagar et al. [6] embedded multi-watermarks on non-overlapped DCT blocks and data encoding is done by utilizing the threshold of the block energy. Significantly, current transform domain works are mostly focusing Discrete Wavelet Transformed (DWT) coefficients to encode the secret data. Among these approaches Babaei et al. [7] adopted multi-watermarking on non-overlapping uniform wavelet blocks. Then Thanki et al. [13] has shown two-level watermarking where secure transformed watermark is added on HH3 and HH4 coefficients of DWT. Further, Singh et al. [14] proposed reliable spread-spectrum method where false noise orders of watermarking bit were coded on the chosen DWT sub band coefficients. Next Mohananthini et al. [9] has used composite multiple watermarking on DWT sub band coefficients for both medical and color images. After that Mohananthini et al. [10] has shown better performances for segmented watermarking with watermarks are embedded on LL2 sub-band of 2 level DWT coefficients. Also lately same authors [8] have further emphasized better effectiveness for the segmented type with encoding of two watermarks on the odd-rows & columns and even-rows & columns of wavelet coefficients. In addition, to these Sadh et al. [11] has also stressed the segmented concept with separate hiding of encrypted watermarks onto the different regions of green and blue components by adopting spatial encoding techniques.

Apart from these above-mentioned literature study some works have also utilized the digital signature concept through visual cryptographic approach for stronger authenticity. In this context, Radharani et al. [12] has embedded the concern signature shares imperceptibly onto the particular wavelet transformed coefficients. Here the original signature is recovered by merging two of its applicable shares such as that hidden share detected from the cover image and its respective pair available with the receiver.

In view of these present literature study, it is quite clear that the existing works have very rarely focused on designing the trusted data security protocol satisfying the critical challenges. Although one such work was very recently reported by Chowdhury et al. [15], where dynamical secret fabrication of multiple signatures are focused and it is decided on the basis of dynamical casting of a visible logo image. However, this work even does not cover all the critical data security issues strongly, and does not employ the most trustable authentications for both the client and server side. So in a nutshell the existing approaches are mostly lacking trustable data security protocols or solutions covering all the crucial data security issues and hence this proposed scheme address such issues with critical enhancements as discussed next.

**2.2** **Enhancement over existing approach**

For validating the data and resolving the ownership issues the signatures of both the issuer and the concerned candidate are secretly embedded in the produced e-document. The hash value, derived from the certificate data is used to cast multiple copies of four different signatures in the e-certificate. The proposed concept can be properly visualized in Fig-1 where each of the four equally partitioned regions of the cover image are further divided into four equal segments P1, P2, P3, P4 which are non-overlapped. The concerned signatures are fabricated based on circular orientation of the fragments of the signature and the matrix interval of the signature data casting. For convenience, the secret signature bits are embedded on pixel bytes of a 2x2 sub image block matrix. Here the concerned data matrix interval is found from the specific certificate data and the starting signature fragment index is determined by deriving the hash values. The pictorial representation in Fig.1 clearly demonstrates multi-copy hosting of each signature where the same signature casting pattern is followed for all the regions. This secret signature hiding concept is elaborated for region-2 of the e-certificate by focusing on a sample 2x2 sub block matrix, shown with the elements [A11, B11, C11, D11]. Here Partition-1 hosts the University’s ***Company’s copyright signatur***e ***(ASign)*** with its circular fragment orientation started with fragment index = 4, which is derived from the hash operation on ***policy number (PNO***). Hence the first matrix element A11 hosts the bit for the 4th fragment of the ***ASign***. Similarly, the 2nd element (B11), 3rd element (C11) and 4th element (D11) of the sub block matrix host the signature bit for fragment 1, 2 and 3 of ***ASign*** respectively. . In addition of this signature fragment orientation another hash value is derived from same ***PNO*** which will determine this secret data fabrication matrix interval in segment-1. Further, by adopting this similar idea other concern signatures are also fabricated in the respective segments of region-2. In this aspect the ***Private share key of the client’s fingerprint (PrvtSk)***, ***Client’s fingerprint (CFing) and Nominee’s fingerprint (NFing)*** fabricated on segment-2, 3, 4 respectively. Essentially, all these signatures are fabrications, circular orientation of the signature fragments and also the data matrix interval for casting secret data are decided with the help of the generated hash values. So, to promote strong validation cum authentication, this ***ASign***, ***CFing*** and ***NFing*** are further secretly fabricated based on their concern hash values derived from ***DOM, Amount, PName*** as needed***.***

