On the Security of Secret Sharing over a Ring and the Fast Implementation

Hongru Cao, Sian-Jheng Lin, *Member, IEEE,* Weiming Zhang, *Member, IEEE,* and Yunghsiang S. Han, *Fellow, IEEE*

***Abstract*—Secret Sharing is the method to share secrets among a group of shares, and the secret can be reconstructed if one obtains a predefined number of shares. The polynomial secret sharing is usually constructed over a field. In this paper, a novel polynomial secret sharing over a ring is proposed. In particular, by choosing a certain ring, the Fast Fourier Transform (FFT) can be applied on the encoding of secret sharing. The analysis shows that the proposed secret sharing scheme requires $O(N \log_2 N)$ Boolean operations per secret bit, which improves the prior result $O(8^{\log^* N} N \log_2 N)$ Boolean operations per secret bit. The simulation shows that the proposed scheme is in average 4 times faster than the conventional approach.**

***Index Terms*—Secret Sharing, Finite Ring, Fermat Number, Fast Fourier Transform.**

I. Introduction

SECRET sharing is the scheme to share secrets among a group of participants. A dealer divides the secret s into m shares (or called shadows), and those shares are assigned to the participants. For a (k, m)-threshold secret sharing, the secret can be reconstructed by any k out of m shares, and any $k-1$ out of m shares cannot reveal any information of the secret.

In 1979, Shamir [1] and Blakley [2] introduced two families of threshold secret sharing schemes independently. The former is based on Lagrange interpolations, and the latter is based on the theorem of projective geometry. In Shamir's scheme, the dealer randomly picks a polynomial $f(x)$ of degree less than k, and the constant term of $f(x)$ is the secret. Then the set of shares is given by $\{(x_i, f(x_i))\}_{i=0}^{m-1}$, and $x_i \neq x_j$, for any $i \neq j$. In particular, $f(x)$ can be uniquely determined by any k shares, and thus the secret can be recovered. If the number of shares is less than k, $f(x)$ cannot be uniquely determined, and thus the secret cannot be recovered. Notably, the scheme is theoretically equivalent to a class of linear block codes, Reed-Solomon codes. Moreover, in Blakley's scheme, the secret is a point in k-dimension space, while the shares are $(k-1)$-dimension hyperplane. These hyperplanes go through the secret point, but any three of them should not go through the same $(k-2)$-dimension hyperplane. Any k hyperplanes can uniquely determine the secret point, but any $k-1$ hyperplanes can only determine a line through the secret point.

In 1992, Desmedt and Frankel developed a method [3] to combine secret sharing and RSA signature. In their paper, they extended Shamir's secret sharing algorithm onto ring $\mathbb{Z}_{pq}$ and especially onto its algebra where p, q are safe primes. Then Armand [4] proposed a generalized constraint for Lagrange interpolation over commutative rings in 2004. He provided that the polynomial $f(x)$ can be interpolated from the data set $\{(x_i, f(x_i))\}_{i=1}^{m}$ over a commutative ring R if $x_i - x_j$ is not a zero-divisor for any pair of $x_i, x_j, i \neq j$.

Nowadays, secret sharing have been used in some applications, and some of them are listed as follows.

- *Threshold Digital Signature* is an important topic of threshold cryptography [3]. The purpose of it is to distribute the signature entitlement into shadows. Participants can sign a shadow-signature by using his shadow. If one has enough shadow-signatures (no less than the threshold), one can obtain the available signature. Threshold digital signature is useful in sensitive information protection and group authentication in secrecy.
- *Distributed Key Generation* was introduced by Pedersen in 1991 [5]. In this system, participants try to generate key of a public key system. Although many improvements are raised, the computational efficiency is still an unsolved problem.
- *Secure Multi-Party Computation (SMPC)* [6], [7] aims to solve a function f with n inputs from n participants, and those participants do not trust each other. Thus, the input of f is confidential to others, and every participants should obtain the correct answer. After Gennaro et al. [8] raised an SMPC protocol based on secret sharing in 1998, this approach has been improved by many scholars. In 2015, Zyskind et al. [9] conceived a decentralized computation platform by using SMPC and secret sharing schemes based on blockchain technologies [10], [11].

