International Journal of Engineering & Technology, 7 (3.8) (2018) 147-150

International Journal of Engineering & Technology

Website: www.sciencepubco.com/index.php/IJET

Research paper

HD-Sign: Hardware Based Digital Signature Generation Using True Random Number Generator

Anahita G , Krishnapriya KPM , Shiva Prasad R, Mohan Kumar N*

Department of Electronics and Communication Engineering, Amrita School of Engineering, Coimbatore, Amrita Vishwa Vidyapeetham, India
**Corresponding author E-mail: n_mohankumar@cb.amrita.edu*

Abstract

With the recent advancements in the field of computing, a fair share of easier and safer practices to exchange and share information between multiple parties have propped up. While some of these are improvisations, a few such as the Digital Signatures, have fast replaced conventional signing practices. It's wide use and acceptance in the industry as well as officially, has necessitated higher security to protect data integrity and privacy. These digital Signatures are generated on the basis of various schemes that are designed to accommodate efficiency, crypto security and algorithmic complexity. This paper proposes an alternate method named HD-SIGN for generating these digital signatures in accordance with Secure Hash Function and 512-bit SRNN cryptographic algorithm. With the aid of a TRNG module, a modification to produce a large number with two prime factors and a set of natural numbers in a pair of public and private keys has been incorporated. The LSFR based TRNG module which helps maintain the 'True Randomness' of any generated number has been used for this purpose. Further, the random nature of the generated sequence to be used in the digital signature, has been tested with the help of standard NIST tests. The Hamming distance has also been analyzed as a security metric for the proposal, implying the degree of unpredictability of the generated true random sequences.

Keywords: *Hardware Security; Digital Signature; Random number; TRNG; Public key cryptography; NIST*

1. Introduction

With the advent of technology in the global age, the use of traditional physical signatures have fast been replaced in most applications such as online trade, information exchange over a network or sealing decisions between partners of a corporate organization. These applications emphasize the importance and necessities of a public key cryptosystem to secure the information being transmitted or received, while maintaining complete security.

Digital signatures have been a stride towards this vision and fundamentally differ from hand written signatures in that they are never constant and act as a function of the document that signs it, while being dependent on the checksum which in turn relies on the time period of its production. While it also relies on the entire transmitted bit stream and a secret key, it can be checked without knowledge of the secret key. This projects the need to keep private keys [1] secure especially when a single, constant private key is employed for a long period of time, such as in the case of electronic cash keys and authority certifying keys. These Digital signatures are equipped with the capacity to detect tampering of data as well as to identify and authenticate an authorized official signatory. Further, the user in question has the added advantage of having evidence that the digital signature genuinely belongs to an authorized/ claimed signatory and is tamper proof.

Digital signatures work on asymmetric cryptography using a public key algorithm such as RSA [2]. A digital signature is created using signature software (such as an email program) which creates a one-way hash code of the electronic data to be signed. The private key is then employed to encrypt the hash. The encrypted hash encompassing essential information such as a hashing algorithm is together, the digital signature. It is sufficient to encrypt the hash instead of the entire message or document since a hash function converts an arbitrary input into a value of a fixed length that is relatively shorter, making hashing a viable option over signing.

This dependency of the signature - on the entire document, maintains the sanctity of the digital signature as it cannot be attached to any other document i.e., it is unique to a particular document and even minor changes or variations produced on it, will affect the digital signature. This implies that even a single character alteration or deletion would result in a different value, an attribute that helps validate data integrity using the public key of the signer as a tool to decrypt the hash. Since the digital signature is highly protected, receiver-sender integrity is maintained as it is unique and denial of usage on either party remains impossible viz., non-repudiation is strictly checked [3].

Digital signatures find applications in most modern email programs and online transactions that demand extreme scrutiny and security. In Pharmaceutical and surgical block chains which involve multiple levels of processing and manufacturing, these digital signatures have actively replaced traditional paper signatures for approval. Organizations such as digital signatures have been supporting the transition to replace conventional signing methodologies.

The NIST-approved digital signature algorithms like DSA, RSA, and ECDSA in conjunction with hash functions are used to generate and also to verify the generated digital signatures [4].

The latest version of the FIPS 186 (which was originally launched in 1994) FIPS 186-4 launched in 2013, allows more levity [5] on the use of random number generators while aligning with the Public Key Cryptography Standards. This paper proposes to capitalize on this volatility by incorporating an alternate method of using a novel TRNG scheme to generate random numbers, while keeping in line with the various protocols deployed status quo.

