Received: 28 December 2021 | Revised: 28 April 2022 | Accepted: 2 May 2022

IET Quantum Communication

DOI: 10.1049/qtc2.12041

ORIGINAL RESEARCH

Digital signature technique with quantum-dot cellular automata

Arpita Kundu[1] | Bikash Debnath[2] | Jadav Chandra Das[3] | Debashis De[4]

[1]Institute of Engineering and Management, Kolkata, India

[2]Department of Computer Science and Engineering, Koneru Lakshmaiah Education Foundation, Andhra Pradesh, Guntur, India

[3]Department of Information Technology, Maulana Abul Kalam Azad University of Technology, Nadia, West Bengal, Simhat Haringhata, India

[4]Department of Computer Science and Engineering, Maulana Abul Kalam Azad University of Technology, Nadia, West Bengal, Simhat Haringhata, India

Correspondence

Arpita Kundu, Institute of Engineering and Management, 266/1, Gopal Lal Tahkur Road, Kolkata-700036, India.
Email: kunduarpita96@gmail.com

Abstract

Quantum-dot cellular automaton (QCA) is efficient nanotechnology that may be used as an alternative to Complementary Metal Oxide Semiconductor technology. Here, computation relies upon the electron's polarisation, revealing binary information. Quantum-dot cellular automaton is an appropriate opportunity for the upcoming age of advanced digital frameworks. Security in transferring data is essential since a lot of valuable information is present. Digital Signature is a process where data is transferred from a receiver to an authenticated sender only. In this paper, tile-based Exclusive NOR gate (XNOR) is designed, which is more stable than the regular majority gate-based circuit. It is used to create a novel circuit for authentication of data which is based on tiles. It develops a QCA architecture that works on the principle of Digital Signature. The architecture validates and proves the authentication of the message sent from the sender to the receiver. The decrypted digest and the converted digest of the original dispatch are compared in the proposed Digital Validator circuit, which is developed utilising a QCA XNOR gate. The security for communication at the nanoscale level is enhanced due to the use of the SHA-256 algorithm. The simulation results confirm the theoretical results.

KEYWORDS

asymmetric key cryptography, digital signature, energy dissipation, hashing, logic gates, QCA, tile-based

1 | INTRODUCTION

As indicated by the worldwide innovation guide for semiconductors, the Innovation Technology Roadmap for Semiconductors (ITRS) [1], which offers a precise analysis of future advancements, Quantum-dot cellular automaton (QCA) is one of the hopeful solutions for the future [2–4]. It has better performance regarding compactness of the device, clock cycle, and consumption of power; QCA is a powerful alternative to Very Large Scale Integration circuits [5–7]. QCA technology depends on Columbic repulsions. This repulsion generates due to the placement of electrons instead of a change in voltage levels. Cell polarisation reflects the data, which are regulated by inputs and clock signals. QCA is one of the recent technology which functions as a replacement for transistor-based circuits as new technology.

The primary computational unit of QCA is the QCA cell [8–10]. It is square in shape and has tetrad quantum wells. Inside these two tetrad wells, there are electrons. Due to Columbic repulsion, they dwell in diametric positions inside the cell. Due to polarisation, tunnelling activity occurs within the cell. As a result, QCA cells have two configuration states [11–15]. They represent the binary "0" and "1" values through polarisation of −1 and +1, respectively. The basic logic gate of QCA is a three-input majority voter gate (MV) composed of 5 QCA cells and NOT gate, that is, inverter (IV). The expression of MV is M (a, b, c) = ab + bc + ac. One of the three inputs of a two-input OR gate is set to polarisation +1, that is, binary 1, whereas one of the three inputs of an AND gate is fixed to polarisation −1, that is, binary 0 [16–18]. Here, we emphasise the tile-based design since they are fault tolerant and occupy less area and number of cells than gate-based designs.

QCA has the potential to generate newer systems to perform the Digital Signature process. A mathematical way of validating the integrity and authenticity of communication or digital document is known as a digital signature. It is the digital counterpart, but it has more security built-in. The purpose is to prevent tampering and fraud in digital communications. Security and its originality check are essential features required for a message to be transferred. Digital Signature checks the

originality of the message [19]. The two mutually authenticating cryptographic keys of public-key cryptography are used in digital signatures. The person who makes the digital signature encrypts signature-related data using a private key, and the signer's public key is the only method to decode that data.

If the signer's public key cannot access the document, there is an issue with the document or the signature. This is the method for verifying digital signatures. It safely reaches from the hand of the sender to the actual receiver, and the message remains intact. In this article, an approach for secure communication is shown through a digital signature performed by QCA technology and the proposed design for the validation of the message. The benefits of the paper are as follows:

- A tile-based Exclusive NOR gate (XNOR) is designed.
- A tile-based digital signature circuit is designed using QCA technology.
- Asymmetric key cryptography is achieved for the first time in QCA.
- Circuit complexity has been calculated.
- The results are validated against similar and dissimilar bits.

2 | PROPOSED WORK

A digital signature is asymmetric key cryptography [19]. It is explored in Figure 1. Asymmetric key cryptography consists of two keys. They are Public Key and Private Key. The sender possesses the private key, and the receiver contains the public key to decrypt the message. In Digital Signature, the message is encrypted in two phases. In the first step, the message is encrypted into a 256-bit format using the Hash Algorithm (SHA-256), known as digest [20]. The digest comprises 64 hexadecimal bits. The private key encrypts the digest bits in the second phase. Later the original message and the encrypted digest are sent to the receiver.

The receiver obtains both the original message as well as the encoded digest value. The receiver decrypts the digest value using the public key. On the other hand, the receiver converts the original message into a digest value using the same SHA-256 algorithm. Then both the digest values are decrypted, and generated digest is compared; if both of them are equal, then the message is authenticated; otherwise, if they are not the same, then this means that the message has been tampered by someone.