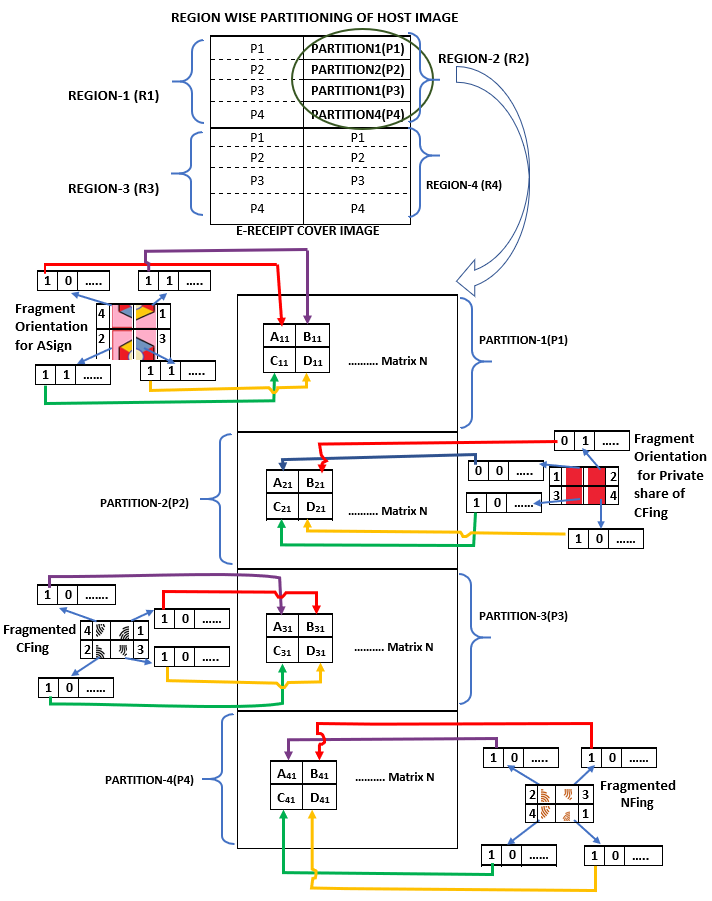


Fig1: Signature Fabrication on Auto Generated e-document

**2.3 Proposed e-Document Authentication Protocol**

The e-certificate authenticity confirmation starts with concealment of issuing authority signature and client fingerprint's private share ***in 1st and 2nd partition of each region of the e-certificate by the server.*** Further, the client’s biometric fingerprint and the concerned nominee’s fingerprint are also embedded ***in 3rd and 4th partition of each region of the e-certificate by the server***. Importantly, each of these signature fabrications are done based on two hash values derived from the four important data components of the e-certificate. In this aspect, the first hash value decides the starting signature fragment index for circular orientations of the embedded signature fragments while the second hash value determines the block interval for this signature fragment data casting. Critically, these hash values are derived from the ***Policy No. (PNO), Date of Maturity (DOM), Amount and Policy Holder’s Name (PName) for fabricating the signatures*** in the first, second, third and fourth partition of each region respectively. Vitally, all these sensitive data and fingerprints are maintained in the server database for future verification. This authenticated e-certificate is now issued to the concerned client for future validation of this e-certificate regarding reimbursement of the invested amount. This validation is confirmed by the insurance authority by employing those same hash functions executed on the same critical data of the e-certificate and by sensing those secretly fabricated signatures from the authenticated e-certificate as produced by the client. to ensure the ownership or the authentication aspects, the public share of the client fingerprint is collected at the verification time and merged with the detected private share found from the e-certificate to technically confirm the client's digital signature scenario. Additionally, the on-spot taken biometric fingerprint of the client or the nominee present to take the claim is also matched with the applicable fingerprint sensed from the respective partition of the e-certificate confirms the non-repudiation aspect of this scheme. The step wise explanation of the protocol is as follows:

**Step1:** Policy Company secretly fabricates Authority’s copyright signature ***(ASign)*** on the first partition of each region of the digital bond of the policy. In this context the server of Policy Company derives two different hash values from the ***policy number (PNO)***. The concept of signature embedding is pictorially visualized in the ***Fig A***. Two hash values (F11, F12) derived from the ***PNO*** determine the starting fragment index **[F11 ϵ {1,2,3,4}]** for the circular orientation of signature fragments and the particular matrix interval **[F12 ϵ {1,2,3}]**. Importantly these hash operations are reflected by the following equations (1) and (2) where ri is the ith ϵ {1,2,…n} digit in the premium number.

**F11 = [(PNO + ) Mod 4] + 1**  (1)

**F12 = [(PNO - ) Mod 3] + 1** (2)

**Step2:** The Client’s Thumb Impression(CFing) is segregated into two parts, first one is public share(PubSk) and second one is client’s private share(PrvtSk).The Authoritysecretly embeds ***client’s private share (PrvtSk)*** on the second ***partition*** of each region of the same ***e-policy bond.*** The ***PubSk*** is given to the Client for further use. Importantly, this signature embedding is done on the ***Date of Maturity (DOM)*** and this concept is demonstrated in the ***Fig. A***. Here, two hash values determining the starting fragment index **[F21 ϵ {1,2,3,4}]** for the circular orientation of signature fragments and the particular matrix interval **[F22 ϵ {1,2,3}]** of the signature data embedding. Importantly these two-hash operations are highlighted through the following equation of (3) & (4), where DOM is represented as DD/MM/YYYY.

