Hindawi
Security and Communication Networks
Volume 2022, Article ID 5429195, 13 pages
https://doi.org/10.1155/2022/5429195

Review Article

Cryptographic Accumulator and Its Application: A Survey

Yongjun Ren,[1,2] Xinyu Liu,[2] Qiang Wu,[2] Ling Wang,[2] and Weijian Zhang[3]

[1] *Information Security Evaluation Center of Civil Aviation, Civil Aviation University of China, Tianjin 300300, China*
[2] *School of Computer Science, Engineering Research Center of Digital Forensics, Ministry of Education, Nanjing University of Information Science & Technology, Nanjing 210044, China*
[3] *Network Security and Information Office, Hohai University, Nanjing, China*

Correspondence should be addressed to Weijian Zhang; wjzhang@hhu.edu.cn

Received 20 August 2021; Accepted 29 January 2022; Published 7 March 2022

Academic Editor: Stelvio Cimato

Copyright © 2022 Yongjun Ren et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Since the concept of cryptographic accumulators was first proposed in 1993, it has received continuous attention from researchers. The application of the cryptographic accumulator is also more extensive. This paper makes a systematic summary of the cryptographic accumulator. Firstly, descriptions and characteristics of cryptographic accumulators are given, and the one-way accumulator, collision-free accumulator, dynamic accumulator, and universal accumulator are introduced, respectively. Cryptographic accumulator can be divided into two types: symmetric accumulator and asymmetric accumulator. In the asymmetric accumulator, three different cryptographic accumulator schemes were classified based on three security assumptions. Finally, this paper summarized the applications of cryptographic accumulators in ring signature, group signature, encrypted data search, anonymous credentials, and cryptographic promise.

1. Introduction

The concept of cryptographic accumulators was first proposed in 1993 by Benaloh and de Mare [1], who developed a one-way accumulator encryption protocol that could be used for timestamp and membership testing through a hash function with quasi-commutativeness and one-way property. That is to say, for all $x \in X$ and $y_1, y_2 \in Y$, this one-way hash function $h\colon X \times Y \longrightarrow X$ satisfies the quasi-commutativeness:

$$h(h(x, y_1), y_2) = h(h(x, y_2), y_1). \tag{1}$$

The cryptographic accumulator scheme allows the accumulation of elements from a finite set $X = \{x_1, \ldots, x_n\}$ into a concise value acc_X of constant size, known as a cryptographic accumulator. Because the cryptographic accumulator satisfies the characteristic of quasi-commutativeness, the accumulated value acc_X does not depend on the order of the accumulated elements. Choose $g \in G$ as the base, and the original cryptographic accumulator is defined as

$$\mathrm{acc}_X = h(h(h(\ldots h(h(h(g, x_1), x_2), x_3), \ldots, x_{n-2}), x_{n-1}), x_n). \tag{2}$$

The witness wit_{x_i} of each element $x_i \in X$ in the set is calculated to verify $h(\mathrm{wit}_{x_i}, x_i) = \mathrm{acc}_X$, that is, to effectively prove the membership of element x_i.

At the same time, it is not feasible to find a membership witness for any unaccumulated element $y \notin X$ because of the collision resistance of one-way hash function.

The cryptographic accumulator has several important characteristics, such as being dynamic, robustness, universality, security assumption, and compactness, as shown in Table 1.

Although cryptographic accumulators have been roughly described in the review of cryptographic accumulators published by Ozcelik et al. [6], the summary of this paper is not comprehensive. Therefore, this paper makes a more comprehensive and detailed summary.

The roadmap of this paper is organized as follows: Section 2 introduces the descriptions of cryptographic accumulator. Section 3 classifies the cryptographic accumulators into symmetric accumulator and asymmetric accumulator. In Section 4, cryptographic accumulators based on various security assumptions are introduced in

Table 1: Characteristics of the cryptographic accumulator.

Characteristics	
Dynamic [2]	The cryptographic accumulator has efficient algorithms for adding, deleting, witnessing, and updating elements.
Robustness [1]	The administrator of the cryptographic accumulator does not need to be trusted, and trapdoor information cannot be used to forge witnesses.
Universality [3]	The cryptographic accumulator can provide not only membership proof but also nonmembership proof.
Security assumption [4]	Under the premise of security assumption declaration, the member verification function of the cryptographic accumulator is not affected by attackers.
Compactness [5]	The cryptographic accumulator can map a large set to accumulation value of a smaller order of magnitude, which is manifested by the small storage space required for accumulated value and witness, as well as the low time complexity of updating algorithm.

detail. Section 5 describes the cryptographic accumulator scheme of hidden order group and known order group. In Section 6, the applications of cryptographic accumulator are introduced. The seventh section gives a summary.

2. Descriptions of Cryptographic Accumulators

2.1. One-Way Accumulators. The concept of the cryptographic accumulator originated from the one-way accumulator first proposed by Benaloh and Mare [1]. A one-way accumulator is defined as a set of one-way hash functions with quasi-commutativeness.

One-Way Hash Functions [1, 7]. A family of one-way hash functions is an infinite sequence of families of functions $\{H_\lambda\}_{\lambda \in N}$, where $H_\lambda = \{h_k\colon X_k \times Y_k \longrightarrow Z_k\}$ (k is a security parameter), with the following properties:

(1) For any integer λ and any $h_k \in H_\lambda, h_k(.,.)$ is computable in time polynomial in λ.