To our knowledge, most secret sharing schemes are constructed over finite fields, and only a few of them [3] are constructed over finite rings. In this paper, we consider the secret sharing schemes over finite rings. The contributions are listed below.

- The condition for the secret sharing scheme over a commutative ring is presented.
- The secret sharing scheme over a ring is proposed. The fast implementation of the proposed secret sharing is given. The bit complexity of the proposed ap-

This work is partially supported by the National Natural Science Foundation of China (No. 61671007) and ISTI, Dongguan University of Technology (No. KCYXM2017025).

Cao, Lin and Zhang were with the School of Information Science and Technology, University of Science and Technology of China, CAS Key Laboratory of Electro-magnetic Space Information, Heifei, 230026, China. (E-mail: chrkeith@mail.ustc.edu.cn, sjlin@u-stc.edu.cn, zhangwm@ustc.edu.cn).

Han was with the School of Electrical Engineering & Intelligentization, Dongguan University of Technology, Dongguan, China.(E-mail: yunghsiangh@gmail.com)

TABLE I
NOTATIONS

$\mathbb{Z}$	the integer ring
$\mathbb{Z}_n$	the ring of integers modulo n, $\mathbb{Z}/n\mathbb{Z}$
$\mathbb{F}_n$	the finite field of size n
$R[x]$	the polynomial ring over R
N	$N = 2^n$ is the n-th power of two
$\mathbf{F}_n$	the n-th Fermat number, $2^N + 1$
$\log^* x$	$\log^* x = 0$ if $x \leq 1$, or else $\log^* x = 1 + \log^*(\log x)$
k	the threshold to reconstruct the secret
s	the secret message

proach is $O(N \log N)$, which improves the prior result $O(8^{\log^* N} N \log N)$ by using FFTs on Shamir's schemes, where the size of the ring for the proposed method and that of the field for Shamir's scheme are both N.

In the rest of this paper, Section II introduces secret sharing scheme over commutative rings and shows the security conditions. Section III presents the proposed secret sharing scheme, and Section IV shows the simulation results. Finally, Section V concludes this work.

II. Secret sharing scheme over a commutative ring

Table I tabulates the notations used in this paper. This section introduces Shamir's conventional (k, m)-threshold secret sharing scheme. Then the (k, m)-threshold secret sharing over a commutative ring A with unity is presented.

A. Conventional secret sharing method

The secret is denoted as $s \in \mathbb{F}_N$. In the encoding phase, the dealer first randomly chooses $k-1$ coefficients $a_1, \ldots, a_{k-1}$ to form a polynomial $f(x) = \sum_{i=0}^{k-1} a_i \in \mathbb{F}_N[x]$, and $f(0) = a_0 = s$. Then the dealer calculates and secretly distributes the set of m shares $\{(p_i, f(p_i)) : 1 \leq i \leq m, p_i \neq 0 \in \mathbb{F}_N\}$, where $m \leq N-1$ and $p_i \neq p_j$ for $i \neq j$.

In the decoding phase, assuming that one receives k shares $\{(x_i, y_i = f(x_i))\}_{i=1}^{k}$, the polynomial can be reconstructed via Lagrange interpolation

$$f(x) = \sum_{i=1}^{k} y_i \prod_{j \neq i} \frac{x - x_j}{x_i - x_j}. \tag{1}$$

Then the secret is given by

$$s = f(0) = \sum_{i=1}^{k} y_i \prod_{j \neq i} \frac{-x_j}{x_i - x_j}. \tag{2}$$

This scheme encodes a n-bit secret to m shares via polynomial evaluations over a field $\mathbb{F}_N$, where $N - 1 \geq m$ and $N = 2^n$. To date, the N-point FFT over a field $\mathbb{F}_N$ requires $O(8^{\log^* N} N \log^2 N)$ Boolean operations [12]. This gives the average complexity $O(8^{\log^* N} n \log N)$ per secret bit.