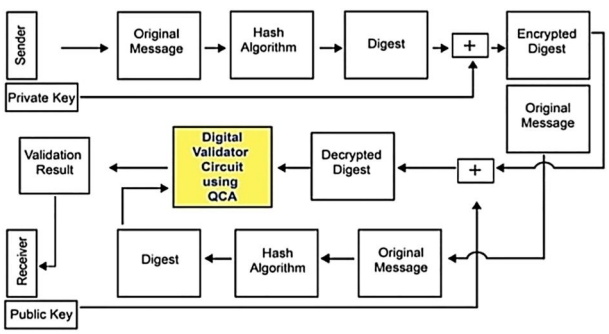

FIGURE 1 Block diagram of a digital signature process using a digital validator circuit

2.1 | Encryption

A message "*hello can you hide me still someone will find me I know him who can find me*" is considered whose digital signature is necessary to check its integrity. At first, it undergoes SHA256 conversion [21]. On conversion, it will generate 64 hexadecimal bit digest. The hexadecimal format obtained on applying the SHA-256 algorithm on the message is converted into"C73A4745861B8DEE35AD525D5DEE-DE4DC42422B52D9357500C5105B0266585F3" message digest. Now, if each of the digits is taken separately, the first one is "C", then "7", "3", "A," and so on. These hexadecimal bits are converted to binary bits. Hexadecimal to the binary conversion of the first four numbers are 1110, 0111, 0011, 1010, and its decimal values are 13, 7, 3, and 10, respectively. Now the Rivest-Shamir-Adleman (RSA) algorithm [12] is applied on each decimal value to encrypt the message. If p and q are two prime numbers, 13 and 11 are considered. Smaller values of p and q are taken to keep the calculation simple. N is the modulus obtained by multiplying the prime numbers together ($N = p \times q$). The value of N obtained is $13 \times 11 = 143$. Now the Euler's toitent ($\phi(N)$) is calculated by $\phi(N) = (p-1) * (q-1)$ that is, $\phi(N) = (13-1) * (11-1) = 12 \times 10 = 120$. Since asymmetric key cryptography requires a pair of private (d) and public (e), the keys are deduced. The private key (d) remains with the sender only, and the public key (e) is transferred to the receiver. The value of e is chosen to be 13 since, according to the RSA algorithm, it must lie within $1 < e < 120$ and satisfies two conditions; they are (i) e should be co-prime to $\phi(N)$ and (ii) e must implement as the public key exponent. Then $d \times e$ must satisfy the eqn. $d \times e = 1 \bmod (p-1; q-1)$. So the values of $d \times e$ must be equal to either $(120 \times 1) + 1$ $(120 \times 2) + 1$ $(120 \times 3) + 1$ $(120 \times 4) + 1$, … $(120 \times n) + 1$, which are 121, 241, 361, 481, .,120n + 1, respectively. If the value 481 is considered, then as mentioned previously, $e = 13$, the value of $d = 37$. All the values obtained by applying the RSA algorithm are deduced and kept in a Table 1.

The formula for calculating the ciphertext (c) using $c = m^d \bmod N$, where private key (d) = 37, digest (m) = 12, modulus (N) = 143 is $c = 12^{37} \bmod 143$, and ciphertext obtained is 12 as the calculation shown in Figure 2. Similarly, for each character of the digest, encryption is performed. From the 2nd character "73A4745861B8DE …5F3" of the digest to the last character of the digest, encryption is performed. Table 2 converts the first 10 characters of the hexadecimal code to

TABLE 1 Parameters of RSA algorithm

Private				Public	
P	q	$\phi(N)$	d	N	e
13	11	120	37	143	13

Abbreviation: RSA, Rivest-Shamir-Adleman.

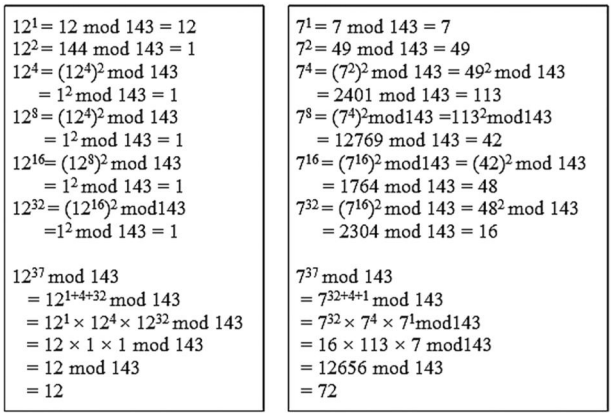

FIGURE 2 Conversion procedure of message to Cipher Text

TABLE 2 First 10 characters of the message encrypted to cipher text

Cipher Text	12	72	42	10	82	72	82	135	112	19
Decimal value	12	7	3	10	4	7	4	5	8	6
Message	C	7	3	A	4	7	4	5	8	6

encoded digest or ciphertext. Calculation of the first two message characters to its ciphertext conversion is shown in Figure 2. The congruence theorem is used to calculate it.

2.2 | Decryption

In the digital signature process, encrypted message digest and the original message are sent to the receiver. Thereafter the encrypted message digest is decrypted back to the original digest. The public key is required for the decryption procedure. Decryption occurs according to $m = c^e \bmod N$ where c denotes the ciphertext, e is the public key, m is the message, and N is the modulus. The values are $c = 12$, $e = 13$ and $N = 143$ as obtained. Thus $m = 12^{13} \bmod 143$, the value of the digest obtained is 12 and its hexadecimal value is 'C' as shown in Figure 3. Table 3 shows the first 10 ciphertext characters of the decoded digest converted back to the original digest. In this procedure total length of the ciphertext is decrypted back to the original message digest "C73A4745861B8DEE35AD525D5DEEDE4DC42422B52D9357500C5105B0266585F3". Since both the message and the decrypted message are transferred to the receiver, SHA256 is reapplied to the message to obtain the digest as obtained beforehand, which is"C73A4745861B8-DEE35AD525D5DEEDE4DC42422B52D9357500C5105B0 26658F3". Each hexadecimal value is converted to its corresponding binary values and compared with one other with the help of the QCA circuit designed. If on comparison, the values are the same, then the obtained results are all ones as displayed in Figure 8, which proves that the message is genuine. Calculation of the first two cipher characters to its conversion to message is shown in Figure 3. The congruence theorem is used to calculate it.

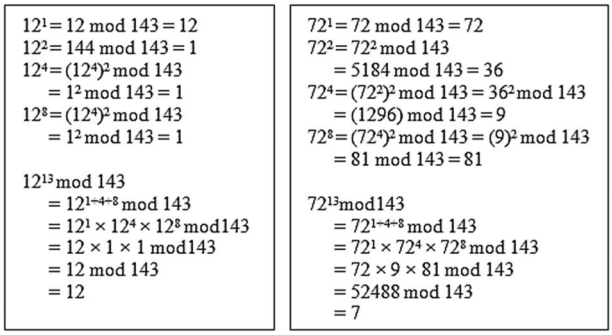

FIGURE 3 Conversion procedure of Cipher Text to message

TABLE 3 First 10 Characters of the ciphertext decrypted to digest

Message	C	7	3	A	4	7	4	5	8	6
Decimal value	12	7	3	10	4	7	4	5	8	6
Ciphertext (c)	12	72	42	10	82	72	82	135	112	19

3 | QCA DESIGN OF THE PROPOSED WORK

3.1 | Tile-based design

Tile structure comprises of NXN cluster of fully populated QCA cell arrays. The subtleties of the tile structure are examined in Ref. [24–29].