(2) Any probabilistic, polynomial-time algorithm A satisfies

$$Pr\left[h_k \overset{R}{\longleftarrow} H_\lambda; x \overset{R}{\longleftarrow} X_k; y' \overset{R}{\longleftarrow} Y_k; x' \longleftarrow \cdot A\left(1^\lambda, x, y, y'\right)\colon h_k(x,y) = h_k(x', y')\right] < \text{negl}(\lambda), \tag{3}$$

where the probability depends on the random selection of h_k, x, y, y' and random output of A.

From the above description, it is seen that the one-way hash function is computable and one-way; that is, given x and y, the calculation of $z = h(x, y)$ can be completed in polynomial time, and if given x, y, and y', the probability of finding x' satisfying $h_k(x, y) = h_k(x', y')$ is too small to be ignored; that is, the conflict between the outputs generated by different inputs is very little.

Quasi-Commutativeness [7, 8]. A function f: $X \times Y \longrightarrow X$ has quasi-commutativeness means that the following equation holds:

$$\forall x \in X, \quad \forall y_1,\ y_2 \in Y\colon f(f(x, y_1), y_2) = f(f(x, y_2), y_1). \tag{4}$$

If a one-way hash function satisfies the quasi-commutativeness, first of all, the forward calculation is easy according to the one-way property, while the reverse calculation is difficult. Second, satisfying the quasi-commutativeness means that, under the condition of given initial value (Seeds), the results of multiple hash operations will not change with different calculation order.

A one-way hash function with quasi-commutativeness can be used to verify whether a value y_i is in a specified set $Y = \{y_i\}$. Specifically, the accumulative results z of Y can be calculated by the one-way accumulative function $h \in H$, using the following formula:

$$z = h(h(\ldots h(h(h(x_0, y_1), y_2), y_3), \ldots, y_{l-1}), y_l). \tag{5}$$

The accumulated value (called partial accumulated value) of $y_i = \{y | y \in Y, y \neq y_i\}$ other than y_i can also be calculated using a one-way accumulative function:

$z_i = h(h(\ \ldots\ (h(h(h(x_0, y_1), y_2)\ , y_3), \ldots, y_{i-1}), y_{i+1}), \ldots, y_{l-1}),\ y_l$. When verifying $y_i \in Y$ is required, the formula z': $z' = h(z_i, y_i)$ is used for calculation. If $z' = z$, $y_i \in Y$.

The above conclusion holds because if the attacker does not know y_i, according to the description of one-way function, it will face the computational difficulty that constructing y' makes $z = h(z_i, y')$ established. Hence, (y_i, z_i) can be regarded as a witness of $y_i \in Y$. During the following discussion, Z_N represents the set of all positive integers, and Z_n represents the set of positive integers with length within n.

One-Way Accumulators [1, 7]. (y_i, z_i) is the witness of $y_i \in Y$ meaning that it meets the following condition:

$$Pr\{k \in Z_N, z_i, y_i :: \exists y' \in Y, y' \neq y_i\colon h(z_i, y_i)\} < \frac{1}{A(k)}. \tag{6}$$

However, there is an obvious problem with the above analysis: Assume that the attacker can only randomly select predictive values y' in a given set Y. In fact, it is entirely possible for an attack to easily find y' satisfying $z = h(z_i, y')$ beyond the value domain set Y, thus destroying the above description of the witness. A strong description is obtained if the attacker's optional range of predictive values is extended beyond the specified set Y.

Strongly One-Way Hash Functions [7]. A family of strongly one-way hash functions is an infinite sequence of families of functions $\{H_\lambda\}_{\lambda \epsilon N}$, where $H_\lambda = \{h_k\colon X_k \times Y_k \longrightarrow Z_k\}$ (k is a security parameter), having the following properties:

(1) For any integer λ and any $h_k \epsilon H_\lambda, h_k(.,.)$ is computable in time polynomial in λ.

(2) Any probabilistic, polynomial-time algorithm A satisfies

$$Pr\left[h_k \overset{R}{\longleftarrow} H_\lambda; x \overset{R}{\longleftarrow} X_k; y \overset{R}{\longleftarrow} Y_k; (x', y') \longleftarrow \cdot A\left(1^\lambda, x, y\right): y' \neq y \Lambda h_k(x, y) = h_k(x', y') \right] < \text{negl}(\lambda). \tag{7}$$

The probability is taken over the random choice of h_k, x, y and random output of A.

One-way property means that, given values $(y_1, y_2, \ldots, y_n)$, their accumulative value z, and another value y', the attacker has difficulty finding the corresponding witness accu′ such that $h(\text{accu}', y') = z$. Strongly one-way property means that, given $(y_1, y_2, \ldots, y_n)$ and z, it is hard to find the value corresponding to (y', accu') so $h(\text{accu}', y') = z$, and $y' \notin (y_1, y_2, \ldots, y_n)$.

2.2. Collision-Free Accumulators. Strongly one-way property does not completely solve the problem of ensuring security in the case of an adversary actively participating in the selection of values to be accumulated (i.e., x and y in the above description are no longer randomly chosen but carefully chosen by the adversary). In order to fill this gap, Baric and Pfitzmann [5] proposed the concept of collision-free accumulators.