B. Secret sharing over a ring

The scheme is similar to Section II-A, but the parameters are chosen on a ring and two conditions are appended. The secret is denoted as $s \in A$. In the encoding phase, the dealer first randomly chooses a polynomial $f(x) \in A[x]$ of degree less than k, and $f(0) = s$. Then the dealer calculates and secretly distributes the set of m shares $\{(p_i, f(p_i)) : 1 \leq i \leq m, p_i \text{ is chosen from } S\}$, where $m \leq |S|$. The set S is a subset of A, and satisfy the following two conditions:

- **Feasibility condition [4]:** This ensures that $f(x)$ can be uniquely determined by any k shares

$$\forall p, q \in S, p - q \text{ is a unit in } A. \tag{3}$$

- **Security condition:** This ensures that the secret s cannot be recovered by any $k-1$ shares:

$$\forall p \in S, p \text{ is a unit in } A. \tag{4}$$

The validity is proved later. The decoding phase is identical to the Shamir's approach.

C. Security condition

This subsection discusses the security of the secret sharing over a ring. To begin with, the proof relies on the following lemma.

Lemma 1. *Given any $k-1$ shares $\{(x_i, y_i = f(x_i))\}_{i=1}^{k-1}$ and any $t \in A$, there exists exact one polynomial $f_t(x) \in A[x]$ of degree less than k, such that $f_t(0) = t$ and $f_t(x_i) = f(x_i)$.*

Proof. With the $k-1$ shares, the interpolated polynomial is given by

$$g(x) = \sum_{i=1}^{k-1} y_i \prod_{j \neq i} \frac{x - x_j}{x_i - x_j}.$$

Then the polynomial $h_c(x)$, $c \in A$, is defined as

$$h_c(x) = c \prod_{i=1}^{k-1} (x_i - x) + g(x). \tag{5}$$

It can be seen that $h_c(x_i) = g(x_i) = f(x_i)$, for $i = 1, 2, \cdots, k-1$.

Then we find out the value of c such that $t = h_c(0)$. That is,

$$t = c \prod_{i=1}^{k-1} x_i + g(0) \tag{6}$$

and

$$c = (t - g(0)) \prod_{i=1}^{k-1} x_i^{-1}. \tag{7}$$

Clearly, (7) is valid when each x_i is a unit in A, and this refers the security condition (4). By plugging (7) into $h_c(x)$, we have

$$f_t(x) = (t - g(0)) \prod_{i=1}^{k-1} (1 - x_i^{-1} x) + g(x). \tag{8}$$

Note that $x_i \neq 0$ for $1 \leq i \leq k-1$. Since $(0, t)$ and $(x_i, y_i), 1 \leq i \leq k-1$, are k distinct solutions of $f_t(x)$, they uniquely determine $f_t(x)$. □

By Lemma 1, if one only collects $k-1$ shares, there uniquely exists one polynomial that satisfies all $k-1$ shares and have any secret $f_t(0)$. Hence, there is no information revealed if one only collects $k-1$ shares.

III. Proposed secret sharing scheme

In this section, the secret sharing scheme over a ring is introduced. Then the validity of the proposed scheme is discussed. Finally, a fast algorithm based on FFT over a ring is illustrated.

A. Proposed scheme

The proposed scheme follows the framework addressed in Section II-A and II-B. Hence, the following gives the definition of the ring and the set of evaluation points. Let $\mathbf{F}_n = 2^N + 1$, denote the n-th Fermat number, for $N = 2^n$, $n \in \mathbb{Z}$. The proposed secret sharing scheme is over $\mathbb{Z}_{\mathbf{F}_n}$, that is the ring of integers modulo $\mathbf{F}_n$. Furthermore, the set $S_n \subseteq \mathbb{Z}_{\mathbf{F}_n}$ is defined as

$$S_n = \{\pm 2^i\}_{i=1}^{N} = \{2^i\}_{i=1}^{2N} \subseteq \mathbb{Z}_{\mathbf{F}_n}. \tag{9}$$

The above equation is due to the fact that $2^{N+j} = 2^N 2^j = -2^j$ for $1 \le j \le N$. Thus, the (k, m)-threshold secret sharing scheme over $\mathbb{Z}_{\mathbf{F}_n}$ can generate $m \le |S_n| = 2N$ shares.

