Implementation of Anti-quantum Communication System using Software-Defined Radio

Hongjia Yang[1] , Jiarui Xu[1] , Haiyu Wang[1] , Chaofan Wen[1] , Guang Wu*[1]
[1]*Department of Electric and Electrical Engineering, Southern University of Science and Technology*
{11911214, 11911116, 12011331, 12012116}@mail.sustech.edu.cn, wug@sustech.edu.cn

***Abstract*—In recent years, the rapid development of quantum computer poses great challenges in the security of traditional wireless communication. This paper proposes a future-oriented high security anti-quantum wireless communication system, which can be applied to the mobile devices of Internet of things with constrained resources, and give them longer service life and flexibility on the basis of ensuring their security. Particularly, the Rainbow anti-quantum signature algorithm is implemented in Software-Defined Radio (Zedboard and AD9361 RF modules) platform and accelerated by Field Programmable Gate Array (FPGA) hardware of Zedboard. The experimental results show that the time of signature generation and verification is shortened significantly under the limitation of the consumption of computing resources. What's more, the proposed anti-quantum wireless communication system has better reconfigurability through flexible RF front-end configurations.**

***Index Terms*—Anti-quantum communication system, Software-Defined Radio, Rainbow**

I. Introduction

Recently, the development of quantum science strongly promotes the progress of quantum computer. Quantum bit, as the basic unit of quantum computer, can be easier to implement parallel operations in that it could be in multiple states at the same time. Compared with traditional computer, quantum computer has stronger computing ability. Thus, with a quantum computer, brute-force-attack is easier to succeed in most of the existing encryption algorithms, such as Data Encryption Standard (DES), EllipseCurve Cryptography (ECC) and Rivest-Shamir-Adleman (RSA) [1]. Certainly, with the development of quantum computer, existing cryptosystem, such as military secrecy, forward security and cyberspace security will face severe challenges. Developing some cryptosystems those can resist quantum computer attacks is of great significance.

Anti-quantum cryptography has been widely studied in recent years. Anti-quantum cryptosystem, also known as post-quantum cryptosystem (PQC), generally refers to a cryptosystem that can resist existing classical computing attacks and future quantum computing attacks, such as Shor's algorithm and Grover's algorithm. In the most existing cryptosystem, there are several cryptographic algorithms that cannot be easily cracked by quantum computers: Symmetric cryptography, Code-based cryptography, Lattice-based cryptography and Multivariate public key cryptography [2].

Multivariate public key cryptography, as one of asymmetric encryption algorithms, is based on the difficulty of the problem of solving multivariate quadratic equations over finite field, which is in generally NP-hard. Generally, multivariate public key cryptography operates in a small finite field, which occupies fewer computing resources than other encryption algorithms. The higher efficiency of multivariate public key cryptography makes it suitable for almost all kinds of hardware and devices, such as field programmable gate array (FPGA), general processor, radio frequency identification (RFID) tags and smart cards. In addition, it also can be applied to the development of block-chain to meet the communication security requirements of resource-constrained Internet of things (IoT) devices [3].

National Institute of Standards and Technology (NIST) launched the post-quantum cryptograph standardization in 2017. Some schemes of hardware implementation in resource constrained devices and FPGAs are summarized and evaluated [4]. In fact, earlier in 2011, Tang Shaohua et al. [5] proposed an efficient hardware implementation scheme of a post-quantum cryptography, that is the Rainbow signature algorithm, which optimized parallel hardware design of Gaussian elimination method, finite field multiplication and inverse. The implementation only needs 198 clock cycles to generate rainbow signature. This design optimized the hardware acceleration of Rainbow signature and verification, but did not consider for the resource-constrained wireless communication systems. Ahmed Ferozpuri et al. [6] developed the high-speed architecture of Rainbow signature algorithm on FPGA with parameterized system solver, reduced the number of clock cycles required, and tested it on Virtex-7 and Kintex-7 FPGA. However, this research still did not involve the testing and verification of communication system. Deepraj Soni et al. [7] designed three variants of Rainbow algorithm with different security strength by synthesizing using high-level synthesis tool. By evaluating the power, area, speed and security parameters trade-off of Rainbow hardware architecture, the research concluded that Rainbow has more delays in the process of key generation than in the process of signature generation and verification.

Wireless attacks take advantage of various vulnerabilities in radio hardware equipment or its communication protocol to carry out deception, eavesdropping, interference and replay attacks. Current communication widely uses traditional encryption method for authentication or information transmission.Once the current major wireless communication protocols such as 5G and WiFi are maliciously attacked, it will cause

serious consequences.

Software-Defined Radio (SDR) is the solution of the next generation wireless communication system. By modifying the software configuration parameters in the equipment, We can quickly switch between different communication protocols, such as 5G/4G-LTE and WiFi [8], and add other functions for them.

