AN EFFICIENT PRIVATE-PRESERVING VLSI ARCHITECTURE USING PAILLIER CRYPTOSYSTEM

1. R.TAMILSELVI, AP/ECE-MPNMJ ENGINEERING COLLEGE 2. R.RANJANI-FINAL ECE-MPNMJ ENGINEERING COLLEGE
3. M.SUBIKSHA-FINAL ECE- MPNMJ ENGINEERING COLLEGE 4.M.HARITHA-FINAL ECE-MPNMJ ENGINEERING COLLEGE
5. S.MEENA-FINAL ECE- MPNMJENGINEERING COLLEGE

Abstract

Recently, compatibility between privacy protection and information utilization has been considered to be a major challenge. Hence, researches on homomorphic crypto-system which can realize secure computation draw growing attention. The Paillier encryption with additive homomorphism is widely used in fields such as electronic voting and secure biometrics. At the same time, the large amount of data in the database and the requirements of Paillier security require computationally expensive processes each authentication and comparison, so the improvement of computational efficiency is particularly important. In this research, in order to reduce the computation time of Paillier, we design an ASIC based hardware architecture dedicated to Paillier encryption, and realize high-speed encryption and decryption by using high radix arithmetic core and the parallelism in the decryption process.

1. Introduction

Recently, with the rapid development of IoT, the communication of information has become more efficient, but at the same time, the problem of increasing security risks has gradually received attention. In the previous encryption mode, if the third party needs to perform statistics on the encrypted data in the database or perform authentication comparison with the new data, it is often necessary to decrypt the ciphertext first, and use the decrypted data to obtain the result. If the three parties are not trusted or attacked by the outside world, the original information in the database will be leaked.

In response to this problem, in recent years, the special cryptographic form which is possible to perform a series of operations on the ciphertext in the cipher domain through the encrypted data to achieve statistical, that is, the homomorphic encryption that can achieve the secret calculus has gained people's attention. This special cryptosystem can be used in many areas such as e-commerce, e-health and secure biometric systems while protecting the confidentiality of users' personal information.

Homomorphic cryptography consists of partial homomorphic algorithms and fully homomorphic algorithms. For fully homomorphic encryption algorithms, Gentry proposed a lattice-based homomorphic encryption scheme, which supports both addition and multiplication homomorphism. However, the noises added by the encryption process in fully homomorphic algorithms increase rapidly as the calculation progresses, and once the threshold is exceeded, the decryption results are unreliable making it difficult to implement the complex verification calculations required by biometrics.

The partially homomorphic encryptions with additive homomorphism such as Paillier algorithm, with low noise and the advantage of calculating lightweight, have the flexibility to construct a more complex homomorphic operations, which has been verified in many applications. But for the huge amount of data in the database, the total time required for a single authentication will still grow significantly with the volume of the database. Since the security level and computational efficiency of encryption are the two conflicting goals, there are often some performance losses in authentication systems that consider security issues. It is necessary to find a solution to improve the efficiency of the operation. In this study, we mainly pay attention to the Paillier Encryption with additive homomorphism. In order to improve the computational efficiency while ensuring security, in this research, with the 65-nm CMOS standard cell library, we designed a 1024-bit Paillier circuit with 256 radix arithmetic.

This paper is organized as follows. Section 2 shows algorithms of Paillier cryptosystem, modular exponentiation and Montgomery multiplication adopted in this study. In section 3, circuit design of Paillier crypto-processor is shown. In Section 4 the results are presented and compared and Section 5 concludes this research.

2. Algorithm

Paillier cryptosystem

Paillier is a public key cryptosystem proposed by Pascal Paillier in 1999 [1]. Paillier is a probabilistic asymmetric algorithm for public key cryptography. The problem of computing n-th residue classes is believed to be computationally difficult. The decisional composite residuosity assumption is the intractability hypothesis upon which this cryptosystem is based.