Baric and Pfitzmann [5] proposed that the cryptographic accumulator needs to be more strict when building FSS mechanisms. Under the strongly one-way property, the attacker may still carefully forge the member value $(y_1', y_2', \ldots, y_n')$ to construct witness accuı for y'. Therefore, a collision-free accumulator is introduced. On the strongly one-way property basis, the member value $(y_1', y_2', \ldots, y_n')$ does not need to be given.

Cryptographic Accumulator Scheme [5, 7]. The scheme of a cryptographic accumulator is a 4-tuple containing 4 polynomial time algorithms (Gen, Eval, Wit, and Ver):

(1) Gen (key generation algorithm): it is a probabilistic algorithm for generating initial parameters. Gen receives two parameters: a security variable 1^λ and an accumulator threshold N, an upper bound on the total number of values that can be securely accumulated, and finally returns an accumulator key k, $k \in k_{\lambda,N}$.

(2) Eval (evaluation algorithm): it is a probabilistic algorithm for finding accumulated values. Calculate all accumulated values in the set $L = \{y_1, y_2, \ldots, y_{N'}\}$, $N' \in N$, where $y_i \in Y_k$, $k \in K_{\lambda,N}$. Eval inputs $(k, y_1, y_2, \ldots, y_{N'})$ and outputs an accumulated value of $z \in Z_k$ and some auxiliary information of aux, which will be used as an input to other algorithms. Note that Eval outputs the same accumulated value for the same input, and the auxiliary information may be different.

(3) Wit (witness extraction algorithm): it is a probabilistic algorithm for generating member witnesses based on relevant information. Wit inputs an accumulator $k \in k_{\lambda,N}$, a value $y_i \in Y_k$, and auxiliary information aux outputted by Eval $(k, y_1, y_2, \ldots, y_{N'})$; if y_i is in L, a witness $w_i \in W_K$ is outputted to prove that y_i is accumulated within z; otherwise, it returns symbol $\perp$.

(4) Ver (verification algorithm): it is a deterministic algorithm for verifying the membership of a value by witness. Ver inputs (k, y_i, w_i, z) to verify that y_i is accumulated into z and outputs Yes or No according to witness w_i.

N-*Times Collision-Freeness* [5, 7]. A cryptographic accumulator scheme is said to be N-times collision-free when it satisfies the following property: A cryptographic accumulator scheme is said to be N-times collision-free if, for any integer λ and for any probabilistic, polynomial-time algorithm A,

$$Pr\left[k \longleftarrow \text{Gen}\left(1^\lambda, N\right); (y_1 \ldots, y_N, y', w') \longleftarrow A\left(1^\lambda, N, k\right); (z, \text{aux}) \longleftarrow \text{Eval}(k, y_1, \ldots, y_N): (y_1, \ldots, y_N \in Y_k) \cdot \Lambda(y' \notin \{y_1 \ldots, y_N\}) \Lambda (\text{Ver}(z, y', w') = \text{Yes})\right] < \text{negl}(\lambda). \tag{8}$$

where the probability is taken from random output of Gen, Eval, and A.

Collision-Freeness [5, 7]. A cryptographic accumulator scheme is collision-free if it is in all N-times collision-free.

2.3. Dynamic Accumulators. The application of member authentication requires that the selected cryptographic accumulator can not only enable the verifier to authenticate efficiently but also ensure the security. When a member set changes (added or deleted), the accumulated value and witness of each member can be updated efficiently; otherwise, whenever members are added or deleted, all members need to recalculate the current accumulated value and their respective witness. When the member set changes dynamically, the cryptographic accumulator cannot operate efficiently to meet the practical application requirements. For this reason, researchers put forward the concept of dynamic accumulator, which can add, delete, and update operations on the basis of the original 4-tuple.

Dynamic Accumulator Scheme [2, 7]. A dynamic accumulator scheme is a seven-tuple containing seven polynomial time algorithms (Gen, Eval, Wit, Ver, Add, Del, and Upd), where Gen, Eval, Wit, and Ver are the same as in the cryptographic accumulator scheme:

(1) Add (element addition algorithm): it is usually a deterministic algorithm. Given an accumulator key

k, an accumulated value z obtained as the accumulation of some set L of less than N elements, where $L \subseteq Y_K$, $z \in Z_k$, and the value $y' \in L$ to be deleted, it returns a new accumulator value z' corresponding to the set $L \setminus \{y'\}$, along with a witness $w' \in W_k$ for y' and some updated information aux_{Add} which will be used by the Upd algorithm.

(2) Del (element deletion algorithm): it is usually a deterministic algorithm. Given an accumulator key k, an accumulated value z obtained as the accumulation of some set L of less than N elements, where $L \subseteq Y_K$, $z \in Z_k$, and the value $y' \in Y_k$ to be added, it returns a new accumulator value z' corresponding to the set $L \cup \{y'\}$, along with some update information aux_{Del} which will be used by the Upd algorithm.

(3) Upd (witness update algorithm): it is a deterministic algorithm used to update the witness $w \in W_k$ of each existing element in the set $y \in Y_k$ after adding or deleting elements in L. Upd takes k, y, w, op, and aux_{op} as input (where op is either Add or Del) and returns an updated witness w' to prove that y has been accumulated into z'.

2.4. Universal Accumulators. Universal accumulators are dynamic and support (non)membership proofs [3]. Cryptographic accumulators that support membership proof are called positive accumulators, those that support nonmembership proof are called negative accumulators, and those that support both are called universal accumulators [9].