In this paper, we combine the hardware acceleration of anti-quantum algorithm with Software-Defined Radio implementation and propose a future oriented high security wireless communication system, especially for resource constrained IoT and mobile devices. This system gives devices long service life and high flexibility on the basis of ensuring their security, which is of great significance for the information security in the future quantum era. Specifically, this paper proposes and implements a new anti-quantum wireless communication system based on the hardware acceleration of Rainbow signature algorithm on FPGA and SDR platform. The main contributions include the following two aspects:

1) We realize the hardware acceleration of Rainbow signature generation and verification on the Zedboard platform, and indicate the efficiency of hardware acceleration by comparing with software signature.

2) Based on Zedboard and AD9361 Software-Defined Radio platform, we realize an anti-quantum communication system, and verify the effectiveness of the system by experiments.

II. Theoretical Analysis

A. Rainbow Cryptography and Finite Field

Rainbow signature algorithm belongs to multivariate public key cryptography and is based on multi-layer unbalanced Oil-Vinegar polynomials. Compared with other multivariate public key cryptography, Rainbow provides better performance, smaller key and signature [9]. Multivariate public key cryptography and other cryptography use finite field as a number field. A finite field is a non-null field containing only a limited number of elements. For example, let $\mathbb{Z}_{\mathrm{p}}$ be the set of integers modulo a prime number p, then $(\mathbb{Z}_{\mathrm{p}}, +, \times)$ is the field of integers modulo p. We can find that $\mathbb{Z}_{\mathrm{p}} = \left\{\overline{0}, \overline{1}, \cdots, \overline{p-1}\right\}, \overline{p-1} = p-1 \bmod p$. In this field, we have:

$$\begin{cases} \overline{p-n} + \overline{p-m} = \overline{p-n-m} \\ \overline{p-n} \times \overline{p-m} = m \times n \bmod p \\ \overline{p-n} + \overline{n} = \overline{0} \end{cases} \quad (1)$$

m and n should be less than p.

The result of the addition and multiplication of any two elements in a finite field is still in this field. Then we can extend the field of integers modulo a prime number p to a non-prime field $\mathrm{GF}\left(p^n\right) = \mathrm{GF}\left(p\right)[X] / (P)$. P is an irreducible polynomial with degree n. For each finite field, one or more polynomials can be selected as irreducible polynomials and different choices of irreducible polynomial will affect the efficiency of finite field operation. Polynomials with fewer items can significantly increase the efficiency of cryptography in hardware. It is often in the form of $X^n + aX + b$. Here, we use $p = 2$, $n = 8$ and $P = X^8 + X^6 + X^3 + X^2 + 1$.

Because the addition is based on the field $(\mathbb{Z}_2, +, \times)$, it only has two elements and follows the rule:

$$\begin{cases} 0 + 0 = 0 \\ 1 + 1 = 0 \\ 0 + 1 = 1 \end{cases} \quad (2)$$

This can be realized directly by XOR. The multiplication operation adopts the method based on the polynomial basis, which can calculate the results in one clock cycle. For the result of the multiplication of two elements in the finite field $GF(2^8)$, take a modular operation for each power term to the irreducible polynomials and obtain polynomials with degrees no more than 7 and the terms with coefficients exceeding 7 need to be added. In this way, the power term $x^{14}, x^{13}, ..., x^8$ can be reduced and contributed to each term of $x^7, x^6, ..., x^0$, so as to obtain the product. The finite field inversion adopts the inverter based on the Fermat theorem to decompose the $GF(2^8)$ inversion calculation into 7-element multiplication.

B. Rainbow Signature Algorithm

The message waiting for signature has hash value $y(Y_1, Y_2, ..., Y_{\mathrm{m}})$ with length m and $Y_1, Y_2, ..., Y_{\mathrm{m}}$ are elements of finite filed $GF(2^8)$. The signature is $x(X_1, X_2, ..., X_{\mathrm{n}})$ with length n.

The construction of the Rainbow is:

$$L_1 \circ F \circ L_2(x) = y \quad (3)$$

F is the center map transformation, L_1 and L_2 are affine transformations (vector addition and matrix-vector multiplication). "$\circ$" represents the transformation relationship.