B. Validity of the scheme

In this subsection, we show that S_n in (9) meets the two conditions (3) and (4). First, it is obvious that each element $\pm 2^i$ is co-prime with $\mathbf{F}_n$. Thus, each element in S_n is a unit in $\mathbb{Z}_{\mathbf{F}_n}$. This satisfies the security condition (4).

Second, we show that S_n meets the feasibility condition.

Lemma 2.

$$\gcd(A, B) \in \{1, 2^{\gcd(a,b)} \pm 1\},$$

where $A = 2^a \pm 1$, $B = 2^b \pm 1$, *and* $a, b \in \mathbb{N}$.

Proof. WLOG, assume $A \ge B$. First, if $A = B$, $\gcd(A, B) = A = 2^{\gcd(a,b)} \pm 1$. Second, if $A > B$ and $a = b$, $\gcd(A, B) = \gcd(2^a + 1, 2^a - 1) = \gcd(2, 2^a - 1) = 1$. Finally, if $A > B$ and $a > b$, then

$$\begin{aligned} &\gcd(2^a \pm 1, 2^b + 1) \\ &= \gcd(2^a \pm 1 - 2^{a-b}(2^b + 1), 2^b + 1) \\ &= \gcd(-2^{a-b} \pm 1, 2^b + 1) \end{aligned} \tag{10}$$

and

$$\begin{aligned} &\gcd(2^a \pm 1, 2^b - 1) \\ &= \gcd(2^a \pm 1 - 2^{a-b}(2^b - 1), 2^b - 1) \\ &= \gcd(2^{a-b} \pm 1, 2^b - 1). \end{aligned} \tag{11}$$

As $\gcd(-x, y) = \gcd(x, y)$, the above results can be combined to obtain

$$\gcd(2^a \pm 1, 2^b \pm 1) = \gcd(2^{a-b} \pm 1, 2^b \pm 1). \tag{12}$$

(12) can be rewritten as

$$H(a, b) = H(a - b, b), \tag{13}$$

where

$$H(a, b) = \gcd(2^a \pm 1, 2^b \pm 1).$$

By applying (13) recursively, we can obtain

$$H(a, b) = H(\gcd(a, b), \gcd(a, b)).$$

By the argument in the beginning of the proof, we have $H(\gcd(a, b), \gcd(a, b)) = 1$ or $2^{\gcd(a,b)} \pm 1$. This completes the proof. □

Proposition 3. *For each pair of elements* $A, B \in S_n$, *and* $A \ne B$, $A - B$ *is a unit in* $\mathbb{Z}_{\mathbf{F}_n}$.

Proof. For any $A = \pm 2^a, B = \pm 2^b \in S_n$, the statement is equivalent to prove

$$\gcd(A - B, \mathbf{F}_n) = 1. \tag{14}$$

WLOG, we consider $0 \le b < a < 2^n$. From Lemma 2, we have

$$\begin{aligned} \gcd(2^a - 2^b, \mathbf{F}_n) &= \gcd(2^b(2^{a-b} - 1), \mathbf{F}_n) \\ &= \gcd(2^{a-b} - 1, \mathbf{F}_n) \\ &\in \{1, 2^{\gcd(a-b,N)} \pm 1\}. \end{aligned} \tag{15}$$

Assume

$$\gcd(2^a - 2^b, \mathbf{F}_n) \ne 1. \tag{16}$$

As N is a power of two, we have $\gcd(a - b, N) = 2^k$ and $0 \le k \le n$. From the assumption (16), (15) can be written as

$$\gcd(2^a - 2^b, \mathbf{F}_n) = 2^{2^k} \pm 1 \tag{17}$$

The validity of (17) is discussed based on the value of k.