Let input text be m, public key be G and N, private key

be λ and μ, cipher text be C, R is selected randomly in $R \in Z^*_{N^2}$, then the Paillier encryption function *Enc (m)* is defined as follows.

$$C = Enc(m) = G^m \times R^N \bmod N^2 \quad (1)$$

The decryption function *Dec (c)* is defined as follows.

$$m = Dec(C) = L(C^{\lambda} \bmod N^2) \cdot \mu \bmod N \quad (2)$$

Paillier algorithm has the additive homomorphism that the product of two ciphertexts becomes a new ciphertext of the ciphertext of the sum of the two origin plaintexts as follows:

$$E(m_1) \cdot E(m_2) \bmod N^2 = E(m_1 + m_2) \bmod N^2 \quad (3)$$

Modular exponentiation

In 1024bit Paillier, *m, N, μ, λ* are 1024 bit numbers and *C, G, R* are 2048 bit numbers. Like the algorithm for the Paillier Encryption in the previous chapter, the Paillier homomorphic cryptosystem is based primarily on modular exponentiation. It includes a modular operation and integer operations, including encryption, decryption, and the same state during operation and modular multiplication modulo N^2 and N of the modular exponentiation arithmetic. These operations take a lot of time in a security-based (key size higher than 512-bit) operand environment. The implementation of modular exponentiation is often achieved by multiple modular multiplication operations. In hardware implementation, an efficient implementation algorithm for modular exponentiation is a Left-to-Right binary modular exponentiation algorithm. Since the operation performed when the index is 0 or 1 is different, such an algorithm exhibits data correlation in both time and power consumption, and is vulnerable to simple power consumption attacks and time attacks.

The Montgomery Power Ladder algorithm, as shown in Algorithm 1, eliminates the correlation between the operation and the exponent by introducing a certain redundancy, ie, whether the exponent is 0 or 1, a modular multiplication operation and a modular squaring operation will be performed. This will resist simple power attacks and time attacks.

Montgomery multiplication

In the cryptographic operations required to process large integers of hundreds to thousands of bits required by the Paillier cipher, the division at the time of modulo requires a large amount of time. Montgomery multiplication [2] is widely used as an efficient modular multiplication algorithm executed in hardware. The radix-2 Montgomery modular multiplication requires less area than its high cardinal version, but its computational performance is worse. Higher computationally efficient high-base multiplication modules introduce area costs.

Algorithm 1: Montgomery Power Ladder Method With Montgomery Multiplication

Input: X, N, $R(=2^k)$
$E = (e_{m-1}, \dots, e_1, e_0)_2$
Output: $Z = X^E \bmod N$
1: $W = -N^{-1} \bmod N$
2: $A = MontMult(X, R^2, N, W)$
3: $B = MontMult(1, R^2, N, W)$
4: **for** $i = k - 1$ **downto** 0 **do**
5: **if** $Ei = 1$ **then**
6: $B = MontMult(A, B, N, W)$
7: $A = MontMult(A, A, N, W)$
8: **else**
9: $A = MontMult(A, B, N, W)$
10: $B = MontMult(B, B, N, W)$
11: **end if**
12: **end for**
13: $Z = MontMult(B, 1, N, W)$
14: **return** Z

Consulting [3], we studied the cardinality of the most balanced area and computational efficiency, and chose the most appropriate radix value. Finally, we prepared a 256 radix Montgomery multiplier that pursues best balance. The algorithm is shown in Algorithm 2.

Algorithm 2: Radix-r Montgomery Multiplication. Calculate $Z = XXY2^{-r \cdot m} \bmod N$

Input: $X = (x_{m-1}, \dots, x_0, x_0)^r{}_2$
$Y = (y_{m-1}, \dots, y_0, y_0)^r{}_2$
$N = (n_{m-1}, \dots, n_0, n_0)^r{}_2$
$w = -N^{-1} \bmod N$
Output: $Z = XXY2^{-r \cdot m} \bmod N$
1: $Z = 0$, $v = 0$
2: **for** $i = 0$ **to** $m - 1$ **do**
3: $(c_a, z_0) = z_0 + x_{ii} y_0$
4: $t_{ii} = z_0 w \bmod 2^r$
5: $(c_b, z_0) = z_0 + t_{ii} n_0$
6: **for** $j = 0$ **to** $m - 1$ **do**
7: $(c_a, z_{jj}) = z_{jj} + x_{ii} y_{jj} + c_1$
8: $(c_b, z_{jj-1}) = z_{jj} + t_{ii} n_{jj} + c_b$
9: **end for**
10: $(v, z_{m-1}) = c_a + c_b + v$
11: **end for**
12: **if** $Z > N$ **then**
13: $Z = Z - N$
14: **end if**
15: **return** Z

According to the Algorithm 2, our arithmetic core of Montgomery multiplier consists of two adders, a carry save adder, and a radix-r partial product multiplier. The core computes as follows:

$$z = a + b \cdot c + d \tag{4}$$

The number of cycles required for one Montgomery multiplication by this Montgomery multiplier is as follows using the bit size *size* of the input value given from the control block and the radix r.

$$2(size/r)^2 + 3(size/r) + 1 \tag{5}$$

3. Circuit Design and Synthesis

In this design, we focus on designing Paillier crypto-processer to accelerate Paillier cryptosystem's encryption and decryption to further improve its efficiency in a variety of practical applications.