**F21 = [(MM) Mod 4] + 1** (3)

**F22 = [(DD+MM+YYYY) Mod 3] + 1** (4)

**Step3: *The stored thumb impression of the client (CFing)*** is fabricated on the 3rd partition of each region based on hash values derived using ***Amount*** as shown in Fig. A. Here, two hash values determining the starting fragment index **[F31 ϵ {1,2,3,4}]** for the circular orientation of signature fragments and the particular matrix interval **[F32 ϵ {1,2,3}]** of the signature data embedding. Importantly these two-hash operations are highlighted through the following equation of (5) and (6), where di is the ith digit [i ϵ {1,2,…n}] of Amount.

**F31 = [ (Amount + ) Mod 4 ]**  (5)

**F32 = [ (reverse of Amount + ) Mod 3 ] +1** (6)

**Step4:** Now, ***the stored thumb impression of the nominee (NFing)*** is fabricated on the 4th partition of each region based on hash values derived on the basic of ***Policy holder’s name (PName)*** as shown in Fig. A. Here, two hash values determining the starting fragment index **[F41 ϵ {1,2,3,4}]** for the circular orientation of signature fragments and the particular matrix interval **[F42 ϵ {1,2,3}]** of the signature data embedding. Importantly these two-hash operations are highlighted through the following equation of (7) and (8), where di is the ACSII value of ith character [i ϵ {1,3,5,…n}] of ***PName*** and pi is the ACSII value of ith character [i ϵ {0,2,4,…n}] of ***PName***.

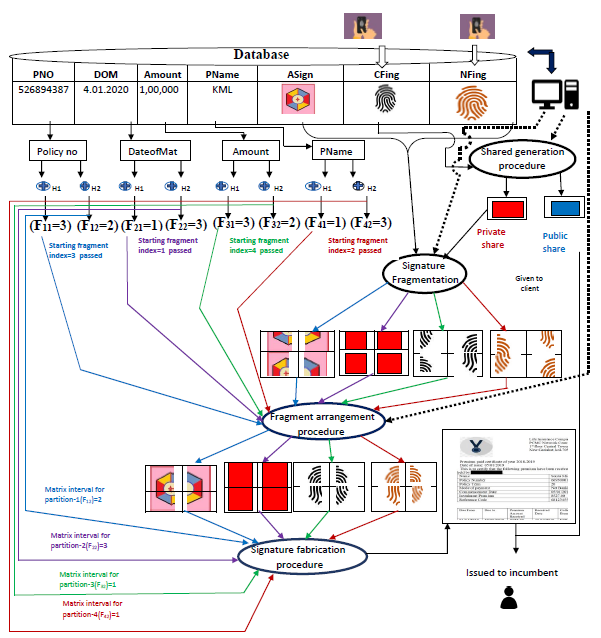
**F41 = [(( )+ ( ))Mod 4 ]**  (7)

**F42 = = [(( )- ( ))Mod 3 ] +1** (8)

**Step5:** This signature embedded digital ***e-bond*** of the policy is issued to the ***policy holder.*** Along with this e-bond a ***Public Share key (PubSk)*** is also issued to the ***policy holder***. This e-bond can be download.

**Step6**: The authorized client login to the server-side database by using the ***policy number (PNO)***as the login id and the concern applicable password.

**Step7**: Server validates client’s authorization and conveys the respective acknowledgment to the client.



**Step8:** To claim the Amount of the policy, the Client or Nominee (as per the rules and regulation of the Policy issuing Authority) of the respective policy have to follow the following Steps after the Maturity Date (DOM) of the policy.

**Step9:** To claim the Amount of the policy the client or the nominee has to visit the issuing authority physically for proper verification and validation.

**Step10:** The Client or the Nominee has to produce the digital e-bond of the policy and the Public Share key which were issued to the client at the time of policy opening to the Authority.

**Step11:** At this stagethe Authority checks the Policy Number valid or not. If it is valid then only further steps will be applicable.

**Step 12**: Server computes those same hash values **F11**& **F12** byadopting Eq. (1) & (2) and utilizing the uploaded ***policy number*** (***PNO***). Similarly, other exact same hash value sets **(F21, F22), (F31, F32)** are computed by the server through Eq. (3, 4), (5, 6) with utilization of the uploaded ***Date of Maturity(DOM), Amount.*** Hash value **(F41, F42) are** computed based on the ***Policy holder’s name(PName)*** as fetched from server utilizing the Eq, (7, 8).

**Step 13:** At this stage the server utilizes those concern hash values applicable for the concern partition and detects the respective signatures from the particular partition of the ***e-bond*** of the policy. So, by tracking the data matrix intervals like **F12**, **F22**, **F32** and **F42** through those derived hash values the server extracts the concern signature fragments of ***ASign, PrvtSk , CFing*** and ***NFing*** respectively. Critically, such fragments of ***ASign, PrvtSk, CFing*** and ***NFing*** are also detected from the concern ***partition*** 1, 2, 3 and 4 of each region respectively.

**Step 14:** Now, each of the four extracted signature fragments found from the concern partition of the particular region is merged based on the respective starting fragment index derived as hash value. So, by following this method the concern signatures ***ASign, PrvtSk, CFing*** and ***NFing*** are reconstructed by utilizing the starting fragment index hash values **F**11, **F**21, **F**31 and **F**41 respectively. Hence, with this idea, four copies of each of the signature are recovered from different regions of the authenticated digital ***e-bond***.