This paper proposes using an n-bit TRNG design which couples low complexity with high degree of randomness. The proposal is advantageous in terms of the lack of design ramification as it does not require any elaborate additions and in terms of maintaining pure randomness of generated sequences. Since the design is fundamental and exploits the obvious randomness available, it is a model that converts a disadvantage into an active security advantage. Keeping in line with the most recent web encryption algorithms available, no software end modifications would be necessary to employ the suggested method of signature generation. The system's security and complexity has been verified by the standard NIST test with results appended in this paper.

The paper is organized as follows: The first section provides an overview to the Digital Signature, followed by the second section which highlights the generation scheme involved. The third section brings forth the proposed architecture using TRNG and the fourth section provides a NIST test based verification of the method suggested. The final section documents the results and the conclusion.

2. Literature Survey

Digital Signature is a cryptographic primitive which is fundamental in authentication, authorization and nonrepudiation. Digital Signature is typically based a type of cryptography known as symmetric cryptography. The formulation of Digital Signature is based on a Digital Signature Algorithm [6], which consists of key generations such as private keys and public keys. The private key is used in the signature generation process and the public key is used in the signature verification process.

Schemes for Digital Signature Algorithms are based on parameters such as Efficiency, Anonymity Services, Security and Enhanced signing and verification capabilities [7]. Of the available schemes, the Batch scheme has been the most favourably opted, seconded by the Forward Secure scheme. In cases requiring multiple signing simultaneously, the batch scheme is most suitable. Employing the batch scheme requires that the digital messages are batched together prior to the implementation. The scheme employs random numbers to generate and authenticate the signers. To reconstruct the verification schemes, it is essential that the attackers must have knowledge of the random number scheme, which has proved to be practically infeasible.

A standard hash algorithm has a few intrinsic properties. It converts a message of any size into a digital string of a particular fixed length. This length is called the "hash length" or "L" and in accordance with RFC 4270 "To find a pair of messages M1 and M2 with coincident hash values takes 2^(L/2) attempts to crack". For a hash length of any reasonable length, this is a complicated process to solve. But employing tools such as the Hashcat and Aircrack-ng can brute force the attempts to crack the match. It is hence essential that the security of the documents that are digitally signed is more pervasive through the layers. In incorporating the suggested randomness, even a targeted hack could result only in coded data and non-decodable random bits of dis-similar patterns that render robotic spidering or tool spoofing useless.

As shown in the Fig 1, the original message which is the 'data' is converted into a hashed message using hashing function, upon which private key algorithm is implemented to generate a digital signature.