Firstly, do inverse transformation of L_1 to y: $\bar{y} = L_1^{-1}(y) = F \circ L_2(x)$, $\bar{y}(\bar{Y}_1, \bar{Y}_2, ..., \bar{Y}_{\mathrm{m}})$. Specifically, the relationship for y and $\bar{y}$ is: $y = p\bar{y} + q$. And p is an $m * m$ matrix, q is a $1 * m$ matrix, that is:

$$\begin{bmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_{\mathrm{m}} \end{bmatrix} = \begin{bmatrix} p_{11} & \cdots & p_{1\mathrm{m}} \\ \vdots & \ddots & \vdots \\ p_{\mathrm{m}1} & \cdots & p_{\mathrm{mm}} \end{bmatrix} \begin{bmatrix} \bar{Y}_1 \\ \bar{Y}_2 \\ \vdots \\ \bar{Y}_{\mathrm{m}} \end{bmatrix} + \begin{bmatrix} q_1 \\ q_2 \\ \vdots \\ q_{\mathrm{m}} \end{bmatrix} \quad (4)$$

p and q belong to private key. Define $\bar{x}$ as $\bar{x} = L_2(x)$. $\bar{x}(\bar{X}_1, \bar{X}_2, ..., \bar{X}_{\mathrm{n}})$ is the result from affine transformation L_2 for signature x, and $\bar{y} = F(\bar{x})$.

Then, compute $\bar{x}$ from $\bar{y}$ and central map transformation F. Central map is a multi-layer Oil-Vinegar structure, composed of multi-layer unbalanced Oil-Vinegar polynomials. Each layer has several vinegar variables and oil variables, and all variables are elements of finite field $GF(2^8)$. For each layer, vinegar variables are known and oil variables need to be solved. Firstly, randomly choose values for vinegar variables of the first layer, and substitute them into multivariate polynomials of the first layer to solve for oil variables in that layer. If the equations have a solution, move to next layer. Otherwise, choose a new

set of values for vinegar variables. Then, the vinegar variables and oil variables of the first layer are connected together to be vinegar variables of the second layer, and are substituted into the polynomials to solve for oil variables of that layer. The later operations are similar until the last layer's oil variables are solved. Vinegar variables and oil variables of the last layer are connected to be $\bar{x}$. The number of vinegar and oil variables for each layer can be chosen but need to satisfy the relationship that the sum of oil variables' number from all layers is the same as the elements' number of $\bar{y}$.

Suppose the first layer has v_1 vinegar variables, noted as $(V_{1\_1}, V_{1\_2}, ..., V_{1\_v_1})$, and has o_1 oil variables noted as $(O_{1\_1}, O_{1\_2}, ..., O_{1\_o_1})$. The second layer has v_2 vinegar and $(V_{2\_1}, V_{2\_2}, ..., V_{2\_v_2}) = (V_{1\_1}, V_{1\_2}, ..., V_{1\_v_1}, O_{1\_1}, O_{1\_2}, ..., O_{1\_o_1})$, and has o_2 oil variables noted as $(O_{2\_1}, O_{2\_2}, ..., O_{2\_o_2})$. The central map transformation has two layers. Then $\bar{x}$ generated by central map is:

$$\begin{aligned}\bar{x} &= (V_{2\_1}, V_{2\_2}, ..., V_{2\_v_2}, O_{2\_1}, O_{2\_2}, ..., O_{2\_o_2}) \\ &= (V_{1\_1}, V_{1\_2}, .., V_{1\_v_1}, O_{1\_1}, O_{1\_2}, ..., O_{1\_o_1}, O_{2\_1}, O_{2\_2}, ..., O_{2\_o_2}) \\ &= (\bar{X}_1, \bar{X}_2, ..., \bar{X}_{\mathrm{n}})\end{aligned} \tag{5}$$

For each layer, according to central map transformation, we need to solve n linear equations and n is the oil variables' number of that layer. Thus, the total number of equations for the central map is equal to the elements' number of $\bar{y}$. Each equation has the form as follows (use the t'th equation of the s layer as an example):

$$\bar{Y}_{\mathrm{o_1+o_2+...+o_{s-1}+t}} = \sum_{\mathrm{i=1}}^{o_{\mathrm{s}}} \sum_{\mathrm{j=1}}^{v_{\mathrm{s}}} \alpha_{\mathrm{t\_ij}} O_{\mathrm{s\_i}} V_{\mathrm{s\_j}} + \sum_{\mathrm{i=1}}^{v_{\mathrm{s}}} \sum_{\mathrm{j=i}}^{v_{\mathrm{s}}} \beta_{\mathrm{t\_ij}} V_{\mathrm{s\_i}} V_{\mathrm{s\_j}} + \sum_{\mathrm{i=1}}^{v_{\mathrm{s}}} \gamma_{\mathrm{t\_i}} V_{\mathrm{s\_i}} + \sum_{\mathrm{i=1}}^{o_{\mathrm{s}}} \delta_{\mathrm{t\_i}} O_{\mathrm{s\_i}} + \eta_{\mathrm{t}} \tag{6}$$