Assuming that k is a security parameter, the safe universal accumulator of the input $\{\chi_k\}$ family is a family of functions $\{F_k\}$ with the following properties [3]:

(i) Effective generation: there is an effective probabilistic polynomial time algorithm G, which generates a random function F_k on input 1^k. Moreover, G also outputs some auxiliary information about f, expressed as aux_f.

(ii) Efficient evaluation: each $f \in F_k$ is a polynomial time function, which outputs a value $h \in g_f$ when inputting $(g, x) \in g_f \times \chi_k$, where g_f is the input domain of the function f and χ_k is the input domain to accumulate the element.

(iii) Quasi-commutativity: for all $f \in F_k$, all $g \in g_f$, and all $x_1, x_2 \in \chi_k$, if $f(f(g, x_1), x_2) = f(f(g, x_2), x_1)$, $f(g, X)$ can represent $f(f(\ldots(g, x_1), \ldots))$, $x_m)$.

(iv) Membership witness: for each $f \in F_k$, there is a membership validation function ρ_1. Set $c \in g_f$ and $x \in \chi_k$. If $\rho_1(c, x, \omega_1) = 1$, the value ω_1 is called membership witness.

(v) Nonmembership witness: for each $f \in F_k$, there is a nonmembership validation function ρ_2. Set $c \in g_f$ and $x \in \chi_k$; the value ω_2 is called nonmembership witness if $\rho_2\ (c,\ x,\ w_2) = 1$.

(vi) Security: for all polynomial-time probability, attacker A_k satisfies

$$pr\left[f \longleftarrow G\left(1^k\right); g \longleftarrow g_f(x, w_1, w_2, X) \longleftarrow A_k\left(f, g_f, g\right) x \in \chi_k; X \subset \chi_k; \rho_1(f(g, X), x, w_1) = 1; \rho_2(f(g, X), x, w_2) = 1\right] = \text{neg}(k). \tag{9}$$

Then, the universal accumulator scheme is safe.

Table 2 provides description of different types of cryptographic accumulators.

3. Symmetric and Asymmetric Accumulators

3.1. Symmetrical Accumulators. The symmetric cryptographic accumulator is a trapdoor-free structure and does not require witness verification. In random oracle models, the existing structures are secure. The symmetric accumulator [14] basically consists of a one-way function $f\colon Y \longrightarrow X$ and a vector $x \in X$ of length l, initialized to the 0 vector. This set of values $\{y_1, y_2, \ldots, y_n\}$ accumulates as vector z: $z = x \vee f(y_1) \vee f(y_2) \vee \ldots \vee f(y_n)$, where $\vee$ is contained by bit. Given the accumulative vector z and values y_i, verify that membership in the accumulative vector includes calculating $v = f(y_i)$ and verifying that, $\forall k \in [[0, l-1]]$, $v_k = 1$ means $z_i = 1$. Symmetric accumulator does not need to calculate the witness. But it is stuck with the long output of cryptographic accumulators. Actually, the length of the cryptographic accumulator depends also on the number of values added to the cryptographic accumulator and not only on the security parameters.

Nyberg [15] proposed a symmetric accumulator. The idea is to use the hash function to generate hash values for the values to be accumulated. Each hash value h is considered to consist of r blocks of size d bits $h_1, h_2, \ldots, h_r$ composition. Then, by mapping each block to one bit, map such code to an r bit string. Accumulated value z is calculated as the coordinate directional bit product corresponding to the string to be accumulated. To verify the membership, the values y and the corresponding bit string y' with r length can be calculated. Check that, for all $1 \leq i \leq r$, when $y_i' = 0$, $z_i = 0$.

Bloom filter [16] can be used as a cryptographic accumulator. Furthermore, Yum et al. [17] proved that it is superior to other symmetric accumulators. Secure Bloom filter consists of k hash functions $\{f_i\colon Y \longrightarrow X\}$. These functions actually belong to the hash family. Each hash function uniformly returns a vector index. To add a value to

Table 2: Descriptions of the cryptographic accumulator.

Description		
One-way accumulator [10]	One-way hash function [11]	A family of one-way hash functions is an infinite sequence of families of functions $\{H_\lambda\}_{\lambda\epsilon N}$, where $H_\lambda\{h_k\colon X_k \times Y_k \longrightarrow Z_k\}$, with the following properties: ① For any integer λ and any $h_k \epsilon H_\lambda$, $h_k(.,.)$ is computable in polynomial time in λ; ② for any probabilistic, polynomial-time algorithm A, (3) is satisfied, where the probability is taken over the random choice of h_k, x, y, y' and the random coins of A.
	Quasi-commutativity	A function $f\colon X \times Y \longrightarrow X$ is said to be quasi-commutative if (4) is satisfied.
	One-way accumulator	A one-way accumulator is defined as a family of one-way hash functions with quasi-commutativeness. This description is elegant and simple, but, in order to clarify the basic function of the security cryptographic accumulator, the ability to intuitively accumulate set L as a small value can be proved only for element $y \in L$. In fact, the one-way property imposed by the second requirement is often too weak for applications where the attacker can choose some value to accumulate.
	Strongly one-way hash function	A family of strongly one-way hash functions is an infinite sequence of families of functions $\{H_\lambda\}_{\lambda\epsilon N}$, where $H_\lambda = \{h_k\colon X_k \times Y_k \longrightarrow Z_k\}$, having the following properties: ① For any integer λ and any $h_k \epsilon H_\lambda$, $h_k(.,.)$ is computable in polynomial time in λ; ② for any probabilistic, polynomial-time algorithm A, (7) is satisfied, where the probability is taken over the random choice of h_k, x, y, y' and the random coins of A.
Collision-free accumulator [5]	Cryptographic accumulator scheme	The cryptographic accumulator scheme is a 4-tuple of polynomial-time algorithm (Gen, Eval, Wit, and Ver)
	N-times collision-freeness	A cryptographic accumulator scheme is said to be N-times collision-free if, for any integer λ and for any probabilistic, polynomial-time algorithm A, probability is taken from Gen, Eval, and random coins of A.
	Collision-free	When a cryptographic accumulator scheme is N-times collision-free for any value of N polynomial in λ, it is called collision-free.
Dynamic accumulator [12]		Dynamic accumulators include polynomial-time algorithms (Gen, Eval, Wit, Ver, Add, Del, and Upd) for 7-tuples.
Universal accumulator [13]		Universal accumulators are dynamic and support membership and nonmembership proofs.