There are two kinds of tile-based designs. The active and passive tiles can both be utilised [26]. It is considered active if a tile implements a combinational logic function with at least two literals. If a tile implements only one literal's logic function, such as wire and Inverter (INV) functions, it is considered to be passive.

An n X n square grid of QCA cells can be used to create a tile. Both fully filled (FP) and non-completely populated (NFP) grids can be employed as fundamental logic blocks. In comparison to gate-based circuits, the clocking used is simpler. The 3 × 3 grid is particularly appealing for tile-based design because it enables various logic functions as illustrated in Ref. [26]. We suggested a 3 × 3 tile-based XNOR gate in QCA in this work. Exclusive-OR, XNOR, Parity checker, Multiplexer, and NAND-NOR Inverter are already available in tile-based QCA when two orthogonal tiles are combined.

In comparison to gate-based designs, tile-based designs are more efficient. The tile-based design needs fewer cells than the gate-based design as demonstrated in Ref. [26]. The basic logic primitive, as presented in this article, is the MV-like tile, which executes the majority function with selective inversion at the input. The MV-like tile gives an advantage in terms of area efficiency by integrating the features of MV and INV. It has been demonstrated that a tile-based design gains significant area and delay compared to a typical gate-based design [26]. In a tile-based design, all the components of the designs must be 3 × 3 tile-based.

3.2 | Building blocks

The XNOR [22] gate is the basic building block of the proposed Digital Signature Validator (DSV) circuit. It is not designed with the basic majority gates and NOT gates; instead, it has been developed using tile logic.

The block diagram of XNOR is explored in Figure 4a. It consists of two inputs and one output. Its circuit representation is displayed in Figure 4b. The inputs are A and B. In Figure 4b, the output obtained is $W = \overline{A} \cdot \overline{B} + A \cdot B$. The tile-based QCA layout for XNOR is exposed in Figure 4c. It requires 13 QCA cells and 0.02 μm^2area, which is simulated in the QCA Designer simulator [23].

In Figure 4d, the simulation result of the XNOR gate designed in Figure 4c is explored. The output W starts from the 2ndclock pulse. The inputs are A and B, respectively. Inputs provided to A and B are "00,110,011" and "01,010,101" , respectively. The output is "10,011,001" , which proves that when the two input bits are alike at that time, the result is "1" and when they are different, the output is "0".

3.3 | Digital signature validator

The block diagram of the DSV circuit is designed in Figure 5a. It consists of eight inputs and one output. The circuit diagram is revealed in Figure 5b. It is produced using four 3 × 3 tile-based XNOR gates as shown in Figure 4b. Additionally, it requires three AND gates. All are based on 3 × 3 tiles. The connectors connecting the four XNOR gates also use passive 3 × 3 tiles, used to interconnect as Ref. [26]. This circuit also has the advantage of efficiency in terms of area and delay compared to the gate-based circuits. For the eight inputs, the set of inputs A, B, C, and D, as shown in Figure 5b, is specified for a four-bit binary number achieved from the hexadecimal number of input bits of the decrypted digest obtained at the receiver end. The other four-bit binary number obtained from the hexadecimal number of the input bits of the message converted by the SHA-256 algorithm in the receiver part are E, F, G, and H, respectively, as shown in Figure 5b, and acts as another set of inputs. Each XNOR gate first checks whether two bits are similar or dissimilar, and the output of two XNOR gates is combined using an AND gate, such as, XNOR1 and XNOR2 are connected using the AND gate. Similarly XNOR3, and XNOR4 are integrated and indicated by the blue dotted lines. As AND gate generates "1" when both the inputs are "1", if any of the inputs from XNOR gates are "0", it will become "0". Once more, these two blocks of XNOR gates, containing two XNOR gates in each block, are combined using AND gate to obtain the final output. The output of the DSV circuit is equivalent to "1" only considering both the inputs provided to each four XNOR gates are similar.

The Quantum-dot cellular automaton (QCA) realisation of Digital Signature Validator (DSV) is also explored in Figure 6 from Equations (1–6). It consists of 97 cells and 0.13 μm^2area.

$$P = \overline{AE} + AE \tag{1}$$

$$Q = \overline{BF} + BF \tag{2}$$

$$R = \overline{CG} + CG \tag{3}$$

$$S = \overline{DH} + DH \tag{4}$$

$$DSV\_Out = (P.Q).(R.S) \tag{5}$$

$$DSV\_Out = (\overline{AE} + AE).(\overline{BF} + BF).(\overline{CG} + CG).(\overline{DH} + DH) \tag{6}$$