**Step15:** At this moment, the reconstructed ***Private Share Key (PrvtSK)*** is merged with the ***Public Share Key (PrvtSK)*** to obtain the Client Thumb Impression (CFing).

**Step16:** Further, the three reconstructed signature copies (***ASign, CFing*** and ***NFing***) and produced Client Thumb Impression ***(CFing)*** are matched.

In order to match the signatures in presence of the Client, for partition 3 ***CFing*** is matched with on spot taken Thumb Impression of the Client and for partition 1 and 2 ***ASign*** and ***CFing*** are matched to the original ***ASign*** and ***CFing*** copy stored at the server database. For, this processor Nominee’s Thumb Impression is not needed to verify. But, if Client is not present, then ***CFing*** and ***Asign*** will be matched to original signature copy which are stored in server database. And, the ***NFing*** will be verified with on spot taken Thumb Impression of the Nominee. Now, the best-detected copy for each signature is tracked. Now if at least one of the signature copies for each signature category satisfies a predefined threshold value of matching then server conforms the total validation of the e-Certificate. The step-wise mathematical discussion of this final data validation and authentication procedure is highlighted as follows.

Let us assume, N[reg][sig][part] stores the region-wise four decomposed signatures for each signature orientation type. Hence the array index reg ϵ {1, 2, 3, 4} denoting the concern region number while sig ϵ {1, 2, 3, 4} is used for storing each signature copy related to a specific orientation and part ϵ {1, 2, 3, 4} is for the 4 decomposed part of each signature. Further Sort\_Merge operation is done to get the MARGED [reg][sig] matrix which stores the reconstructed signature images of each region. The function ***PrivatePublicShareMerged()*** is used merge the ***Client’s public share(PubSk)*** with ***Client’s private share(PrvtSk)*** to deliver the output as ***Client’s Thumb impression (CFing)***.Then each signature stored in different regions are compared with the signature image stored in STORE [sig] array. The STORE [sig] array is constructed by fetching ASign from the server’s database and on the spot taken ***CFing, NFing***(as per step:16). If signatures of MARGED [reg][sig] and signatures of STORE [sig] matches with some predefined threshold value then a success message is sent to the client, else a failure message is generated.

STORE [sig]sig = 1 to 4 =[ ***ASign, CFing, CFing*** and ***NFing***];

N[reg][sig][part]reg= 1 to 4, sig = 1 to 4, part= 1 to 4 = FUNC-1 (I2, **F**12, **F**22, **F**32, **F**42);

MARGED [reg][sig]reg = 1 to 4, sig =1 to 4 = Sort\_Merge (N[reg][sig][part]reg=1 to 4, sig= 1 to 4, part =1 to 4, **F**11, **F**21, **F**31, **F**41);

for sig=1 to 4

MERGED [reg][2] = PrivatePublicShareMerged(MERGED[reg][2],PrivateShareKey)

**END**

for reg = 1 to 4

for sig = 1 to 4

**If** (MERGED[reg][sig] = = STORE[sig])

Count[sig] ← Count[sig] + 1;

**Else**

Continue;

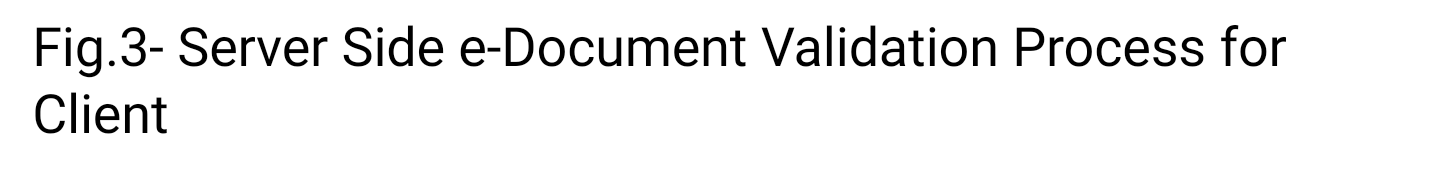
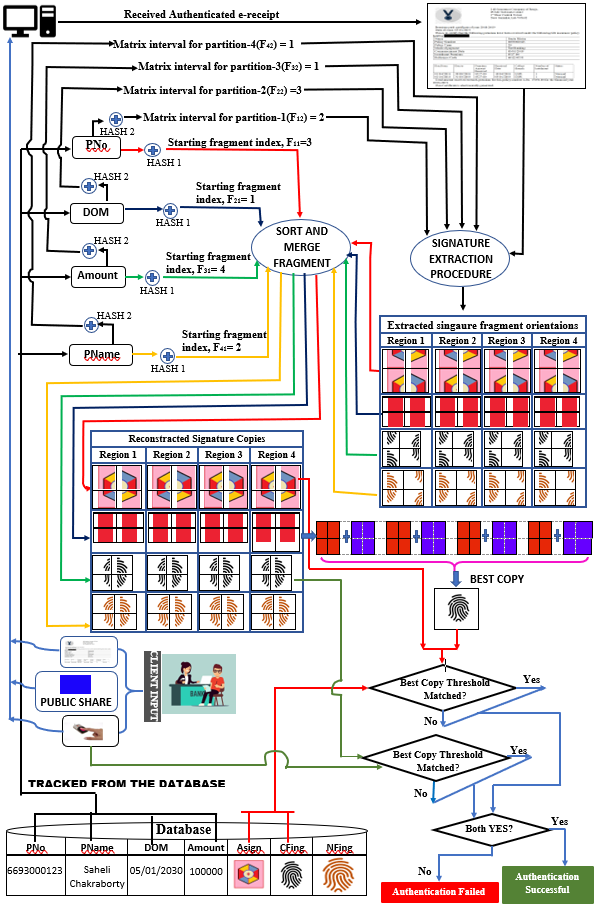
**If** (Count[1] && Count[2] &&Count[3] && Count[4]==1)