the cryptographic accumulator, it is fed to each hash function to get k indexes. The bit of x at these indexes is set to 1. To verify that a given value is accumulated, k hash functions are applied again to obtain the vector index. If any bit of the accumulative vector is 0 at these indexes, then the value is definitely not accumulated. If all the bits at these indexes are 1, then an incorrect positive response may be obtained. Another variant of Bloom filter has been studied in the past, where the hash function is replaced by a hash-based message authentication code (HMAC).

It can be noted that, in the case of symmetric accumulators, the size of l increases as the number of elements in the filter increases or the false positive rate is set as low.

3.2. Asymmetric Accumulators. The first cryptographic accumulator proposed is asymmetric and requires witness verification [1]. This construct takes the modulus $f\,(x, y) = x^y \bmod N$ as a one-way and quasi-commutative function because it satisfies

$$f\left(f\left(x,\, y_1\right),\, y_2\right) = \left(x^{y_1}\right)^{y_2} = \left(x^{y_2}\right)^{y_1} = f\left(f\left(x,\, y_2\right),\, y_1\right). \tag{10}$$

For power operations for one-way accumulators, the module is chosen as the product of two safe prime numbers p and q of equal size. If $(p-1)/2$ is also a prime, prime p is safe. Malicious attacker who knows the accumulated value z may forge witness w for the randomly selected value y by finding the initial value x verifying $x^y \bmod N = z$. However, this is not feasible under the RSA assumption.

Table 3 shows the development of symmetric and asymmetric accumulators.

4. Accumulator Based on Various Security Assumptions

Table 4 shows the evolution of different types of security assumptions.

4.1. Accumulator Based on Hash Tree

4.1.1. Hash Tree. Hash tree, in cryptography and computer science, is a tree data structure in which every leaf node is labeled with the hash of the data block, while the node other than the leaf node is labeled with the encrypted hash of its child node label. Hash trees can efficiently and securely validate the contents of large data structures. A prime resolution algorithm is selected to build a hash tree [20]. Consecutive primes starting at 2 are selected to build a ten-level hash tree. The node of the first layer is the root node, and there are two nodes under the root node. The second layer has three nodes under each node, and so on; that is, the number of children of each node layer is a continuous prime number. By the tenth level, there are 29 nodes under each node. The children of the same node, from left to right,

Table 3: Development process.

Symmetric accumulator	Bloom filter constructs an cryptographic accumulator. A symmetric accumulator is proposed to generate hash values for the values to be accumulated hash functions.
Asymmetric accumulator	The first cryptographic accumulator proposed in 1993 is asymmetric and requires witness validation.

Table 4: Security assumptions.

Hash-based construction	1993 [1]	Proposing the first cryptographic accumulator which is asymmetric and requires witness verification.
	2000 [18] 2002 [19]	Proposing the first universal dynamic accumulator satisfying nonrepudiation.
	2008 [20]	A universal accumulator structure based on the hash tree is proposed, which satisfies the concept similar to the nonrepudiation, called a strong universal accumulator.
RSA assumption	1996 [15]	Imposing one-way property (applications where some adversaries can access the list of values to accumulate may not succeed).
Strong RSA assumption	1999 [21]	Proposing a trapdoor-free accumulator.
	2002 [2]	Dynamic accumulator.
	2005 [22]	Dynamic accumulator for bilinear pairs.
t-SDH	2004 [23]	Elliptic curves construction of cryptographic accumulator.
	2007 [24]	(Nonmembership proof is inevitable) providing a dynamic accumulator, then called a universal accumulator.
	2009 [25]	Dynamic pairing accumulator (more efficient witness update algorithm).
t-DHE	2005 [26]	Bilinear map accumulator.
	2009 [25]	t-bound accumulator scheme based on t-DHE assumption is presented.

represent different remainder results. For example, the second layer node has three children. So, from left to right, 0 is divided by 3, 1 is divided by 3, and 2 is divided by 3. The remainder of the mod operation on a prime number determines the path of processing.

4.1.2. Accumulator Based on Hash Tree. In a hash tree, values are associated with the leaves of a binary tree. The value of the sibling node is hash in order to calculate the value associated with its parent node, and so on, until the value of the tree root is obtained. The root value of the tree is defined as the cryptographic accumulator of the set of values associated with the leaves of the tree [20]. The hash tree cannot be directly used to obtain the functions of general and dynamic accumulators. In fact, cumulative sets need to add and remove elements (tree node values if a hash tree is used), while generating nonmembership proof. So, instead of associating values with the leaves of the tree, a pair of continuously accumulated set elements are associated. To prove that element x is not in the accumulative set, it is now equivalent to indicating that a pair $(x_{\propto}, x_{\beta})$ (where $x_{\propto} < x < x_{\beta}$) belongs to the tree, but pairs $(x_{\propto}, x)$ and (x, x_{β}) do not belong to the tree.