$$\begin{bmatrix} \sum_{\mathrm{j=1}}^{v_{\mathrm{s}}} \alpha_{1\_1\mathrm{j}} V_{\mathrm{s\_j}} + \delta_{1\_1} & \cdots & \sum_{\mathrm{j=1}}^{v_{\mathrm{s}}} \alpha_{1\_o_{\mathrm{s}}\mathrm{j}} V_{\mathrm{s\_j}} + \delta_{1\_o_{\mathrm{s}}} \\ \vdots & \ddots & \vdots \\ \sum_{\mathrm{j=1}}^{v_{\mathrm{s}}} \alpha_{\mathrm{o_s\_1j}} V_{\mathrm{s\_j}} + \delta_{\mathrm{o_s\_1}} & \cdots & \sum_{\mathrm{j=1}}^{v_{\mathrm{s}}} \alpha_{\mathrm{o_s\_o_sj}} V_{\mathrm{s\_j}} + \delta_{\mathrm{o_s\_o_s}} \end{bmatrix} \begin{bmatrix} O_{\mathrm{s\_1}} \\ \vdots \\ O_{\mathrm{s\_o_s}} \end{bmatrix} = \begin{bmatrix} \sum_{\mathrm{i,j=1}}^{v_{\mathrm{s}}} \beta_{1\_\mathrm{ij}} V_{\mathrm{s\_i}} V_{\mathrm{s\_j}} + \sum_{\mathrm{i=1}}^{v_{\mathrm{s}}} \gamma_{1\_\mathrm{i}} V_{\mathrm{s\_i}} + \eta_1 + \bar{Y}_{\mathrm{o_1+...+o_{s-1}+1}} \\ \vdots \\ \sum_{\mathrm{i,j=1}}^{v_{\mathrm{s}}} \beta_{\mathrm{o_s\_ij}} V_{\mathrm{s\_i}} V_{\mathrm{s\_j}} + \sum_{\mathrm{i=1}}^{v_{\mathrm{s}}} \gamma_{\mathrm{o_s\_i}} V_{\mathrm{s\_i}} + \eta_{\mathrm{o_s}} + \bar{Y}_{\mathrm{o_1+...+o_{s-1}+o_s}} \end{bmatrix} \tag{7}$$

$\alpha, \beta, \gamma, \delta, \eta$ are all coefficients and belong to private public. The equation has linear, quadratic entries of vinegar variables, linear entries of oil variables, constants entries and multiplication entries but no quadratic entries of oil variables. Substitute vinegar variables into the equations and generate a linear polynomial of oil variables. After transposition and collecting similar terms, we write the equations as matrix form:

And we can solve the oil variables of that layer through Gauss elimination method.

Finally, after inverse affine transformation L_2 for $\bar{x}$, we generate signature x. $\bar{x} = gx + h$, and g, h belong to private key.

$$\begin{bmatrix} \bar{X}_1 \\ \bar{X}_2 \\ \vdots \\ \bar{X}_{\mathrm{n}} \end{bmatrix} = \begin{bmatrix} g_{11} & \cdots & g_{1\mathrm{n}} \\ \vdots & \ddots & \vdots \\ g_{\mathrm{n1}} & \cdots & g_{\mathrm{nn}} \end{bmatrix} \begin{bmatrix} X_1 \\ X_2 \\ \vdots \\ X_{\mathrm{n}} \end{bmatrix} + \begin{bmatrix} h_1 \\ h_2 \\ \vdots \\ h_{\mathrm{n}} \end{bmatrix} \tag{8}$$

To verify the authenticity of a signature x, $\bar{F}$ is the public given to the receiver, which is computed by $L_1 \circ F \circ L_2$. If $\bar{F}(x)$ is equal to message y, then the signature is accepted, otherwise rejected.

Although verifying the signature is easy, computing the public key can be not so simple. We should scrutinize the process of verifying. First, we have L_2 for $\bar{x}$, we generate signature $\bar{x}$ by ${L_1}^{-1}$

$$\begin{bmatrix} X_1 \\ X_2 \\ \vdots \\ X_{\mathrm{n}} \\ 1 \end{bmatrix} = \begin{bmatrix} g_{11} & \cdots & g_{1\mathrm{n}} & h_1 \\ \vdots & \ddots & \vdots & \vdots \\ g_{\mathrm{n1}} & \cdots & g_{\mathrm{nn}} & h_{\mathrm{n}} \end{bmatrix}^{-1} \begin{bmatrix} \bar{X}_1 \\ \bar{X}_2 \\ \vdots \\ \bar{X}_{\mathrm{n}} \\ 1 \end{bmatrix} \tag{9}$$