In this case, the maximum modulus M (which equals to N^2) and the maximum bit width of the cipher domain are 2048 bits, that is, the bit width of the input and output of each core multiplication operation is 2048 bits. We use a total of eight 2048-bit Blocks, and one 256-bit Block to form our Registers.

Paillier homomorphic cryptosystem mainly consists of encryption, homomorphic operation and decryption. We describe the computation flow of encryption and decryption of it, which includes modular multiplication (ModMult) and exponentiation (ModExp) operations.

Encryption function requires two ModExp operations and decryption function requires only one. According to the Montgomery ladder method, each step in ModExp requires two times of modular multiplication in Algorithm 2. We totally use two ModMult components each of which consists of one MontMult Block, and using each of them to perform each of the exponentiation operation in encryption. One of the ModMult components is reused for the last modular multiplication operation required in the Encryption.

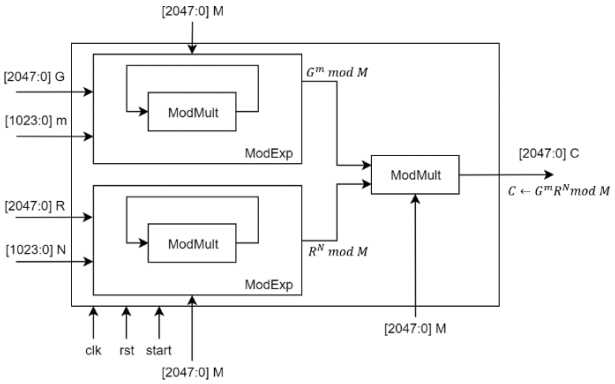

Figure 1. Encryption architecture

Figure 2 illustrates the computation flow of decryp-tion, which includes modular multiplication, exponent-tiation, and $L(u)$ operation. At the time of decryption, both exist ModMult components are reused to perform the same only one exponentiation operation in parallel to achieve the reduce of calculation time because they operate using different input resources. $L(u)$ is a division module to perform $(u-1)/n$. We use restoring method to implement our division component. And the precomputed value μ is multiplied with the result of $L(u)$ with the reused ModMult component to obtain the final outcome of the decryption.

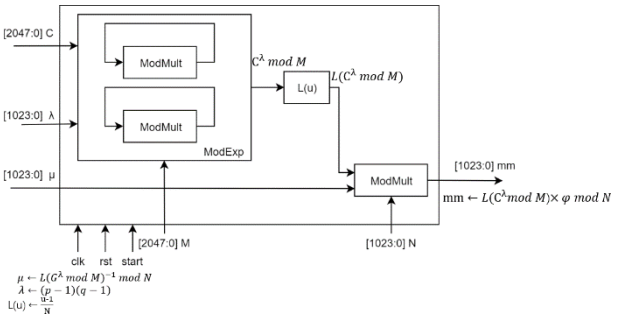

Figure 2. Decryption architecture

4. Result

The experimental results on chip in this design are shown in Table 1. At this time, the gate count is 1820kGates. Although the maximum frequency of this design is only about 14% compared with the previous study, the clock cycles of the encryption process are reduced to 7% of the previous research. Due to the parallelization in the decryption process, the clock cycles of the decryption process are reduced by nearly half compared to the encryption process, which is about 4% of the previous research. The time cost by the encryption process is 5.90 ns, which is about 52% of the previous research. The total decryption process costs 2.74 ns, which is about 20% of the previous research. Compared with the previous research, the computational efficiency has been greatly improved. The homomorphic operation in the cipher domain and the decryption of the calculation result cost a total of 2.86 ns. With such efficiency, the application such as [4] can achieve nearly 500 times data authentication comparisons in one second.

Decryption operation is considered to be the core com-putation in applications such as secure authenticate. The results presented in Figure 3 show the trend of frequency, operation time, power and energy per Paillier decryption operation with VDD. The figure shows that with the growth of VDD, the energy shows an upward trend. When VDD = 1.3V, we get the fastest one with the energy cost of 1.7mJoules. Out chip can also work at such low voltage as VDD = 260mV, with the energy of only 0.13mJoules.