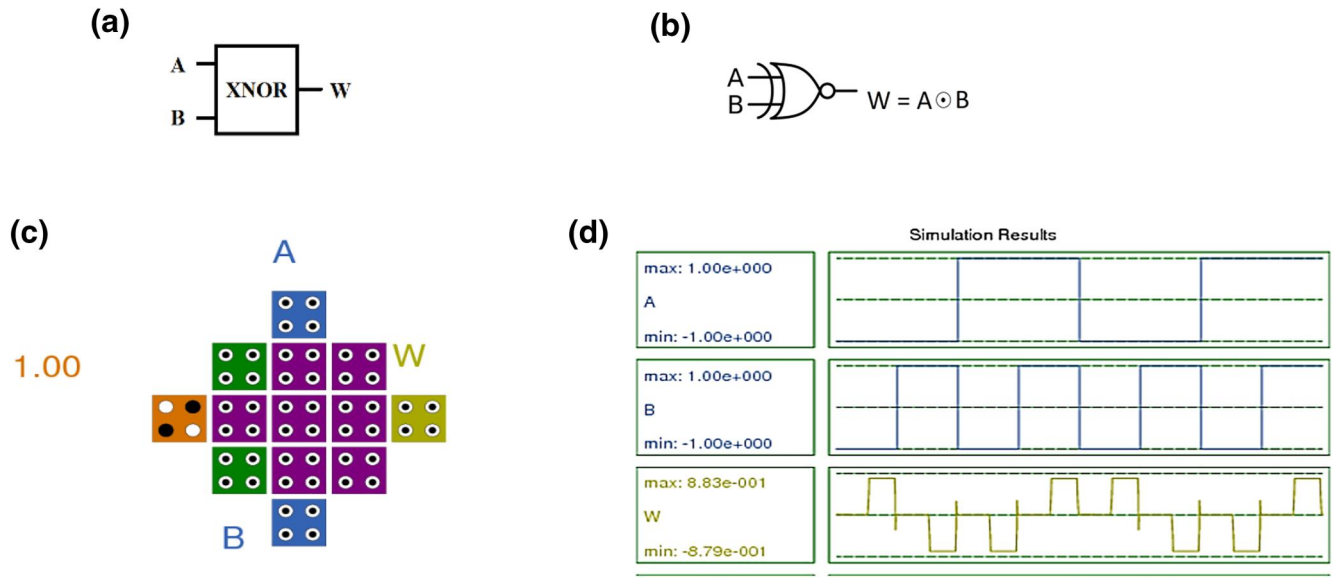

FIGURE 4 Exclusive NOR gate (XNOR) (a) Block diagram (b) Circuit representation, (c) Quantum-dot cellular automaton (QCA) realisation, and (d) Simulation results