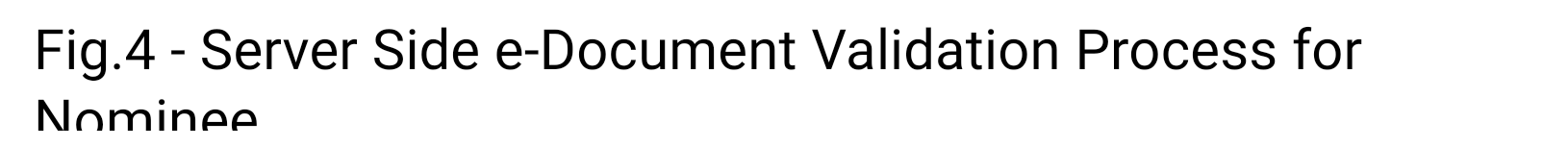
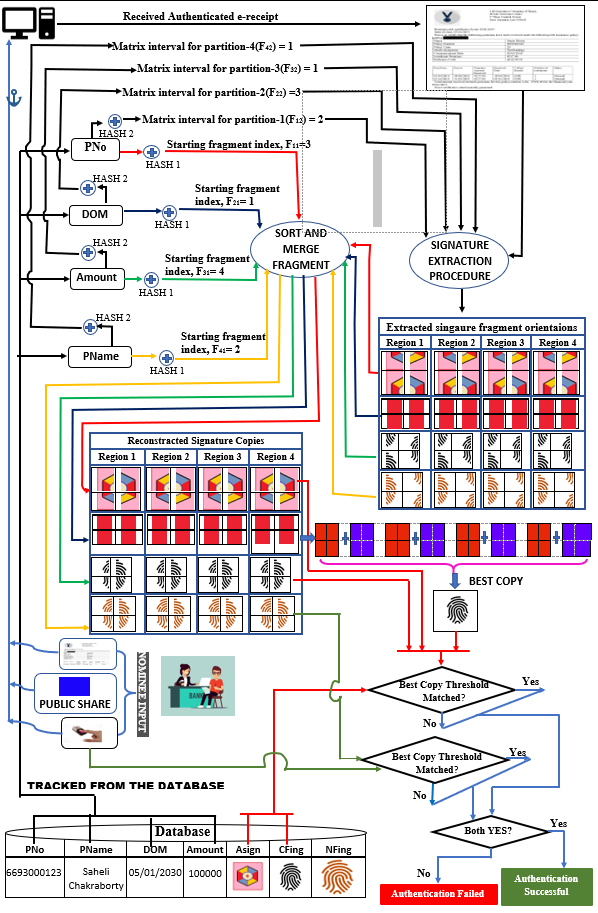
Send(“Successful Authentication”);

**Else**

Send(“Authentication Fail”);

**End;**





|  |  |
| --- | --- |
| **3.** | **SYSTEM ANALYSIS & DESIGN** |
|  | **3.1 Signature Bit Fabrication and Detection**  The cover host image is logically divided into four similar regions, which are further divided into four equal partitions (as shown in Fig.1). Now, each partition is segregated into 2\*2 consecutive nonoverlapping sub-block matrices of pixel bytes. Now, one single bit is fabricated on each pixel byte element in such 2\*2 sub-block matrix. To achieve this, the pixel bytes of sub-block matrix is transformed encoding single bit of the signature. Further, these bit coded pixel byte elements are reverse transformed to obtain the final signature bit fabricated matrix. To implement such signature encoding technique, 1st,2nd,3rd matrix elements are forward transformed using the operations shown through the matrix in (10). Interestingly, 4th element of sub-block matrix kept intact for spatial encoding of a single signature bit on it to optimize the robustness and data hiding imperceptibility issue. Finally,1st,2nd,3rd elements are passes through reverse transformation procedure to achieve the ultimate signature bit fabricated matrix which is sent. For, bit extraction 1st,2nd,3rd matrix elements are forward transformed and their hidden bits are extracted from co-responding components. Here the 4th element is directly extracted. Now, these distinguished bits from co-responding matrix elements are sorted in proper sequence to get the respective fragments. **3.3.1 Signature Fragment Bit Insertion Algorithm** |