4.1.3. Development Process of the Accumulator Based on Hash Tree. Buldas et al. [18, 19] proposed the first universal dynamic accumulator satisfying nonrepudiation (called the nonrepudiable certifier and formalized in the context of the cryptographic accumulator). Its construction is based on collision-resistant hashes and hash trees. Then, a universal accumulator structure based on hash tree is proposed, which satisfies the concept similar to nonrepudiation (the scheme is called strong universal accumulator). Recently, another cryptographic accumulator based on hash tree has been introduced, which uses the promise of modular operations on RSA composite modules based on binary polynomials as a collision-resistant hash function.

4.2. Accumulator Based on RSA Assumption

4.2.1. RSA Assumption. RSA hard problem means that, $\forall y, z, n \in Z_n^1, \exists x \in Z_n$: $z = x^y \bmod n$ is known. The RSA assumption refers to the fact that the RSA assumption is computationally infeasible for all polynomial-time algorithms A [5]; that is,

$$Pr\{y, z, n \in Z_n :: \exists x\colon z = x^y \bmod n\} \leq \frac{1}{A(n)}. \tag{11}$$

According to the RSA hard problem assumption, first, the function $z = x^y \bmod n$ satisfies the one-way property. Second, the function $z = x^y$ satisfies the quasi-commutativeness. That is, $\forall\, y_{1,}\, y_2$: $z(z(x,\ y_{1,}),\ y_2) = (x^{y_1})^{y_2} = x^{y_1 y_2} = z(z(x,\ y_2),\ y_1)$ is established.

When the modulus N is large enough and is generated randomly and the exponential y and value z are given, it is difficult to calculate x satisfying $x^y \bmod N = z$. However, as informally noted in [1] and later recognized in Nyberg [15], the one-way property imposed in the description may not succeed for applications where certain adversaries have access to the list of values to accumulate. To remedy, a stronger property called strongly one-way property should be considered, where choices do not impose y' on the attacker as one-way hash functions.

4.2.2. Strong RSA Assumption. The strong RSA hard problem means that, $\forall z$, $n \in Z_s$, $\exists x \in Z_p$, y: $z = x^y \bmod n$ is known, where Z_p is the set of prime numbers. The strong

RSA hard problem assumption means that the strong RSA hard problem is computationally infeasible for all polynomial-time methods A [2]; that is,

$$Pr\left\{z \in Z_n, n \in Z_s :: \exists x \in Z_n, y \in Z_p: z = x^y \bmod n\right\} \leq \frac{1}{A(n)}. \tag{12}$$

In contrast to general RSA, the strong RSA hard problem assumption allows free choice of combinations (x, y); that is, the attacker can choose not only the base of the exponential function but also the exponent. In addition, the strong RSA assumption also requires that the exponent be prime, while the general RSA assumption has no special requirement for the exponent. For the strong RSA hard problem assumption, there is no strict proof that it is computationally feasible. Again, there is no rigorous theoretical proof that it works on a computer.

When the modulus N is large enough and randomly generated and given the value z, it is difficult to find x and y that satisfy $x^y \bmod N = z$ as previously demonstrated; impact resistance can be obtained under strong RSA assumptions only if the value to be accumulated is prime.

Cryptographic accumulators without trapdoor should be able to be constructed. Trapdoors are unnecessary in the cryptographic accumulator scheme. The side that provides N during system setup also knows trapdoors p and q. Unfortunately, the side that knows p and q can completely bypass the security of the system. Because by knowing p and q, it is possible to recover the initial value and then independently accumulate additional values and generate false witnesses. A trapdoor-free solution will not rely on trusted online or offline services. Then a trapdoor-free accumulator is introduced, which is proved to be safe in the standard model. The authors suggest the use of a generalized RSA module with unknown complete factorization and call it RSA-UFOS. A number N is an RSA-UFO, and if N has at least two large prime factors p and q, then no participant in the union, including those that produce N, will be able to find an N that splits into factors N_1 and N_2, thus making $P|N_1$ and $q|N_2$. A probabilistic algorithm is also proposed to generate such numbers. Under the standard model, security is proved under a new assumption called “strong RSA-UFO assumption.” This assumption is very similar to the strong RSA assumption, with the only difference being that module N is set to RSA-UFO.

4.2.3. Accumulator Based on Strong RSA Assumption. All schemes in this setting are [1, 5] extensions. The accumulator acc_x is defined as $\mathrm{acc}_x \longleftarrow g^{\prod_{x\in X} x} \bmod N$, where N is an RSA modulus consisting of two large safe prime numbers p and g, which is randomly drawn from the cyclic group of the quadratic remainder of N. There are $sk_{\mathrm{acc}}, pk_{\mathrm{acc}} = (p, q, N)$ and the witnesses of the value x_i given by $\mathrm{wit}_{x_i} \longleftarrow \mathrm{acc}_x^{x_i^{-1}} \bmod N$. Obviously, if the value x_i not included in acc can forge witness $\mathrm{wit}_{x_i} \longleftarrow \mathrm{acc}_x^{x_i^{-1}} \bmod N$, then the strong RSA assumption will be broke. Because of the product relation of the accumulated value in the exponent, the domain of the accumulated value is limited to prime number. Note that when a given witness wit_a (i.e., $\mathrm{wit}_b \equiv \mathrm{wit}_a^c \pmod N$), accumulating a compound number will allow $a = b \cdot c$ derivation of the witness for each of its factors, to accumulate sets from more general domains, an appropriate mapping from these domains to prime numbers will be required (see [27]).