1) $k = n$: In this case, $\gcd(a - b, N) = 2^k = N$, thus $N | a - b$. However, since $0 \le b < a < 2^n$, we have $a - b < N$, which contradicts $N | a - b$.
2) $k < n$: Clearly, $\gcd(\mathbf{F}_i, \mathbf{F}_j) = 1$ for any $i \ne j$. Then we have

$$\gcd(2^{2^k} + 1, \mathbf{F}_n) = \gcd(\mathbf{F}_k, \mathbf{F}_n) = 1. \tag{18}$$

Furthermore, as $2^{2^k} - 1 = \prod_{i=0}^{k-1} \mathbf{F}_i$, we have

$$\gcd(2^{2^k} - 1, \mathbf{F}_n) = \gcd(\prod_{i=0}^{k-1} \mathbf{F}_i, \mathbf{F}_n) = 1. \tag{19}$$

From (18) and (19), we conclude that

$$\gcd(2^{2^k} \pm 1, \mathbf{F}_n) = 1. \tag{20}$$

From (17) and (20), we have

$$\gcd(2^a - 2^b, \mathbf{F}_n) = 1, \tag{21}$$

which contradicts the assumption (16).

Thus, we conclude that (14) is valid. This completes the proof. □

TABLE II
COMPARISON OF THE ENCODING COMPLEXITIES OF SECRET SHARING SCHEMES

	Bit complexity per secret bit
Shamir's scheme [1], [14]–[16]	$O(8^{\log^* N} N \log N)$
Kurihara's scheme [17], [18]	$O(N^2)$
Proposed scheme	$O(N \log N)$

C. Fast implementation for encoding process

Given a polynomial $f(x) \in \mathbb{Z}_{\mathbf{F}_n}[x]/(x^{2N}-1)$ and $N = 2^n$, the encoding process is to evaluate $f(x)$ at S_n defined in (9). Precisely, given $f(x) = \sum_{j=0}^{2N-1} a_j x^j$, the encoding process is to calculate

$$f(\omega^i) = \sum_{j=0}^{2N-1} \omega^{ji} a_j, \qquad \forall i = 0, 1, \ldots, 2N-1, \tag{22}$$

where $\omega = 2$ is the $2N$-th root of unity. From (22), the encoding process can be implemented as a $2N$-point Fourier transform over $\mathbb{Z}_{\mathbf{F}_n}$. In order to reduce the encoding complexity, we apply the radix-2 FFT over $\mathbb{Z}_{\mathbf{F}_n}$, which was also used in big integer multiplications [13].

Next we consider the Boolean complexity of the FFT over $\mathbb{Z}_{\mathbf{F}_n}$. It is known that a $2N$-element radix-2 FFT requires $2N \log_2(2N) = O(N \log_2 N)$ additions and $N \log_2(2N) = O(N \log_2 N)$ multiplications. For the addition, it is clear that an addition requires $O(N)$ Boolean operations in $\mathbb{Z}_{\mathbf{F}_n}$. For the multiplication, it is noted that the twiddle factors in the FFT are always a power of $\omega = 2$. Thus, the multiplications in FFT can be written as

$$a \cdot 2^i \pmod{2^N + 1}, \tag{23}$$

where $i \in \{0, 1, \ldots, 2N-1\}$ and $a \in \mathbb{Z}_{\mathbf{F}_n}$ is the data element. Clearly, the multiplication given in (23) can be implemented by a shift and a subtraction, and this requires $O(N)$ Boolean operations. In summary, the Boolean complexity of the FFT over $\mathbb{Z}_{\mathbf{F}_n}$ is $O(N^2 \log_2 N)$. Thus, the proposed algorithm encodes an N-bit secret to $2N$ shares in $O(N^2 \log_2 N)$ Boolean operations, and the average complexity is $O(N \log_2 N)$ per secret bit.