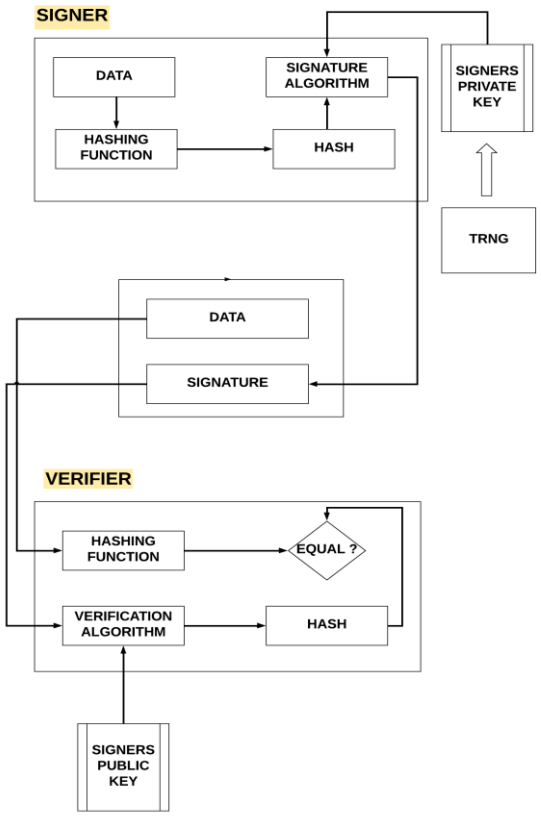

Fig.1: Digital Signature generation

For many applications, pseudorandom number generators (PRNGs)—which take a short random string with high or medium periodicity are quite satisfactory. Embedding unique signatures with minimalistic hardware or area overhead with security at three levels using a PRNG is proposed by Manoj Reddy et.al. [14] Simona Buchovecká et al.[8] proposed a TRNG design based on ROPUF circuit which utilizes ring oscillator as its source entropy. [9]Li Xiao-fei et al., talk about an algorithm to enhance the security of the random number. An Android based true random number generation architecture is proposed in Ashok. et.al.[16].In our paper, we further provide and analyze two improved scopes: (1) Enhancing the security of random numbers; (2) Establishing more complex links between the random number and the private key, so it is difficult for a hacker to use random number to indirectly attack the private key. The focus here is on enhancing the complex link between the random number and the private key. The proposal of the idea of including the TRNG module will highly increase the security of the system. In order to decrypt it, the third party will require computational capacity for cracking the random number segment because the source entropy for the TRNG circuits is natural jitters and noise. Hence, this will increase the computational complexity for a third party. The TRNG design we propose is based on ROPUF circuit.

3. Pseudo Algorithm

In this paper we focus on the generation of key to use the TRNG module. The proposed architecture for the implementation of the digital signature scheme using TRNG module is shown in Fig 2. This is a pre-designed standard method for Digital signature generation but incorporates the TRNG module as a novelty for additional efficiency and security.

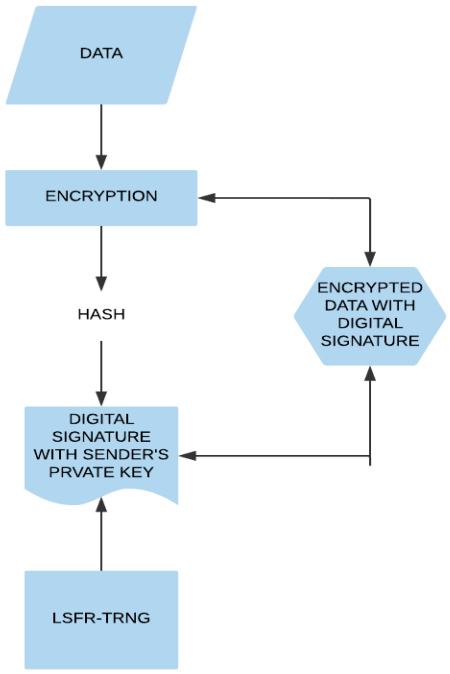

Fig.2: Proposed Digital Signature generation with TRNG module

The data that requires confidentiality is first encrypted with the help of a hashing function. The algorithm is similar to the SHA or the Secure Hash Algorithm [10] that is often employed in database management security events, in order to secure and store passwords and usernames from registration onwards. Once the hash is generated, it is appended to the Digital signature that also holds the sender's private key. The TRNG module aids in the creation of this true random private signature. The encrypted data with the digital signature is then sent to the receiver who receives this to decrypt and access.

The key generation algorithm starts with the generation of two prime numbers. The existing ROPUF based TRNG module will be an efficient method to generate these random numbers thereby increasing the efficiency and security of the system as it maintains randomness in that, no two generated bit-streams can match consecutively and the probability of decoding a 16 bit randomly generated bit stream is very meager. These advantages are inferred from the NIST test performed and using the Hamming distance as a metrics for comparison [11].

3.1 Proposed TRNG Architecture for HD-Sign

In this section, we adopt a n-bit True Random Number Generator design .The block diagram shown in Fig. 3. Illustrates the adopted randomly seeded LFSR based TRNG module. The 8-bit LFSR module consists of n/2 number of 8-bit LFSRs in parallel, a clock divider module consisting of four D-flip flops and a D-flip flop at the end of each LFSR to latch the output of the LFSR at the positive edge of the input clock signal. Thus this design facilitates throughput generation of n-bit random numbers by utilizing the model of n/2 8-bit LFSRs in parallel [12] [13]. The random number generator incorporated in the design is backed by the use of a seed generator which causes the S-R Latch to enter the metastability state when both the S and R inputs are simultaneously '1'. With each iteration, when the TRNG is invoked, a unique 'n' bit sequence is generated owing to the randomness caused by the seeding of the SR-latch based seed generator.