Table 1. Comparison with previous implementation

	Design	Platform	Resources	Encryption				
				# of Cycles	Frequency [MHz]	Time [ms/OP]	Power [mW]	Energy [mJ/OP]
①	San[5]	Xilinx 7vx330t-3 28nm	3690 slices / 45DSP48E1 blocks	4,357,692	386	11.29	993*1	11.21
②	This work	65nm FDSOI	1820kGates	327,995	0.16	2000.77	0.11	0.29
					27.78	11.81	111.90	1.32
					52.63	6.23	568.44	3.54
					55.56	5.90	715.00	4.22

	Decryption					Add Operation + Decryption					VDD [V]
	# of Cycles	Frequency [MHz]	Time [ms/OP]	Power [mW]	Energy [mJ/OP]	# of Cycles	Frequency [MHz]	Time [ms/OP]	Power [mW]	Energy [mJ/OP]	
①	4,414,420	323	13.67	993*1	13.57	~4,418,659*2	323	13.68	993*1	13.58	1.00
②	174,704	0.16	1096.99	0.12	0.13	178,955	0.15	1163.21	0.11	0.13	0.26
		30.30	5.66	116.70	0.66		30.30	5.91	116.63	0.66	0.75
		58.82	2.91	584.40	1.70		55.56	3.22	559.92	1.73	1.20
		62.50	2.74	741.78	2.03		62.50	2.86	745.16	2.04	1.30

*1 Estimated by Xilinx Power Estimator (XPE) – 2019.1
*2 Estimated by Total # of cycles and # of ModMult with n^2 in San[5]

5. Conclusion

Paillier with homomorphism has high application value in the field of secret calculation. Since the security level and computational efficiency of encryption are the two conflicting goals, there are often some performance losses in authentication systems that consider security issues.

We propose a high-speed Paillier crypto processor that can be used to overcome the cumbersome calculations of each authentication. For the first time, we attempted to design an ASIC implementation for a Paillier-based homomorphic cryptosystem. In this study we designed a high-performance cryptographic processor with a 1024-bit Paillier homomorphic algorithm. Our hardware architecture primarily uses 32-bit IO and 256-bit arithmetic core to implement the operations required for Paillier cryptosystem.

We conducted experiments and simulations to evaluate the proposed design in terms of latency, area, and time to complete encryption and decryption. Our empirical results show that the proposed architecture is much faster than the FPGA implementation of the Paillier homomorphic cryptosystem in the previous study. Our research results show that this study proposes a good solution to solve the performance loss problem caused by the application of a secure authentication system at a higher security level and database data volume. In addition to this application, our design can also be considered for other applications, such as e-voting and third-party statistics.

Acknowledgments

The tools used in this research were conducted through the cooperation of Synopsys, Inc. and Mentor, Inc. through the VLSI Design and Education Center, The University of Tokyo. Also, the SOTB 65nm CMOS technology information used in this research is provided by Renesas Electronics Corporation through the VLSI Design and Education Center, the University of Tokyo.

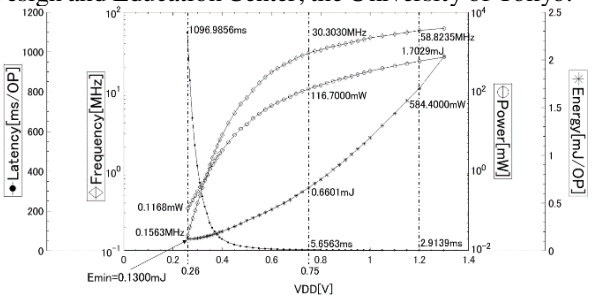

Figure 3. Paillier Decryption Results on chip

References

[1] P. Paillier, "Public-Key Cryptosystems Based on Composite Degree Residuosity Classes," Advances in Cryptology – EUROCRYPT '99, Vol. 1592, pp. 223-238 (1999).

[2] P. L. Montgomery, "Modular Multiplication Without Trial Division," Mathematics of Computation, Vol. 44, No. 100, pp. 519-521 (1985).

[3] A. Miyamoto, N. Homma, T. Aoki and A. Satoh, "Systematic Design of RSA Processors Based on High-Radix Montgomery Multipliers," IEEE Trans. On VLSI Systems, Vol. 19, No. 7, pp.1136-1146 (2011).

[4] Y.Ma, L.Wu, X.Gu, J. He, Z.Yang, "A Secure Face-Verification Scheme Based on Homomorphic Encryption and Deep Neural Networks," in Proc.IEEE Access, 5, pp.16532-16538(2017).

[5] I.San, N.At, I.Yakut, H.Polat, "Efficient paillier cryptoprocessor for privacy-preserving data mining," in Security and Communication Networks Vol. 9, Issue 11, pp. 1535-1546 (2016).