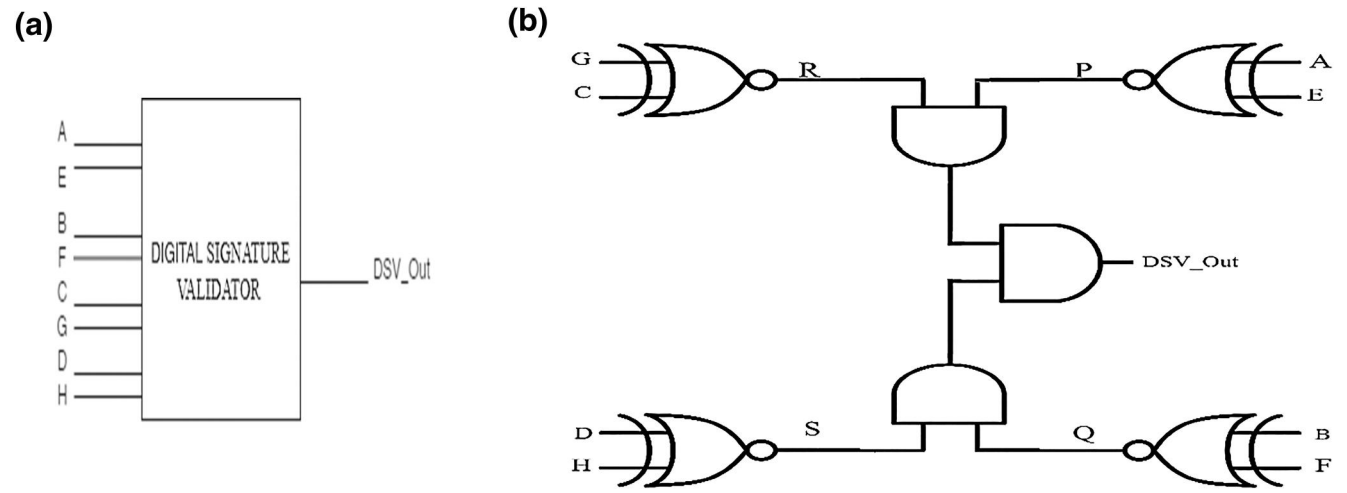

FIGURE 5 DSV (a) Block diagram (b) Circuit diagram

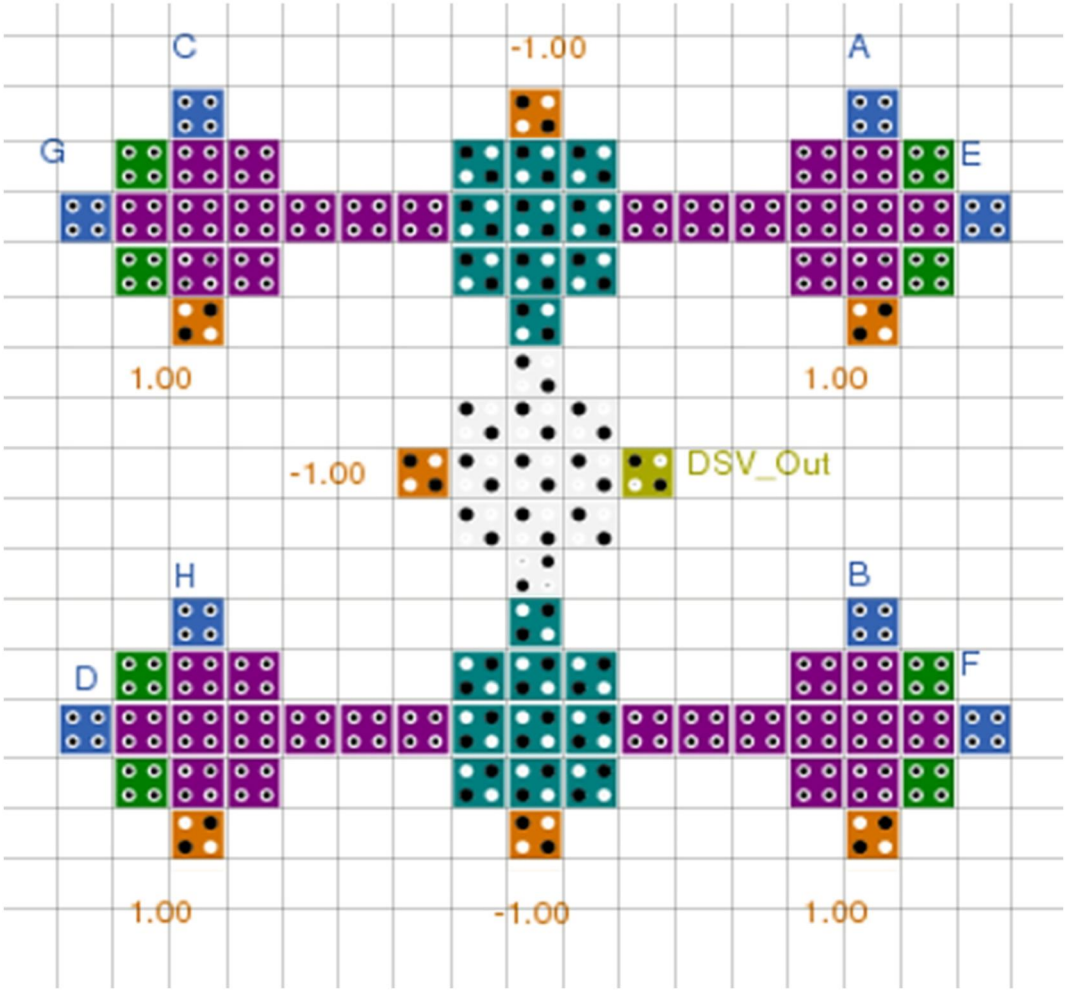

FIGURE 6 QCA realisation of Digital Signature Validator (DSV) architecture

In Figure 7 simulation result of the suggested DSV circuit is explored. Eight inputs are present in Figure 6. Four inputs A, B, C, and D, belong to one group, and E, F, G, and H belong to another group. The first four inputs denote the binary format of one hexadecimal digit, and the other four inputs denote the binary format of another hexadecimal digit. The first

hexadecimal digit denotes the value obtained on decrypting the encrypted digest. The second hexadecimal digit denotes the value obtained by applying the SHA-256 algorithm to the original data. For simulation, equal hexadecimal numbers are considered as inputs from each group. The same hexadecimal value for both sets of inputs is considered, that is, "9 A 9 A". The binary representation of "9 A 9 A" is "1001 1010 1001 1010". Then the input values for A = 1111, B = 0000,

TABLE 4 Complexity measurement

Proposed QCA circuit	# Cell	Total area (μm^2)	% Of acquired area	Latency (clock cycle)
XNOR gate	13	0.02	4.76	0.75
Digital signature validator	97	0.13	41.93	2.25

Abbreviation: XNOR, Exclusive NOR gate.

TABLE 5 Comparison table with the previous XNOR circuit

XNOR circuits	Design structure (Tile-Based/Gate Based)	Total cells	Total area (μm^2)	Usage in area (%)
Proposed XNOR gate	Tile-based	13	0.02	4.76
XNOR gate [30]	Gate-based	33	0.426	25.82
XNOR gate [31]	Gate-based	40	0.013	NS
XNOR gate [32]	Gate-based	32	0.25	44
XNOR gate [33]	Gate-based	25	0.02	25
XNOR gate [34]	-	13	0.01	4.2

Abbreviations: NS, Not Specified; XNOR, Exclusive NOR gate.

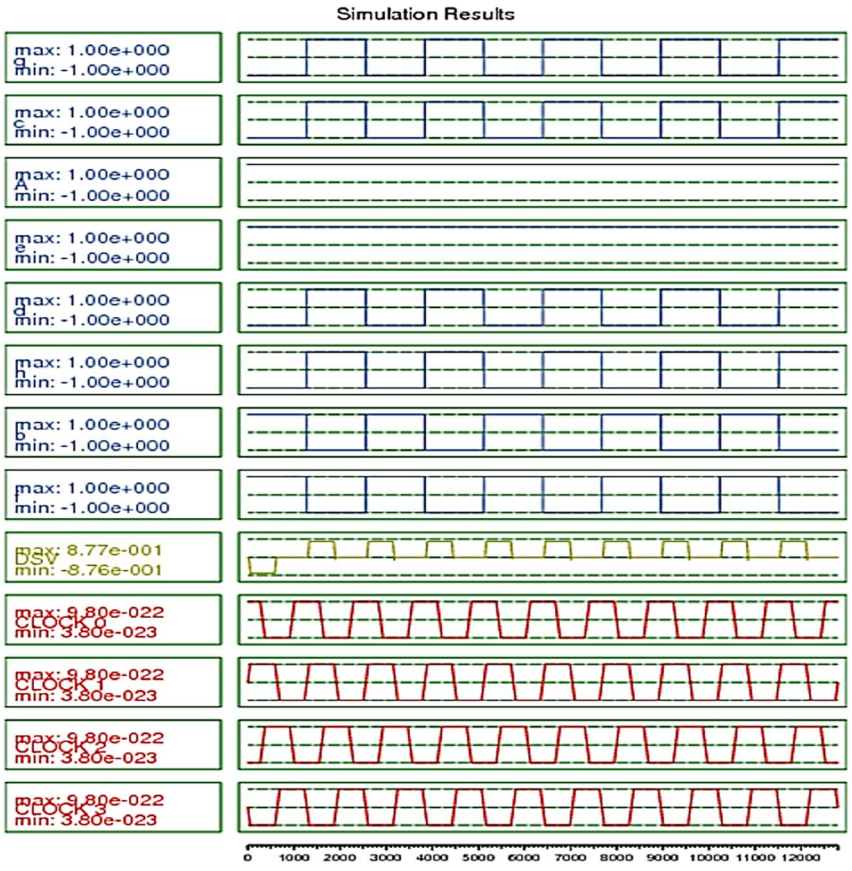

FIGURE 7 Simulation results of Digital Signature Validator (DSV)

C	7	3	A	4	7	4	5
1100	0111	0011	1010	0100	0111	0100	0101
8	6	1	B	8	D	E	E
1000	0110	0001	1011	1000	1101	1110	1110
3	5	A	D	5	2	5	D
0011	0101	1010	1101	0101	0010	0101	1101
5	D	E	E	D	E	4	D
0101	1101	1110	1110	1101	1110	0100	1101
C	4	2	4	2	2	B	5
1100	0100	0010	0100	0010	0010	1011	0101
2	D	9	3	5	7	5	0
0010	1101	1001	0011	0101	0111	0101	0000
0	C	5	1	0	5	B	0
0000	1100	0101	0001	0000	0101	1011	0000
2	6	6	5	8	5	F	3
0010	0110	0110	0101	1000	0101	1111	0011

FIGURE 8 Hexadecimal digest values converted to their binary equivalent of the original message