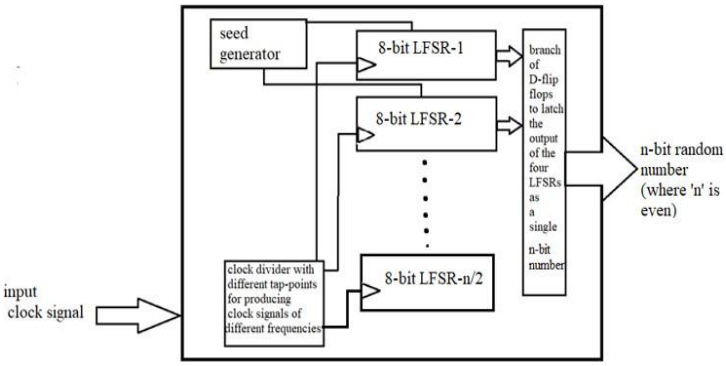

Fig.3: 'n' bit generator-TRNG module

4. Test Results and Analysis

This section elucidates the test results and simulation results of the randomness of the true random generator. The randomness of the sequences are tested by generating the design using NIST tests [13] [15] such as frequency mono-bit test, Runs test, Frequency test within a block, Longest run of ones in a block test, Overlapping template matching test, Non-overlapping template matching test, Cumulative sums test and Spectral Test. The following section illustrates the detailed results.

4.1 NIST Test Result for Proposed TRNG

The execution of the NIST tests was performed using sequences generated by design for multiple runs and the results are illustrated in Table 1.

Table 1: NIST results for 16 Bit TRNG Signature

Test Names	16 bit			
	Sequence 1	Sequence 2	Sequence 3	Sequence 4
Freq. Monobit Test	0.3213	0.6171	0.1336	0.3173
Freq. Test Within A Block	0.2428	0.3851	0.6382	0.5425
Runs Test	0.3642	0.4706	0.9742	0.1625
Longest Run of Ones in a Block Test	0.0000	0.0000	0.0000	0.0000
Overlapping Template Matching Test	0.8030	0.7011	0.9548	0.8645
Non-Overlapping Template Matching Test	0.0000	0.0000	1.0000	0.0000
Cumulative Sums (Cusum) Test	1.0000	1.0000	0.3627	1.0000
Spectral Test	0.1560	0.2561	0.7456	0.7456

By analyzing sequence 1, it is observable that each of the tests has provided a value that's in accordance with the algorithms used by for running them. Out of the eight tests, six tests have proved the sequence to be random in nature. The length of longest number of runs (ones) for the generated sequence is minimal as the binary values keep alternating. Hence, this test has failed. Similarly, Non-overlapping Template Matching test has also resulted in zero because this test rejects sequences exhibiting too many or too few occurrences of a given pattern. Since, the generated sequences switch in a random fashion, the high or low occurrence of a specific pattern is less.

Table 2: NIST results for 32 Bit TRNG Signature

Test Names	32 bit			
	Sequence 1	Sequence 2	Sequence 3	Sequence 4
Freq. Monobit Test	0.0133	0.6171	0.1336	0.1573
Freq. Test Within A	0.6603	0.7161	0.6660	0.9316

Block Runs Test	0.3473	0.975	0.0649	0.0083
Longest Run of Ones in a Block Test	0.0000	0.0000	0.0000	0.0000
Overlapping Template Matching Test	0.8949	0.8999	0.7249	0.6549
Non-Overlapping Template Matching Test	0.0000	0.0000	1.0000	0.0000
Cumulative Sums (Cusum) Test	1.0000	1.0000	0.3627	1.0000
Spectral Test	0.6464	0.2561	0.7456	0.1687

For each of these tests, a threshold value of 0.01 is considered such that if the value is greater than 0.01, the sequence is considered to be random. Hence, the results in the table prove that the sequence generated by the proposed design is random is nature.

4.2 Hamming Distance Based Signature Analysis

We have employed the security metrics as the Hamming distance to test for the obscuring complexity. The difference of bits available between the bit-streams of a few patterns has been listed and is applicable to the entire range of bit-streams generated.

Table 3 clearly indicates bit difference and a mean squared error of a minimum of 50%. It is the minimum desired value by the standard metric for data obfuscation. Thus it is inferred and seen that the security metrics that has been approved and employed viz., Hamming Distance is satisfied by the series of generated bit streams. This proves the obscurity achieved and the probability of decryption is minimal with security >50% in all cases.