And as the same, for the L_1 for Y, with the equation (4), we have:

$$\begin{bmatrix} \bar{Y}_1 \\ \bar{Y}_2 \\ \vdots \\ \bar{Y}_{\mathrm{m}} \\ 1 \end{bmatrix} = \begin{bmatrix} p_{11} & \cdots & p_{1\mathrm{n}} & q_1 \\ \vdots & \ddots & \vdots & \vdots \\ p_{n1} & \cdots & p_{\mathrm{nn}} & q_{\mathrm{n}} \end{bmatrix}^{-1} \begin{bmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_{\mathrm{m}} \\ 1 \end{bmatrix} \tag{10}$$

focusing on each $\bar{Y}_k$, Rewrite polynomials in matrix form, we gather each lay oh vinegar and oil variables and 1 to form a new vector, and gather the private key $(\alpha, \beta, \gamma, \delta, \eta)$ to form a new matrix, and note the matrix as C[k], means that each layer is corresponding to each C[k], With the equation (6), for each Y_k we have equation(11).

then we put the equation (9) and (10) into equation(11) so that we can get the equation of Y_k, and we can easily find the public key with the whole equation.

$$\bar{Y}_{\mathrm{k}} = \begin{bmatrix} V_{k\_1} \\ V_{k\_2} \\ \vdots \\ V_{k\_\mathrm{v_k}} \\ O_{k\_1} \\ O_{k\_2} \\ \vdots \\ O_{k\_\mathrm{o_k}} \\ 1 \end{bmatrix}^T \begin{bmatrix} \beta_{k\_11} & \cdots & \beta_{k\_1\mathrm{v_k}} & \alpha_{k\_11} & \cdots & \alpha_{k\_\mathrm{o_k}1} & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & \cdots & \beta_{k\_\mathrm{v_k}\mathrm{v_k}} & \alpha_{k\_\mathrm{v_k}1} & \cdots & \alpha_{k\_\mathrm{v_k}\mathrm{o_k}} & \cdot \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdot \\ \gamma_{k\_1} & \cdots & \gamma_{k\_\mathrm{v_k}} & \delta_{k\_1} & \cdots & \delta_{k\_\mathrm{o_k}} & 1 \end{bmatrix} \begin{bmatrix} V_{k\_1} \\ V_{k\_2} \\ \vdots \\ V_{k\_\mathrm{v_k}} \\ O_{k\_1} \\ O_{k\_2} \\ \vdots \\ O_{\mathrm{k}\_\mathrm{o_k}} \\ 1 \end{bmatrix} \tag{11}$$

and we also have:

$$Y_{\mathrm{k}} = \begin{bmatrix} X_1 \\ X_2 \\ \vdots \\ X_n \\ 1 \end{bmatrix}^T L1{L_2}^T C[k] L_2 \begin{bmatrix} X_1 \\ X_2 \\ \vdots \\ X_{\mathrm{n}} \\ 1 \end{bmatrix} \tag{12}$$

So we can get the matrix $L_1{L_2}^T C[k] L_2$ as D[k], and we change the D[k] into an upper triangle matrix named $D_0[k]$,$D_0[k]$ is the public keys.

C. VHDL Implementation of Rainbow

We use VHDL hardware programming language to realize Rainbow signature algorithm [10]. In terms of programming structure, Rainbow algorithm is mainly realized by finite state machine for it has the characteristics of clear structure, easy control and easy synthesis. The system design block diagram is shown in Fig. 1. Rainbow and Gaussian elimination state machine are used as the top-level modules, which are connected with the components of private key, finite field multiplier and inverter.

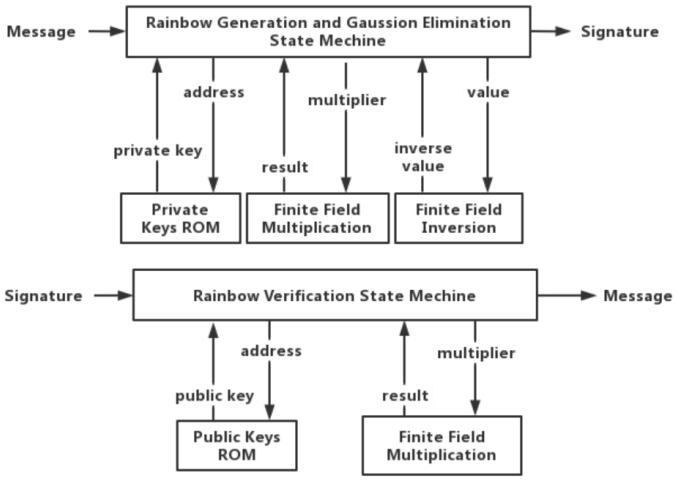

Fig. 1. Rainbow signature generation and verification block diagram

The private key component stores the private key $(\alpha, \beta, \gamma, \delta, \eta, p, q, g, h)$ required by the Rainbow algorithm. The private keys are arranged according to the order of use in generating signature. When it comes to the state where the private key input is required, the private key is retrieved from the private key component with an address that is added by 1 in sequence for use. The finite field multiplier and the inverse are implemented respectively according to the above method, accept the input and return the result $x(X_1, X_2, ..., X_{\mathrm{n}})$.