**Block Transformation Procedure**

Let**,** Sub matrix block of a region is **Mbn=[ a, b, c , d ]**, where {a, b, c, d} ϵ{0,1,.,255} is the pixel byte values and bn ϵ{1,..,N} is the concern matrix number. The transformations performed for all the four regions of the whole cover image is given as -

**Forward Transform**

|  |  |
| --- | --- |
| **A1=(a-c)/2** | **A2=(b-c)/2** |
| **A3=(a+b)/2** | **A4=d** |

**Mbn / =** (10)

|  |  |
| --- | --- |
| **a=A1+A3-A2** | **b= A2+A3-A1** |
| **c=A3-A1-A2** | **d=A4** |

**Mbn// =**

(11)

Here Mbn=[Aj] is the transformed matrix, where Ai denotes transform element at co-responding index j ϵ{1,2,3,4}.Now, using matrix operation (10) one signature bit is encoded on each transformed element of Mbn/,to obtain Mbn//.Then reverse transformation is used using matrix operation (11) on Mbn// to produce the final bit casted matrix Mbn///.

To perform signature bit encoding on respective matrix element, two-bit coding functions (FUNCTION1(Aj, Фi) and FUNCTION2(Aj, b)) are applied where both functions operate on Aj as follows

Pj = **FUNCTION1(Aj,** Фj**):**

**Start**

**If**(Фj==0) **Then**

**[** **If** ((Aj Mod 3) == 0) **Then** Pj = Aj;

**Else**  Pj = (Aj – (Aj Mod 3));

**]**

**If**(Фj==1) **Then**

**[ If** ((Aj Mod 3) ==0) **Then** Pj = (Aj+1);

**Else**  Pj = Aj;

**]**

Return Pj;

**End**

**Rfj** = **FUNCTION2(Aj, b):**

**Start**

**If** ((Aj Mod b) == 0) **Then** (L = Aj);

**Else** L = (Aj – (Aj Mod b));

U = L+ b;

**If**(U>256) **Then** Rfj = 254;

**Else** Rfj = (L+U)/2;

Return Rfj;

**End**

Here, FUNCTION1 returns the co-responding coded bit value as Pj while taking co-responding transform value Aj and Фj, the particular bit to encode. as argument. On the other hand, FUNCTION2 takes the transformed value Aj and the particular multiple b ϵ {4,6,8} which to be considered while determining the threshold reference point Rfj with respect to Aj. FUNCTION2 calculates the upper multiple (U) and lower multiple (L) number of b with reconsidering Aj and finds their mid-value as the threshold reference point (Rfi). FUNCTION2 returns Rfj, which is further used for bit encoding on Aj. To understand this bit hiding concept the whole algorithm of bit hiding on concern matrix elements are step wise discussed as follows –

**Input:** A color host image and four-color signature images.

**Output:** Authenticated cover image hosts four signatures.

**Method:** At the beginning, the co-responding partition of the host image is segmented into consecutive sets of 2x2 sub-block matrices of pixel bytes as Mbn. Then each of the elements is transformed through matrix operation as shown in (10) which gives transformed matrix, Mbn/. After that, the fraction part of the element of Mbn/ is separated. In each of the transformed integer part of the sub-block matrix element Aj single signature bit is encoded to produce Cj in Mbn//. Now, for each of the element previously separated fraction part of Mbn/ and integer part ( Cj ) of Mbn// is clubbed together with co-responding elements. Now, lastly updated Mbn// is processed through reverse transformation using matrix operation (11) which will produce the final signature bit fabricated matrix Mbn///. As the receiver also operates on this Mbn///, so the receiver first performs forward transform using matrix operation (10) on matrix Mbn/// to track those bit codded matrix elements and then fraction part of elements is separated. Further, from the integer part of the element, those respective hidden bits are extracted from the concern transform matrix element. To discuss this bit hiding algorithm let, the found matrix interval for the concern image segment is tr and the steps of coding are-

**For matrix number t =1 to L**

**Do**

**Step1:** Perform forward transform on Mbn as per matrix operation (10).

**Step2:** Separate integer part and fraction part of each sub-matrix block elements. And apply **Step:3** and **Step:4** only on integer part of the elements.

**Step3:** Read chunk of four fragment bits Фjϵ{0,1} for j=1,2..4.

**Step4:** Code each Фjϵ{0,1} on the concern transformed value Aj of Mbn’to get the respective coded value Cj as

**If**(INDEX==0) **Then**

**[**

Rf1 = FUNCTION2(A1,4);

**If** (Ф1 ==1) **Then** C1 = Rf1+1;

**Else** C1 = Rf1-1;

Rf2 = FUNCTION2(A2,6);

**If** (Ф2 == 1) **Then** C2 = Rf2+1;

**Else** C2 = Rf2-1;

Rf3 = FUNCTION2(A3,8);

**If** (Ф3 == 1) **Then** C3 = Rf3+1;

**Else** C3 = Rf3-1;

**]**

**If**(INDEX==1) **Then**

**[**

Rf1 = FUNCTION2(A1,6);

**If** (Ф1 ==1) **Then** C1= Rf1+1;

**Else** C1= Rf1-1;

Rf2 = FUNCTION2(A2,8);