The comparison of the bit complexities of secret sharing schemes are demonstrated in Table II. As shown in Table II, the proposed scheme reduces the encoding complexity at least by a factor $O(8^{\log^* N})$ compared with other schemes.

D. Fast implementation for decoding process

The decoding process is to calculate the secret via (2), that can be rewritten as

$$s = \sum_{i=1}^{k} y_i \prod_{j \neq i} \frac{-x_j}{x_i - x_j} = \sum_{i=1}^{k} y_i G_i, \tag{24}$$

where $G_i = \prod_{j \neq i} (1 - x_i/x_j)^{-1}$. From (9), x_i/x_j in G_i is a power of two, and thus each factor $(1 - x_i/x_j)^{-1}$ in G_i is an element of

$$R = \{(1 - 2^i)^{-1}\}_{i=1}^{2N-1}. \tag{25}$$

In implementations, we can create a table to store all elements of R, and the size of the table is $O(N^2)$ bits.

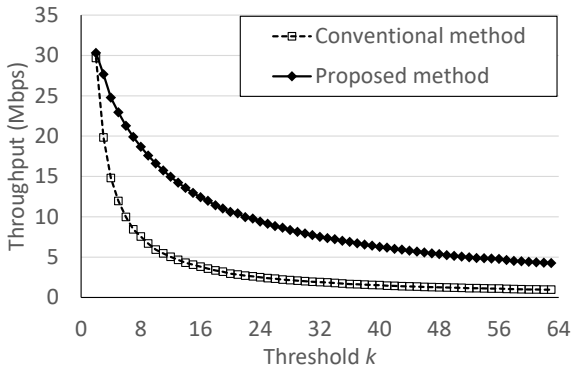

Fig. 1. Throughputs of the encoding process

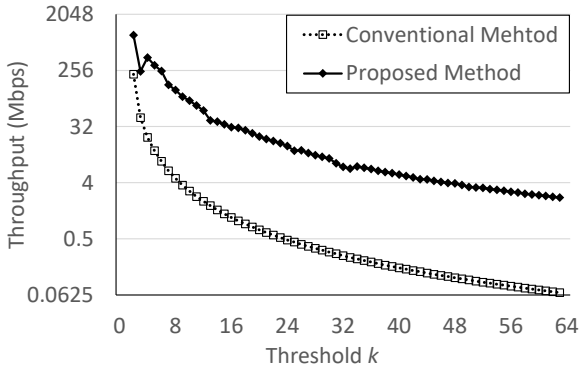

Fig. 2. Throughputs of the decoding process

IV. SIMULATION

This section presents a simulation for the conventional secret sharing scheme and the proposed approach presented in Sec. III-C and Sec. III-D. The programs are written in C, and the compiler is gcc 7.3.0 on the 64-bit version. All programs are running on Windows 7, Intel Core i5-3230M 2.60GHz. In both methods, the ring is chosen as $\mathbb{Z}_{\mathbf{F}_5}$, and the secrets are randomly chosen in $\mathbb{Z}_{\mathbf{F}_5}$.

Figures 1 and 2 shows the throughputs of the two methods, for $n = 64$ and $k = 2, 3, \cdots, 63$. As shown in the figure, the improvement is significant when k is large. In average, comparing with the conventional approach, the proposed method is 4.02 times faster in encoding, and 28.07 times faster in decoding.

V. CONCLUSION

This paper gives the security condition for the Shamir's secret sharing over a ring. Based on the condition, a secret sharing scheme, as well as the fast implementation, are proposed. The proposed encoding algorithm requires $O(N \log N)$ Boolean operations per secret bit, and this improves the prior result $O(8^{\log^* N} N \log^2 N)$ by the secret sharing over a field. The simulations show that the proposed encoding algorithm is around 4 times faster than the conventional approach, and the proposed decoding algorithm is around 28 times faster than the conventional approach. In the future, it is interesting to explore the secret sharing schemes over other commutative rings.