Rainbow and Gaussian elimination state machine has 35 states, mainly including the overall input, output, control module, L_1 and L_2 affine transformation modules, center map transformation module and Gaussian elimination module of the algorithm. The signature generation algorithm runs in the direction of the arrow in the state transition diagram in Fig. 2. Note that in order to simplify the transition diagram, the pointing arrow of the central control is omitted.

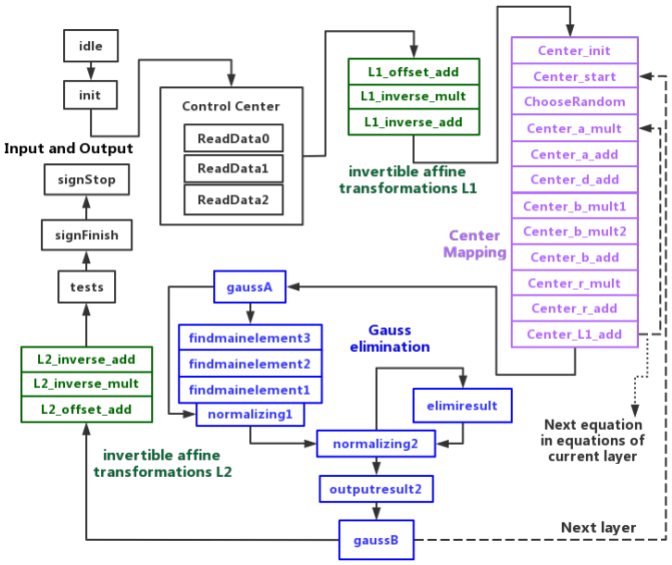

Fig. 2. State transition diagram

III. Implementation

We use Xilinx Zedboard and ADI AD9361 for communication experiment verification. Zedboard adopts XC7Z020-CLG484 FPGA in Zynq-7000 series , which includes processing system (PS) and programmable logic (PL) parts. The PS part adopts ADI-Kuiper-full operating system. Zedboard includes dual core Cortex-A9 MPcore, with a PL frequency

of 100MHz, 512 MB DDR3, 256 MB Quad-SPI Flash and 10/100/1000 Ethernet.

AD9361 is a high-performance and highly integrated RF transceiver for 3G and 4G base station applications. It has two independent direct frequency conversion receivers and has excellent noise parameters and linearity. AD9361 supports $2*2$ MIMO communication. The operating frequency range of AD9361 receiver local oscillator (LO) is 70 MHz to 6.0 GHz, the operating frequency range of transmitter LO is 47 MHz to 6.0 GHz, and the supported channel bandwidth range is 200 kHz to 56 MHz.

Zedboard and AD9361 are connected through FPGA mezzanine card (FMC) interface. The PL part drives AD9361 and configures the registers in AD9361 through FMCI, so as to control its modulation mode, carrier frequency, receive mode and other parameters. In this paper, AD9361 is controlled by ADI IIO Oscilloscope software.

In our experiment, one set of Zedboard and AD9361 are used as transmitter to transmit wireless signals, and the other set is used as receiver. The working process of our wireless communication system is: input the hash value of the message to be signed into transmitter, and the transmitter first completes the Rainbow signature generation of the hash value through hardware acceleration at the Zedboard PL and generate signature. The signature is sent to AD9361, which is first converted into bit data, and then mapped into constellation point coordinate data according to the selected modulation mode. After that, the constellation point data is written into the corresponding registers of AD9361. The AD9361 of the transmitter performs up sampling and pulse forming on the constellation point data and obtain the complex baseband waveform. Finally, it transmits the waveform through the antenna. The receiver AD9361 receives the signal sent by the transmitter, demodulates corresponding constellation point data through frequency offset correction, and then parses the bit data and converts it into signature. Then, the signature information is sent to Zedboard PL, and the hash value of the message is obtained through the Rainbow verification algorithm to complete the wireless communication process.

Fig. 3 shows the structure diagram of the whole system. The transmitter and receiver are connected with power, VGA display, keyboard, mouse and other peripherals, and connected with the control host through Ethernet Serial, Programming UART serial bus and Command control UART serial bus, so as to monitor the operation status. HDMI outputs the GUI interface of the operating system. VGA is driven by VHDL to display the Rainbow signature generation and verification information. Programming UART serial bus sends the FPGA burning program, and Command control UART serial bus receives the instructions from the control host to the operating system.