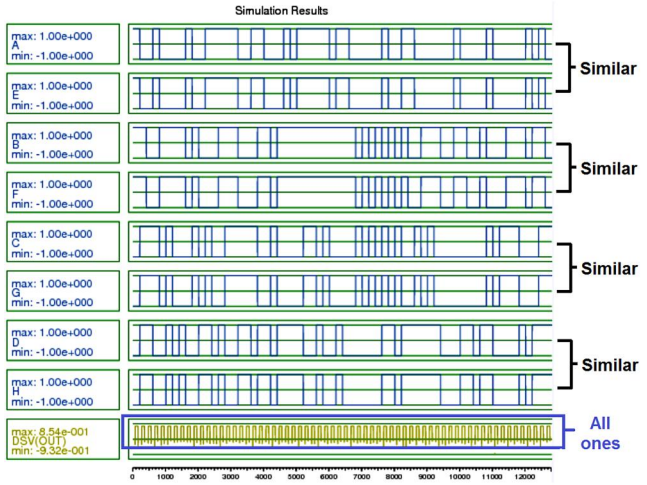

FIGURE 9 Simulation results of Digital Signature Validator (DSV) for similar bits

C = 0101, D = 1010 E = 1111, F = 0000, G = 0101, and H = 1010. A and E are input to XNOR1, B and F input to XNOR2, C and G are inputs to XNOR3, and D and H are input to XNOR4, respectively. The QCA DSV designed in Figure 6b shows that if and only if the input bits of the 1st pair A and E, 2nd pair B and F, 3rd pair C and G, and the last pair D and H, all are the same then only the output DSV(out) is 1. Since the sequence of input as mentioned in this example are similar for this reason, the output of the four XNOR gates is one, and when it applies to the AND gate, the final output (DSV (OUT)) value remains "1" for the given input sequence. The result follows the theoretical values.

3.4 | Complexity analysis

The circuit complexities of the proposed circuits are displayed in Table 4. It keeps the accounting details in terms of quantity of majority gates, amount of QCA cells used, the entire area occupied, cell area, % of the acquired area, and delay used.

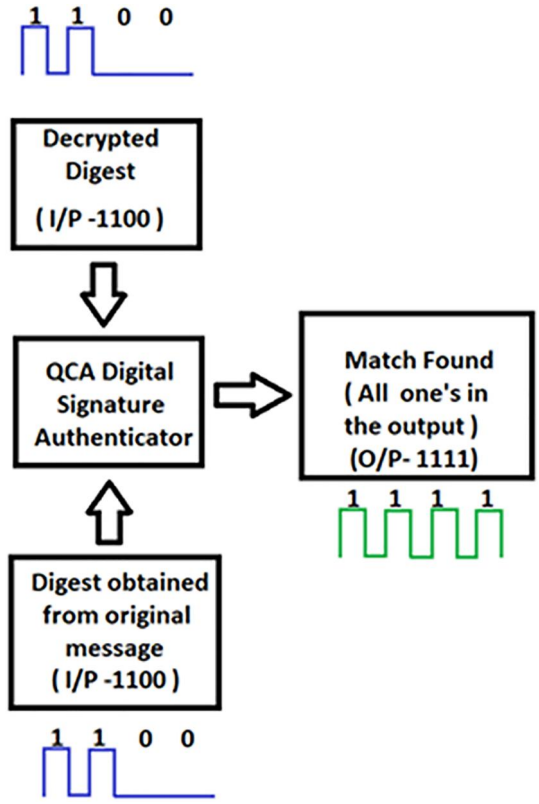

FIGURE 10 Validation of similar bits

3.5 | Comparison of the suggested QCA XNOR and the previous circuit

Table 5 shows the comparative analysis of the QCA XNOR gate presented in this paper to prior circuits.

The proposed circuit is based on fully populated 3 × 3 QCA tiles. From Table 5, it is observed that the proposed XNOR gate comprises less number of cells, the total area occupied is less, and the percentage of usage in the area is also less compared to other XNOR gates. As mentioned in Ref. [24–29], the tile-based designs are more stable than gate-based designs.

4 | VALIDATION OF THE PROPOSED WORK

4.1 | Validation for a similar message

Simulation is performed using the values obtained in the proposed example presented in section IIA. On application of SHA-256 algorithm on the original message the hexadecimal values are "C73A4745861B8DEE35AD525D5-DEEDE4DC42422B52D9357500C5105B0266585F3". The same data is obtained by encrypting the decrypted value. Each hexadecimal value is converted to its corresponding binary format, represented in Figure 8.

Now, each hexadecimal value is passed through two input groups after conversion into binary bits. They are ABCD and EFGH, respectively. Same binary values are passed through both groups. The first hexadecimal bit is "C" whose binary value is "1100". The corresponding values for inputs are A = 1, B = 1, C = 0, D = 0 for the first set of information and E = 1, F = 1, G = 0, H = 0 for the second set of inputs, respectively. A similar operation is performed for other 63 hexadecimal numbers obtained from Figure 8. Simulation is done, and the result is obtained in Figure 9. The simulation result inputs A and E, B and F, C and G, and D and H are similar. The output obtained is 64, proving that both the values are identical and the data is authenticated. Figure 10 shows the first letter of the decrypted digest and the digest obtained from the original message compared. And since both of them are similar, the output comprises all ones.

4.2 | Validation for dissimilar message

For validation check with dissimilar bits, simulation is performed with the values obtained from the proposed example and by changing the example. The text message is "***hello can you hide me still someone will find me I know him who can find me***". On application of SHA-256 algorithm on the original message the digest obtained is "C73A4745861B8DEE35AD525D5DEEDE4DC42422B52D9357500C5105B0266585F3". The hexadecimal to binary conversion is shown in Figure 8. If the original text is changed to ***"hello can you hideous still someone will find me I know him who can find me". In the above sentence, the " me " is changed to "us" in the 2nd statement.*** On applying SHA256 the total digest will convert to "EA9F6F6-B7352A07C1779A228CB8938BDD12F10F30373B4F3AA6-D13E4C228DC4F". Each hexadecimal value of the new digest is converted to its binary form, represented in Figure 11.