Table 3: Hamming Distance for 16 bit

Sequence generated	Sequence on alternate run	Hamming distance	MSE
0000000000000000001011011011010110	0000000000000000000000000000000	9	56%
0000000000000000011100011001 0101	0000011000000000000000000000001	9	56%
0000000000000000000001110110 1001	0000000000000000000000000000011	8	50%
0000000000000000010111010101 1100	0000000000000000000000000000001	10	62%
0000000000000000110110010010 1000	0000000000000000000000000000001	8	50%
0000000000000000001100001100 0111	0011110000000000000000000000001	10	62%
0000000000000000100111000010 1110	0000000000000000000000000000001	8	50%
0000000000000000111001000000 1000	0001011000000000000000000000000	8	50%
0000000000011000010010001 1100	0000000000000000000000000111100000	8	56%

5. Conclusion

In this paper, we have attempted to combine a proficient True Random Number Generator (TRNG) into the generation of Digital Signature. The thought behind this consideration is to upgrade the security of the computerized signature algorithm, in this way making a boundary for programmers to create the same. For setting up this idea, the previously proposed LFSR-TRNG logic was incorporated. The utilization of TRNG will set up an unpredictable connection between the arbitrary number and the private key, consequently making THE HD-SIGN troublesome for the programmer to utilize random number to indirectly attack the private key.

References

[1] Negi, D., Negi, A. and Agarwal, S., 2016. The complex key cryptosystem. *Int. J. Appl. Eng. Res*, *11*, pp.681-684.

[2] Verma, Amit, Simarpreet Kaur, and Bharti Chhabra. "Design and Development of Robust Algorithm for Cryptography using improved AES technique "; *International Journal of Computer Science and Information Security* 15.3 (2017): 66.

[3] Faz-Hernández, Armando, et al. "A Secure and Efficient Implementation of the Quotient Digital Signature Algorithm (qDSA)." *International Conference on Security, Privacy, and Applied Cryptography Engineering*. Springer, Cham, 2017.

[4] McKay, Kerry A., Kerry A. McKay, Larry Bassham, Meltem Sonmez Turan, and Nicky Mouha. *Report on lightweight cryptography*. US Department of Commerce, National Institute of Standards and Technology, 2017.

[5] Polk, W. Timothy, et al. "Cryptographic algorithms and key sizes for personal identity verification." *NIST Special Publication* 800 (2015): 78-4.

[6] Bansal, Dimple, Manish Sharma, and Aayushi Mishra. "Analysis of Digital Signature based Algorithm for Authentication and Privacy in Digital Data." *International Journal of Computer Applications* 161.5 (2017).

[7] Chen FL, Liu WF, Chen SG, Wang ZH. Public-key quantum digital signature scheme with one-time pad private-key. Quantum Information Processing. 2018 Jan 1;17(1):10.

[8] Buchovecká S, Lórencz R, Kodýtek F, Bucek J. True random number generator based on ROPUF circuit. InDigital System Design (DSD), 2016 Euromicro Conference on 2016 Aug 31 (pp. 519-523). IEEE.

[9] Xiao-fei, L., Xuan-jing, S., & Hai-peng, C. (2010, April). An improved ElGamal digital signature algorithm based on adding a random number. In *Networks Security Wireless Communications and Trusted Computing (NSWCTC), 2010 Second International Conference on* (Vol. 2, pp. 236-240). IEEE.

[10] Dworkin, Morris J. *SHA-3 standard: Permutation-based hash and extendable-output functions*. No. Federal Inf. Process. Stds.(NIST FIPS)-202. 2015.

[11] Zhu, Chengzhang, Longbing Cao, Qiang Liu, Jianping Yin, and Vipin Kumar. "Heterogeneous metric learning of categorical data with hierarchical couplings." *IEEE Transactions on Knowledge and Data Engineering* (2018).

[12] Varchola, Michal. "FPGA based true random number generators for embedded cryptographic applications." *Dec* 1 (2008): 74-76.

[13] Shiva Prasad R., Anirudh Siripagada, Santthosh Selvaraj, Mohankumar N. "Random Seeding LFSR based TRNG for Hardware Security Applications", Proc. of 2nd Intl. Conf. on Integrated Intelligent Computing, Communication & Security (ICIIC-2018), 2018

[14] Manoj Reddy, Akshay K P, Giridhar R, Kharan SD, Mohankumar N, "BHARKS: Built-in hardware authentication using random key sequence, Proc of 4th IEEE Conference on Signal Processing Computing and Control (ISPCC), pp 200-204, Salon, 2017

[15] Tang, Q., Kim, B., Lao, Y., Parhi, K. K., & Kim, C. H. (2014, September). True random number generator circuits based on single- and multi-phase beat frequency detection. In *Custom Integrated Circuits Conference (CICC), 2014 IEEE Proceedings of the* (pp. 1-4). IEEE.

[16] Ashok Kumar Mohan; Dr. Nirmala Devi M.; Dr. M. Sethumadhavan; Santhya R.,"A Selective Generation of Hybrid Random Numbers via Android Smart Phones", International Journal of Pure and Applied Mathematics, Volume 118 (2018)