IV. Performance Analysis

We use a small-scale Rainbow structure to test. The hash value of the message to be signed is 4 bytes long. The Rainbow signature adopts Rainbow $(2^8, 2, 2, 2)$, which is a two-layer

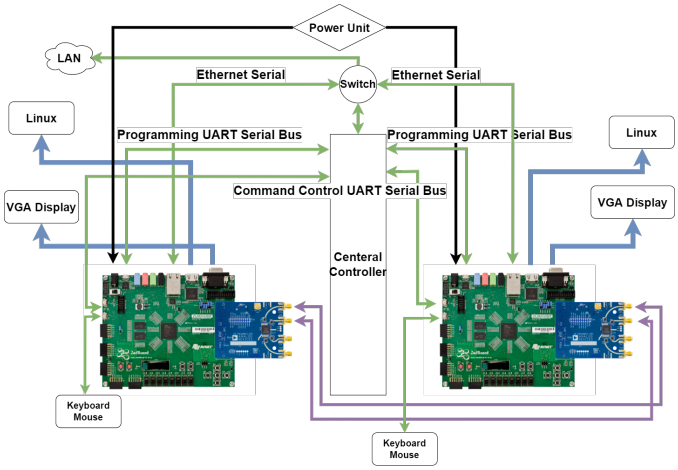

Fig. 3. System structure diagram

Oil-Vinegar structure. There are 2 vinegar variables and 2 oil variables in each layer, and the signature is 6 bytes long. The private key is 136 bytes long and the public key is 112 bytes long. Through VIVADO simulation and onboard verification, the results are shown in TABLE I.

It is observed that with the higher Maximum Freq, FPGA can use expend less time and take lower LUTs and FFs, because FPGA can finish one Clock Cycle with less time

For the same parameters, we test it on the PS part, using C language for Rainbow signature generation and verification. From the result, our FPGA hardware acceleration improves signature generation and verification by 76.2% and 90.5% respectively.

In the communication part from AD9361, we use 2.4GHz carrier, 6MHz bandwidth and 3 Mbps sampling rate to map the Rainbow signature generated by transmitter Zedboard PL to the constellation coordinate data of 4QAM, 16QAM, 64QAM and 256QAM for waveform transmission. We send 2048 symbols in each modulation. We export the constellation data received by the receiver AD9361 and obtain the constellation after normalization from MATLAB in Fig. 4. According to Fig. 4 we can find that though the constellation is not equidistantly spaced, the data representing the same symbol is still clustered together. It indicates that the communication is reliable as long as we use an appropriate symbol map table. We can use the location where each received point is aggregated as the standard symbolic mapping location. And the standard mapping table is shown in Fig. 4. According to Fig. 4 we can see the blue points represent the result of the receiver constellation, and the red points represent the correct QAM mapping. We can use error to represent the distance between received signal and the correct signal, it can be calculated by the following formula:

$$\boldsymbol{e}_{\mathrm{i}} = \|\boldsymbol{r}_{\mathrm{i}} - \boldsymbol{s}_{\mathrm{i}}\| \tag{13}$$

Here $\boldsymbol{r}_{\mathrm{i}}$ is the vector of the i-th received symbol, $\boldsymbol{s}_{\mathrm{i}}$ is the vector of correct mapping symbol, and $\boldsymbol{e}_{\mathrm{i}}$ is the error of the i-th symbol. The distribution of error is shown in Fig. 5. The error

TABLE I
RAINBOW SIGNATURE GENERATION AND VERIFICATION PERFORMANCE

Algorithm	FPGA Family	Clock Cycles	Maximum Freq	Execution Time	LUTs(Utilization%)	FFs(Utilization%)
Rainbow(Generation)	Zynq-7000	715	50 MHz	14.3 μs(in average)	9479(17.82%)	1966(1.85%)
Rainbow(Verification)	Zynq-7000	455	100 MHz	4.55 μs(in average)	1117(2.10%)	303(0.28%)
Rainbow(Generation)	Zynq processing system	-	667 MHz	60 μs(in average)	-	-
Rainbow(Verification)	Zynq processing system	-	667 MHz	48 μs(in average)	-	-

is quite small, it corroborates with the result in Fig. 4. It means the receiver correctly parses the signature and completes the signature verification, which indicates the feasibility of our system.