Each hexadecimal value is passed through two groups of inputs after conversion into binary bits. They are ABCD and EFGH, respectively. The first hexadecimal bit "C" from Figure 8, whose binary bit is "1100" then the corresponding values of A = 1, B = 1, C = 0, and D = 0 are the first set of

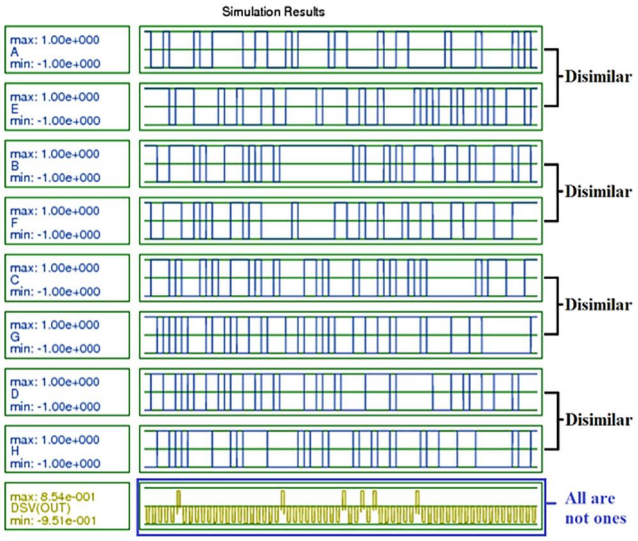

FIGURE 12 Simulation results of Digital Signature Validator (DSV) for dissimilar bits

E	A	9	F	6	F	6	B
1110	1010	1001	1111	0100	1111	0100	1011
7	3	5	2	A	0	7	C
0111	0011	0101	0010	1010	0000	0111	1100
1	7	7	9	A	2	2	8
0001	0111	0111	1001	1010	0010	0010	1000
C	B	8	9	3	8	B	D
1100	1011	1000	1001	0011	1000	1011	1101
D	1	2	F	1	0	F	3
1101	0001	0010	1111	0001	0000	1111	0011
0	3	7	3	B	4	F	3
0000	0011	0111	0011	1011	0100	1111	0011
A	A	6	D	1	3	E	4
1010	1010	0100	1101	0001	0011	1110	0100
C	2	2	8	D	C	4	F
1100	0010	0010	1000	1101	1100	0100	1111

FIGURE 11 Hexadecimal digest values converted to their binary equivalent of the changed message

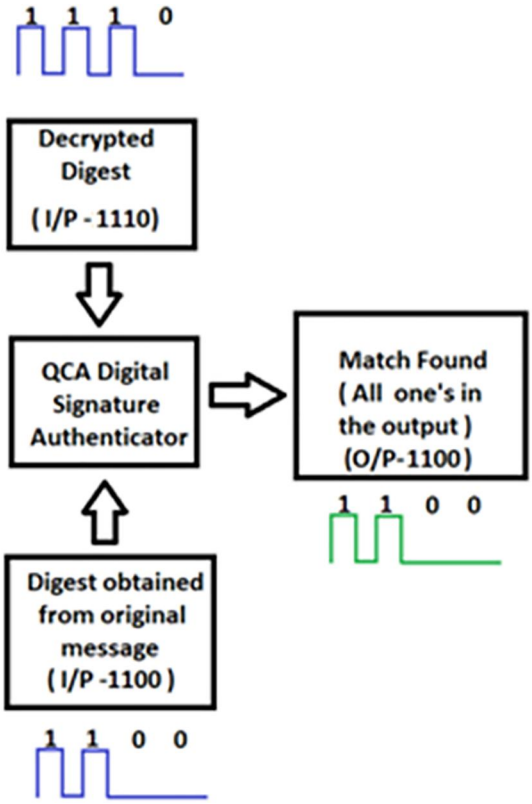

FIGURE 13 Validation of dissimilar bits

inputs. Whereas the first hexadecimal digit obtained from the changed message is "E". The binary bits obtained from Figure 11 corresponding to "E" is "1110". The corresponding values for inputs are E = 1, F = 1, G = 1, and H = 0 for the second set of information.

Similarly, other 63 hexadecimal numbers are deduced from Figures 8 and 11, respectively. Simulation is done, and the result is obtained in Figure 12. The simulation result shows that the inputs A and E, B and F, C and G, and D and H are dissimilar. The output obtained is maximum zeroes which proves that when both the input values are non-identical, the data is not authenticated. The digital signature is also invalid since the message has been changed. Figure 13 shows the first letter of the decrypted digest and the digest obtained from the original message compared and since both of them are dissimilar, the output doesn't comprise of all ones.

5 | CONCLUSION

In this article, a DSV circuit in QCA technology is presented. This circuit is utilised as one of the building blocks to achieve the architecture of the digital signature process. A 3 × 3 tile-based DSV circuit was obtained using 3 × 3 tile-based XNOR gates. The tile-based designs are fault tolerant and stable in comparison to gate-based methods. The security for communication at the nanoscale level is enhanced due to the use of the SHA-256 algorithm. The slightest change within the original message converts the digest. The architecture validates and proves the authentication of the message sent from the sender to the receiver. To perform the data validation, similar and dissimilar messages are provided as input to the DSV circuit. The simulation results confirm the theoretical results. In the future, the DSV circuit can be served for blockchain technology.

CONFLICT OF INTEREST

Declaration of Conflict of Interests

ORCID

Bikash Debnath https://orcid.org/0000-0002-5339-2293
Jadav Chandra Das https://orcid.org/0000-0001-5308-5077

REFERENCES

1. International Technology Roadmaps of Semiconductor (ITRS-2013)
2. Patidar, M., et al.: An Ultra-area-efficient Full Adder Circuits Design Based on Nanoscale QCA Technology, pp. 3713–3728. Design Engineering (2021)
3. Dhanush, T.N., Parashar, V.S., Premananda, B.S.: Analysis of QCA-Based Serial Concatenated Convolution Coding Encoder for Error Correction. In Recent Trends in Electronics and Communication. Lecture Notes in Electrical Engineering, vol. 777. Springer, Singapore. https://doi.org/10.1007/978-981-16-2761-3_37,2021
4. Safaiezadeh, B., et al.: Design and simulation of QCA-based 3-bit binary to gray and vice versa code converter in reversible and non-reversible mode. Optik. 251, 168464 (2022). https://doi.org/10.1016/j.ijleo.2021.168464
5. Debnath, B., Das, J.C., De, D.: Cryptographic models of nano-communicaton network using quantum dot cellular automata: a survey. IET Quantum Communication. 2(3), 98–121 (2021). https://doi.org/10.1049/qtc2.12013
6. Das, J.C., De, D.: Reversible priority encoder design and implementation using quantum-dot cellular automata. IET Quantum Communication. 1(2), 72–78 (2021). https://doi.org/10.1049/iet-qtc.2020.0009
7. Ahmadpour, S.S., Mosleh, M., Heikalabad, S.R.: The design and implementation of a robust single-layer QCA ALU using a novel fault-tolerant three-input majority gate. J. Supercomput. 76(12), 10185 (2020). https://doi.org/10.1007/s11227-020-03249-3
8. Majeed, A.H., et al.: Single-bit comparator in quantum-dot cellular automata (QCA) technology using novel QCA-XNOR gates. Journal of Electronic Science and Technology. 19(3), 100078 (2020). https://doi.org/10.1016/j.jnlest.2020.100078
9. Patel, U.P., Patel, A.K., Suthar, F.A.: The study of digital signature authentication process. Journal of Information, Knowledge and Research in Computer Science and Applications, 1(2) (2019). Issn: 0975 – 6728
10. Sadeghi, M., Navi, K., Dolatshahi, M.: Novel efficient full adder and full subtractor designs in quantum cellular automata. J. Supercomput. 76(3), 2191–2205 (2019). https://doi.org/10.1007/s11227-019-03073-4
11. Mohaghegh, S.M., Sabbaghi-Nadooshan, R., Mohammadi, M.: Designing ternary quantum-dot cellular automata logic circuits based upon an alternative model. Comput. Electr. Eng. 71, 43–59 (2018). https://doi.org/10.1016/j.compeleceng.2018.07.001
12. Lent, C.S., Tougaw, P.: A device architecture for computing with quantum dots. Proc. IEEE. 85(4), 541–557 (1997). https://doi.org/10.1109/5.573740
13. Ahmadpour, S.S., Mosleh, M., Heikalabad, S.R.: An efficient fault-tolerant arithmetic logic unit using a novel fault-tolerant 5-input majority gate in quantum-dot cellular automata. Comput. Electr. Eng. 82106548 (2020). https://doi.org/10.1016/j.compeleceng.2020.106548
14. Abedi, D., Jaberipur, G.: Decimal full adders specially designed for quantum-dot cellular automata. IEEE Transactions on Circuits and Systems II: Express Briefs. 65(1), 106–110 (2017). https://doi.org/10.1109/tcsii.2017.2703942