Certain cryptographic accumulator schemes in this setting [2] also provide dynamic functionality. Simply summing the cryptographic accumulator and its witness can add values to the cryptographic accumulator without any secret. On the contrary, if the value x_j is to be deleted, the $x_j - th$ root of the cryptographic accumulator must be calculated, which is difficult to solve under strong RSA assumptions without sk_{acc}. However, after removing the value, membership witnesses can still be publicly updated using arithmetic techniques. To update the witness acc_{x/x_j} of the value x_i, find $a, b \in Z$, so that $ax_i + bx_j = 1$ and calculate the new witness as $\mathrm{wit}_{x_i}' \longleftarrow \mathrm{wit}_{x_i}^b \cdot \mathrm{acc}_{x/x_j} \bmod N$ and original witness is wit_{x_i}.

Moreover, cryptographic accumulator scheme provides general functionality because it supports nonmembership witnesses: acc_X is accumulator for set X and $y_j \notin X$. Now it holds that $\gcd \prod_{x\in X} x, y_j = 1$ or equivalently for $a, b \in Z$, $a \prod_{x\in X} x + by_j = 1$. Therefore, $d \longleftarrow g^{-b} \bmod N$ is calculated, where g is the initial value of the empty cryptographic accumulator and forms a nonmembership witness $\mathrm{wit}_{y_i} \longleftarrow (a, b)$. Then, the verification of nonmembership witnesses is completed by checking whether $\mathrm{acc}_x^a \equiv d^{y_i} \cdot g \pmod N$ is established. Similar to what is done for membership, nonmembership witnesses can also be publicly updated (see [24]).

4.3. Accumulator Based on t-SDH Assumption

4.3.1. t-SDH Assumption. Given a tuple $t = (p, G, P)$, where p is prime, G is a cyclic group generated by P and a tuple in the form of value $(P, sP, \ldots s^t P)$ in Z/pZ, where $s \in Z/pZ\{0\}$ [8]. For any probabilistic polynomial-time algorithm A, the following probabilities can be negligible:

$$Pr = \left[A\left(tP, sP \ldots, s^t P\right) = \left(c, \frac{1}{s + c} P\right) \Lambda \left(c \in \frac{Z}{pZ}\right)\right]. \tag{13}$$

Tartary et al. [28] made requirements for the conflict resistance performance of the scheme, thus refuting previous claims against cryptographic accumulators. Attack is based on improperly defined security models in which adversaries have access to functions f and g. The proposed patch includes providing compound functions g (f(.)) to the adversary instead of providing functions f and g, respectively. However, the patches proposed by the authors cannot prevent other types of attacks and have proved the scheme to be unsafe. Camenisch et al. [25] proposed another cryptographic accumulator based on dynamic pairing, which provides a more efficient witness update algorithm.

Fazio and Nicolosi [7] pointed out in their investigation of the cryptographic accumulator that the original structure makes the time to update the witness after m changes the cryptographic accumulator proportional to m. They raised the question of whether batch updates are possible, that is, whether it is possible to build a cryptographic accumulator where the time to update the witness is independent of the number of changes to the cryptographic accumulator set. Wang et al. [29] designed a cryptographic accumulator with batch processing update and then made improvements to solve the above problems. The scheme is based on the Paillier cipher system and is proven to be secure under a new assumption called the extended strong RSA assumption, which is a variant of the strong RSA assumption with modulus N_2. However, contrary to this claim, Camacho and Hevia [30] have shown evidence of an attack and further demonstrated that the time to update the witness in the worst case must be at least $\Omega(m)$. Therefore, this provides impossible results on a cryptographic accumulator with batch update capabilities.

Previous works have produced only membership witnesses, but, in some cases, nonmembership witnesses may be unavoidable. The authors present a dynamic accumulator that supports both membership and nonmembership short witnesses, which they call the universal accumulator. The initial value of the cryptographic accumulator must be public so that nonmembership witnesses can be verified. This construct is based on the RSA function, so only prime numbers are allowed to accumulate.

Karlof et al. [23] used elliptic curves to construct cryptographic accumulators. To add up the values (scalars), multiply them by the public key (i.e., scalars multiply the base point of the curve). Witness generation follows the same algorithm but does not include corresponding values. Validation is simple; if the product of the witness and the value is equal to the accumulated value, it is necessary to check for equality.