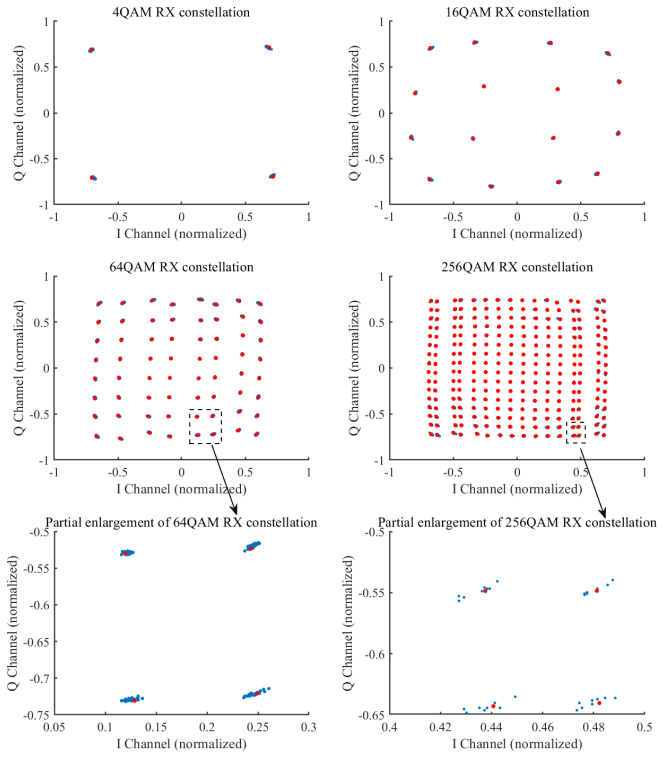

Fig. 4. Receiver constellation results and standard QAM mapping

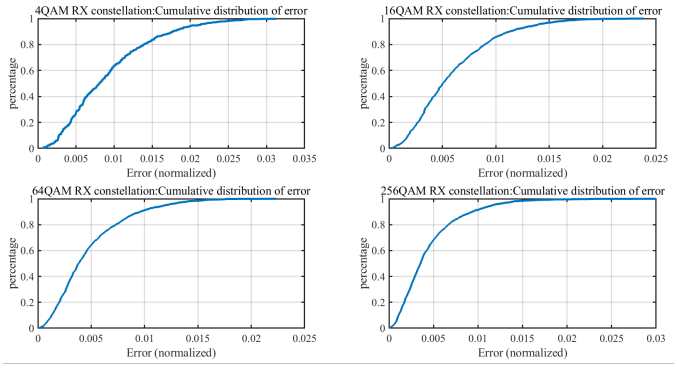

Fig. 5. Cumulative distribution of error under different QAM modulation

V. CONCLUSION

An anti-quantum communication system is proposed in this paper to ensure mobile devices of IoT with limited sources of security and give them longer service life and flexibility. The proposed system is based on Zedboard and AD9361 platform and achieve Rainbow signature algorithm hardware acceleration. Simulation and implementation results demonstrate that this communication system can reduce the computational resource consumption and improve flexibility of communication.

ACKNOWLEDGMENT

This work was supported by the Special Funds for the Cultivation of Guangdong College Students'Scientific and Technological Innovation.("Climbing Program"Special Funds).pdjh2022c0021.

REFERENCES

[1] V.Mavroeidis, K.Vishi, M.D.Zych and A.Jøsang, "The impact of quantum computing on present cryptography", arXiv preprint arXiv: 1804.00200,2018.
[2] D.J.Bernstein and T.Lange,"Post-quantum cryptography," Nature, vol. 549, pp. 188–194, Jan. 2017.
[3] J. H. Khor, M. Sidorov and P. Y. Woon, "Public Blockchains for Resource-Constrained IoT Devices—A State-of-the-Art Survey," in IEEE Internet of Things Journal, vol.8, no.15, pp.11960-11982, 1 Aug.1, 2021, doi: 10.1109/JIOT.2021.3069120.
[4] M.Seliem, K.Elgazzar and K.Khalil, "Towards privacy preserving IoT environments: A survey", Wireless Commun. Mobile Comput., vol.2018, pp.1-15, Nov.2018.
[5] S.Tang, H.Yi, J.Ding, H.Chen and G.Chen, "High-speed Hardware Implementation of Rainbow Signature on FPGAs" in International Workshop on Post-Quantum Cryptography, Springer, pp.228-243, 2011.
[6] A.Ferozpuri and K.Gaj, "High-speed FPGA Implementation of the NIST Round 1 Rainbow Signature Scheme," 2018 International Conference on ReConFigurable Computing and FPGAs (ReConFig), 2018, pp.1-8.
[7] D.Soni, K.Basu, M.Nabeel, M.Manzano, N.Aaraj and R.Karri, Hardware Architectures for Post-Quantum Digital Signature Schemes. Springer, 2021.
[8] José de Jesús Rugeles Uribe, Edward Paul Guillen and Leonardo S. Cardoso, "A technical review of wireless security for the internet of things: Software defined radio perspective", Journal of King Saud University - Computer and Information Sciences, 2021, [online] Available: https://doi.org/10.1016/j.jksuci.2021.04.003.
[9] J.Ding and D.Schmidt, "Rainbow, a new multivariable polynomial signature scheme," in Proc. ACNS, 2005, pp.164–175.
[10] H.Yi, S.Tang, H.Chen and G.Chen, "Fast Implementation of Rainbow Signatures via Efficient Arithmetic over a Finite Field" in Electrical Engineering and Control, Springer, pp.89-96, 2011.