15. Chudasama, A., Sasamal, T.N., Yadav, J.: An efficient design of Vedic multiplier using ripple carry adder in Quantum-dot Cellular Automata. Comput. Electr. Eng. 65, 527–542 (2018). https://doi.org/10.1016/j.compeleceng.2017.09.019
16. Das, J.C., et al.: QCA based error detection circuit for nano communication network. IEEE Access. 7, 67355–67366 (2019). https://doi.org/10.1109/access.2019.2918025
17. Debnath, B., Das, J.C., De, D.: Nano-scale cryptographic architecture design using quantum dot cellular automata. Frontiers of Information Technology & Electronic Engineering 20(11), 1578–1586 (2019). https://doi.org/10.1631/fitee.1800458
18. Heikalabad, S.R., Salimzadeh, F., Barughi, Y.Z.: A unique three-layer full adder in quantum-dot cellular automata. Comput. Electr. Eng. 86, 106735 (2020). https://doi.org/10.1016/j.compeleceng.2020.106735
19. Herbert, H.C., Davis, D.L.: "Digital signature purpose encoding", U.S. Patent No. 6,199,053. (2001)
20. Wolrich, G.M., et al.: "Instruction set for message scheduling of SHA256 algorithm" U.S. Patent No. 8,838,997. (2014)
21. Burnett, S., Paine, S.: The RSA Security's Official Guide to Cryptography". McGraw-Hill, Inc. (2001)
22. Ahmad, N., Hasan, R.: A New Design of XOR-XNOR gates for low power application. In: International Conference on Electronic Devices, Systems and Applications (ICEDSA), pp. 45–49. IEEE (2011)
23. Walus, K., et al.: QCADesigner: a rapid design and simulation tool for quantum-dot cellular automata. IEEE Trans. Nanotechnol. 3(1), 26–31 (2004). https://doi.org/10.1109/tnano.2003.820815
24. Blair, E., Lent, C.: Clock topologies for molecular quantum-dot cellular automata. J. Low Power Electron. Appl. 8(3), 31 (2018). https://doi.org/10.3390/jlpea8030031
25. Chiu, H.N., et al.: PoisSolver: a Tool for modelling silicon dangling bond clocking networks. In: 2020 IEEE 20th International Conference on Nanotechnology, pp. 134–139. IEEE (2020)
26. Huang, J., et al.: Tile-based QCA design using majority-like logic primitives. ACM J. Emerg. Technol. Comput. Syst. 1(3), 163–185 (2005). https://doi.org/10.1145/1116696.1116697
27. Taucer, M., et al.: Consequences of many-cell correlations in clocked quantum-dot cellular automata. IEEE Trans. Nanotechnol. 14(4), 638–647 (2015). https://doi.org/10.1109/tnano.2015.2426058
28. Ganesh, E.N., et al.: Study and simulation of fault tolerant quantum cellular AutomataStructures. International Journal of Computer Theory and Engineering 2, 1793–8201 (2010). https://doi.org/10.7763/ijcte.2010.v2.254
29. Huang, J., Momenzadeh, M., Lombardi, F.: On the tolerance to manufacturing defects in molecular QCA tiles for processing-by wire. J. Electron. Test. Theory Appl. 23(2–3), 163–174 (2007). https://doi.org/10.1007/s10836-006-0548-6
30. Kavitha, S.S, Kaulgud, N.: Quantum Dot Cellular Automata (QCA) Design for the Realization of Basic Logic Gates", 2017 International Conference on Electrical, Electronics, Communication, Computer and Optimization Techniques. ICEECCOT. (2017)
31. Goswami, M., et al.: Efficient realization of digital logic circuit using QCA multiplexer. Proc. of the 2nd Intl. Conf. Bus. Inf.Manage, 165–170. (2014)
32. Sheikhfaal, S., et al.: Designing efficient QCA logical circuits with power dissipation analysis. Microelectron. J. 46(6), 462–471 (2015). https://doi.org/10.1016/j.mejo.2015.03.016
33. Raj, M., Gopalakrishnan, L., Ko, S.B.: Fast quantum-dot cellular automata adder/subtractor using novel fault tolerant exclusive-or gate and full adder. Int. J. Theor. Phys. 58(9), 3049–3064 (2019). https://doi.org/10.1007/s10773-019-04184-7
34. Wang, Lei, Xie, G.: A novel XOR/XNOR structure for modular design of QCA circuits. IEEE Transactions on Circuits and Systems II: Express Briefs. 1–1 (2020). https://doi.org/10.1109/TCSII.2020.2989496

How to cite this article: Kundu, A., et al.: Digital signature technique with quantum-dot cellular automata. IET Quant. Comm. 3(3), 164–173 (2022). https://doi.org/10.1049/qtc2.12041