**If** (Ф2 == 1) **Then** C2 = Rf2+1;

**Else** C2 = Rf2-1;

Rf3 = FUNCTION2(A3,4);

**If** (Ф3 == 1) **Then** C3 = Rf3+1;

**Else** C3 = Rf3-1;

**]**

**If**(INDEX==2) **Then**

**[**

Rf1 = FUNCTION2(A1,8);

**If** (Ф1 ==1) **Then** C1= Rf1+1;

**Else** C1 = Rf1-1;

Rf2 = FUNCTION2(A2,6);

**If** (Ф2 == 1) **Then** C2 = Rf2+1;

**Else** C2 = Rf2-1;

Rf3 = FUNCTION2(A3,4);

**If** (Ф3 == 1) **Then** C3 = Rf3+1;

**Else** C3= Rf3-1;

**]**

**If**(INDEX==3) **Then**

**[**

Rf1 = FUNCTION2(A1,4);

**If** (Ф1 ==1) **Then** C1= Rf1+1;

**Else** C1 = Rf1-1;

Rf2 = FUNCTION2(A2,8);

**If** (Ф2 == 1) **Then** C2 = Rf2+1;

**Else** C2 = Rf2-1;

Rf3 = FUNCTION2(A3,6);

**If** (Ф3 == 1) **Then** C3  = Rf3+1;

**Else** C3 = Rf3-1;

**]**

C4 = FUNCTION1(A4, Ф4);

**Step5:** Previously, separated fraction part and transformed Cj are clubbed together with respective elements.

**Step6:** Accumulate all these bit encoded values of Cj in matrix Mbn**//**.

**Step7:** Apply reverse transform on Mbn**//** as per matrix operation (11) to produce the final bit fabricated matrix Mbn**///.**

**Step8:** If needed perform minor adjustments on Mbn// to keep the final bit fabricated elements of Mbn/// in spatial domain.

**Step9: t** ← **t + tr;**

**End Loop**

## Signature Fragment Bit Extraction Algorithm

**Input:** Authenticated host image with hidden signatures.

**Output:** Four copies of each signature sensed from the cover.

**Method:** For all regions signature coded sub-matrices Mbn/// is forward transformed to get Mbn// and from the elements fraction parts are separated. From the integer section one secret bit is extracted from matrix component Xj of Mbn//. Then These detected bits then packed in proper sequences to form those four signatures and just as coding the extraction of bit Фj from concern matrix element j ϵ {1,2,3,4}. let, the found matrix interval for the concern image segment is tr as mentioned previously, then the step-wise bit encoding on Mbn/ is as follows

**For matrix number t = 1 to L**

**Do**

**Step1:** Read the currently tracked matrix elements of Mbn**///** and use forward transform on them as per matrix operation (10) to produce the transformed matrix Mbn**///**. Then, separate fraction part of each elements which will give bit coded matrix elements Cj , for element index values j = 1,2,3,4.

**Step 2:’**

**If**(INDEX==0) **Then**

**[**

Rf1 = FUNCTION2(A1,4);

**If** (C1>=Rf1) **Then** Ф1= 1;

**Else** Ф1= 0;

Rf2 = FUNCTION2(A2,6);

**If** (C2>=Rf2) **Then** Ф2 = 1;

**Else** Ф2= 0;

Rf3 = FUNCTION2(A3,8);

**If** (C3>=Rf3) **Then** Ф3 = 1;

**Else** Ф3 = 0;

**]**

**If**(INDEX==1) **Then**

**[**

Rf1 = FUNCTION2(A1,8);

**If** (C1>=Rf1) **Then** Ф1 = 1;

**Else** Ф1= 0;

Rf2 = FUNCTION2(A2,6);

**If** (C2>=Rf2) **Then** Ф2 = 1;

**Else** Ф2= 0;

Rf3 = FUNCTION2(A3,4);

**If** (C3>=Rf3) **Then** Ф3 = 1;

**Else** Ф3 = 0;

**]**

**If**(INDEX==2) **Then**

**[**

Rf1 = FUNCTION2(A1,4);

**If** (C1>=Rf1) **Then** Ф1 = 1;

**Else** Ф1= 0;

Rf2 = FUNCTION2(A2,8);

**If** (C2>=Rf2) **Then** Ф2 = 1;

**Else** Ф2= 0;

Rf3 = FUNCTION2(A3,6);

**If** (C3>=Rf3) **Then** Ф3 = 1;

**Else** Ф3 = 0;

**]**

**If**(INDEX==3) **Then**

**[**

Rf1 = FUNCTION2(A1,4);

**If** (C1>=Rf1) **Then** Ф1 = 1;

**Else** Ф1= 0;

Rf2 = FUNCTION2(A2,6);

**If** (C2>=Rf2) **Then** Ф2 = 1;

**Else** Ф2= 0;

Rf3 = FUNCTION2(A3,8);