REFERENCES

[1] A. Shamir, "How to share a secret," *Communications of the ACM*, vol. 22, no. 11, pp. 612–613, 1979.

[2] G. Blakley, "Safeguarding cryptographic keys," in *Proceedings of the 1979 AFIPS National Computer Conference*. Monval, NJ, USA: AFIPS Press, 1979, pp. 313–317.

[3] Y. Desmedt and Y. Frankel, "Shared generation of authenticators and signatures," in *Advances in Cryptology — CRYPTO '91*, Santa Barbara, CA, USA, 1992, pp. 457–469.

[4] M. A. Armand, "Generalized rational interpolation over commutative rings and remainder decoding," *IEEE Transactions on Information Theory*, vol. 50, no. 4, pp. 683–690, 2004.

[5] T. P. Pedersen, "A threshold cryptosystem without a trusted party," in *Advances in Cryptology — EUROCRYPT '91*, Brighton, UK, 1991, pp. 522–526.

[6] A. C. Yao, "Theory and application of trapdoor functions," in *23rd Annual Symposium on Foundations of Computer Science (sfcs 1982)*, Chicago, IL, USA, Nov 1982, pp. 80–91.

[7] O. Goldreich, S. Micali, and A. Wigderson, "How to play any mental game," in *Proceedings of the Nineteenth Annual ACM Symposium on Theory of Computing*, ser. STOC '87, New York, NY, USA, 1987, pp. 218–229.

[8] R. Gennaro, M. O. Rabin, and T. Rabin, "Simplified VSS and fast-track multiparty computations with applications to threshold cryptography," in *Proceedings of the Seventeenth Annual ACM Symposium on Principles of Distributed Computing*, ser. PODC '98, Puerto Vallarta, Mexico, 1998, pp. 101–111.

[9] G. Zyskind, O. Nathan, and A. Pentland, "Enigma: decentralized computation platform with guaranteed privacy. 2015." [Online]. Available: URL: https://enigma. co/enigma_full. pdf

[10] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," 2008. [Online]. Available: https://bitcoin.org/bitcoin.pdf

[11] J. Bonneau, A. Miller, J. Clark, A. Narayanan, J. A. Kroll, and E. W. Felten, "SoK: Research perspectives and challenges for bitcoin and cryptocurrencies," in *2015 IEEE Symposium on Security and Privacy*, San Jose, CA, USA, May 2015, pp. 104–121.

[12] D. Harvey, J. V. D. Hoeven, and G. Lecerf, "Faster polynomial multiplication over finite fields," *J. ACM*, vol. 63, no. 6, pp. 52:1–52:23, Jan. 2017.

[13] A. Schönhage and V. Strassen, "Schnelle multiplikation großer zahlen," *Computing*, vol. 7, no. 3, pp. 281–292, Sep 1971.

[14] L.-J. Pang and Y.-M. Wang, "A new (t, n) multi-secret sharing scheme based on shamirs secret sharing," *Applied Mathematics and Computation*, vol. 167, no. 2, pp. 840–848, 2005.

[15] H.-Y. Chien, J.-K. Jan, and Y.-M. Tseng, "A practical (t, n) multi-secret sharing scheme," *IEICE transactions on fundamentals of electronics, communications and computer sciences*, vol. 83, no. 12, pp. 2762–2765, 2000.

[16] C.-C. Yang, T.-Y. Chang, and M.-S. Hwang, "A (t, n) multi-secret sharing scheme," *Applied Mathematics and Computation*, vol. 151, no. 2, pp. 483–490, 2004.

[17] J. Kurihara, S. Kiyomoto, K. Fukushima, and T. Tanaka, "On a fast (k, n)-threshold secret sharing scheme," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. 91, no. 9, pp. 2365–2378, 2008.

[18] ——, "A fast (k, l, n)-threshold ramp secret sharing scheme," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. 92, no. 8, pp. 1808–1821, 2009.