**If** (C3>=Rf3) **Then** Ф3 = 1;

**Else** Ф3 = 0;

**]**

**If** ((C4 Mod 3) == 0) **Then** Ф4= 1

**Else** Ф4= 0;

**Step3**: Collect these sensed bits in exact sequences.

**Step4:** t← t + tr;

**End Loop**

**4. Conclusion and Scope for Further Work**

The proposed work presents a noble data security scheme for validating digital e-certificates through firm achievement of the four milestones of data security- authentication, confidentiality, integrity and non-repudiation from the both server and client end. The ideology primarily focuses on digital validation at the time of reimbursement of monetary claim related to financial investments. The major research contributions are:

* The scheme is in strong compliance with authentication, confidentiality, integrity and non-repudiation to justify the validation process along with secure data transmissions.
* The dual phase validation process of signature embedding is supported by secure hash computations. In this aspect, the signature fabrication in each segment is based on two hash values determining the block interval of signature data embedding and the authentic circular orientation of signature fragments. This preserves better data integrity scenarios and ensues a new horizon regarding signature embedding concepts.
* The scheme also strengthens robustness and imperceptibility by variably encoding secret data on different transformed pixel bytes and maintaining multi-copy signature embedding.

With these advancements, the proposed approach confirms superiority over the existing approaches with exhaustive simulation results.

|  |  |
| --- | --- |
| **5.** | **REFERENCES** |

1. V. Anitha and R. Leela Velusamy, “Authentication of Digital Documents Using Secret Key Biometric Watermarking”, *International Journal of Communication Network Security*, Vol 1, Issue 4, 2012. ISSN: 2231 – 1882.
2. K. N. Mishra, “AAdhar based smartcard system for security management in South Asia”, *In the Proceedings of International Conference on Control, Computing, Communication and Materials (ICCCCM)*, pp. 1-6, 2016, E. ISBN**:** 978-1-4673-9084-2. doi**:** [10.1109/ICCCCM.2016.7918256](https://doi.org/10.1109/ICCCCM.2016.7918256)
3. H. R. Hasan, “Copyright Protection for Digital Certificate using Blind Watermarking Technique”, *Kurdistan Journal of Applied Research*, Vol 3, Issue 1, pp: 75-79, June 2018. ISSN: 2411-7706.
4. I. Nasir, Y. Weng, J. Jiang, S. Ipson, (2009). “Multiple spatial watermarking technique in color images”,Signal Image & Video Processing (SiViP*)*, Vol. 4, Issue. 2, pp. 145–154. doi: 10.1007/s11760-009-0106-7.
5. S. Behnia, M. Teshnehlab, P. Ayubi, “Multiple watermarking scheme based on improved chaotic maps”, *Communication in Nonlinear* *Science and Numerical Simulation*, Vol. 15, Issue. 9, pp. 2469-78. 2010, [doi:10.1016/j.cnsns.2009.09.042](http://dx.doi.org/10.1016/j.cnsns.2009.09.042)
6. G. Bhatnagar, Q. M. J. Wu, “A new robust and efficient multiple watermarking scheme”, *Multimedia Tools & Application*, Vol. 74, Issue. 19, pp. 8421-8444, 2013, doi:10.1007/s11042-013-1681-8.
7. M. Babaei, K. Ng, H. Babei, H. G. Niknajeh, “Robust multi watermarking scheme for multiple digital input images in DWT domain” *International Journal of Computer and Information Technology*, Vol. 3, Issue. 4, pp. 834-840, 2014
8. M. Natarajan, Y. Govindarajan, “Performance comparison of single and multiple watermarking techniques”, *International Journal of Computer Network and Information Security*, vol. 6, Issue. 7, pp 28-34, 2014. DOI: 10.5815/ijcnis.2014.07.04
9. N. Mohananthini, G. Yamuna, “Image fusion process for multiple watermarking schemes against attacks”, *Journal of Network Communications and Emerging Technologies*, vol. 1, Issue. 2, pp. 1–8, 2015. ISSN: 2395-5317.
10. N. Mohananthini, G. Yamuna, “Comparison of multiple watermarking techniques using genetic algorithms”, *Journal of Electrical Systems & Information Technology*, vol. 3, issue 1, pp.68–80, 2016, DOI: https://doi.org/10.1016/j.jesit.2015.11.009
11. R. Sadh, N. Mishra, S. Sharma, “Dual plane multiple spatial watermarking with self encryption”, *Sadhana, Indian Academy of Sciences*, Vol. 4, Issue. 1, pp. 1-14, 2016
12. Radharani, S., Valamathi, M.L. (2011), “Multiple Watermarking Scheme for Image Authentication and Copyright Protection Using Wavelet Based Texture Properties and Visual Cryptography” Int. Journal of Computer Applications (IJCA), 23(3), 29-36.
13. Thanki, R.M., & Borisagar, K.R. (2013), “Combined DCT–CS Theory Based Digital Watermarking Technique for Color Images” In Proc. of National Conf. on Emerging Trends in Information & Communication Technology (NCETICT 2013), Chennai, Tamil Nadu: Published by Int. Journal of Computer Applications (IJCA), 17-23.
14. A.K. Singh, B. Kumar, M. Dave, A. Mohan, “Multiple watermarking on medical images using selective DWT coefficients”, Journal of Medical Imaging and Health Informatics, vol. 5, Issue. 3, pp. 607-614, 2015.