4.3.2. Accumulator Based on t-SDH Assumption. Nguyen [22] proposed a t-bound accumulator. The cryptographic accumulator uses a group G of prime number p generated by g and has bilinear maps e: $G \times G \longrightarrow G_T$. Here, $pk_{\mathrm{acc}} = (g, g^s, g^{s^2}, \ldots, g^{s^t}, u)$ and $\mathrm{sk}_{\mathrm{acc}} = \mathrm{s}$. The accumulator acc_x of set $X = \{x_1, x_2, \ldots, x_n\} \in Z_p$ $(n \ \leq t)$ is defined as $\mathrm{acc}_x \longleftarrow g^{u \prod_{x \in X}(x+s)}$, and the membership witness $\mathrm{wit}_{x_i} \longleftarrow g^{u \prod_{x \in X \setminus x_i}(x+s)}$ is calculated, where $u \overset{\mathrm{R}}{\longleftarrow} Z_p^*$. Then, check whether acc_X contains the value x_i by verifying whether $\mathrm{eacc}_x,\ g = e(g^{x_i} g^s, \mathrm{wit}_{x_i})$ is true or not. The scheme allows the public evaluation of cryptographic accumulators; that is, $g^{h(s)}$ is obtained by extending polynomial $hx = \prod_{x \in X}(x + s) \in Z_p[X]$ and by evaluating it in G through pk_{acc}. The public calculation of the witnesses of x_i also works on set $X/\{x_i\}$. Furthermore, these witnesses can be updated at a constant time without knowing the secret key (see [22]).

Nguyen's scheme is extended by nonmembership witnesses, and the random value u is eliminated [31, 32]. Previous work also showed how to publicly update nonmembership witnesses within a fixed period of time. Note that these adjustments can also be applied to the latter [31]. The calculation of nonmembership witnesses with value $y_j \notin X$ makes use of the following facts: $hx = \prod_{x \in X}(x + X)$ is divided by the polynomial division remainder of $y_j + X$. Such witnesses take the form of $a, b = (g^{hs - d/y_i + s}, d))$ and may be validated by $\mathrm{eacc}_x, g \overset{?}{=} ea, g^{y_i}, g^s e(g, g^s))$.

4.4. Accumulator Based on t-DHE Assumption

Diffie–Hellman Exponent (DHE) Assumption. The t-DHE problem in a group G of prime order q is defined as follows: Let $g_i = g^{\gamma^i}$, $\gamma \longleftarrow {}_R Z_q$. On input $\{g, g_1, g_2, \ldots, g_t, g_{t+2}, \ldots, g_{2t}\} \in G^{2t}$, output g_{t+1}.

The t-DHE assumption states that this problem is hard to solve.

Camenisch et al. [25] gave a scheme of t-bound accumulator based on t-DHE assumption, like the cryptographic accumulator in t-SDH settings, which uses a group G of prime number p generated by g and has bilinear mapping e: $G \times G \longrightarrow G_T$. Besides, it needs a signature scheme with corresponding key pairs $(sk_{\mathrm{sig}}, pk_{\mathrm{sig}})$. Here, $sk_{\mathrm{acc}} = sk_{\mathrm{sig}}$, public key is $pk_{\mathrm{acc}} = g_1, \ldots, g_t,\ g_{t+2}, \ldots, g_{2t},\ z,\ pk_{\mathrm{sig}} = (g^{\gamma^1}, \ldots,\ g^{\gamma^t}, g^{\gamma^{t+2}}, \ldots, g^{\gamma^{2t}}, e(g, g)^{\gamma^{t+1}}, pk_{\mathrm{sig}})$, and $\gamma \overset{R}{\longleftarrow} Z_p^*$. $X\{x_1, \ldots, x_m\}$ can be accumulated by calculating $\mathrm{acc}_X \longleftarrow \prod_{i=1}^m g_{t+1-i}$ and signing g_i with x_i using sk_{sig}, where $m \leq t$, thus assigning the value of x_i to g_i. The witness wit_{x_j} of $x_j \in X$ is $\mathrm{acc}_X \longleftarrow \prod_{i=1, i \neq j}^m g_{t+1-i+j}$. The membership of x_j can be verified by checking whether $e\, g_j, \mathrm{acc}_X = z \cdot e(g, \mathrm{wit}_{x_j})$ is valid and verifying the signatures of g_j and x_j under pk_{sig}.

This scheme allows public updates for witnesses and cryptographic accumulators to be deleted, as this requires only pk_{acc}. However, if the value x_i is to be added to the cryptographic accumulator, a secret signature key sk_{acc} is required to create signatures on g_i and x_i to link the value x_i to this parameter. Therefore, the public addition of the cryptographic accumulator requires that a signature be included for each potential value to be stored in the public parameter. Obviously, this seems impractical except for the small accumulative domain.

5. Cryptographic Accumulator Schemes in the Hidden Order Group and Known Order Group

Since the introduction of cryptographic accumulator, many cryptographic accumulator schemes with different characteristics have been proposed. Basically, the main work is to construct schemes in hidden order group and known order group [33].

5.1. Hidden Order Group. The original RSA-based schemes have been developed by Baric, which enhance the original concept of collision-free safety. Sander [21] suggested using unknown decomposed RSA modules to construct trapdoor-free accumulators. Camenisch extended the previous scheme to have the ability to dynamically add/delete values to the cryptographic accumulator, which constitutes the first dynamic accumulator scheme. Their plan also supports public updates of existing witnesses, that is, updates without knowing any trapdoor. After that, support for nonmembership witnesses was added, so a universal dynamic accumulator was obtained. They also proposed an optimization scheme to update the documents of nonmembership witnesses more effectively but later found shortcomings [34, 35]. Lipmaa [36] generalized the RSA accumulator to a module over a Euclidean ring. In all the above schemes, the accumulative domain is limited to primes to ensure that there is no conflict. Tsudik and Xu [37] proposed a variant, which allows the accumulation of semiprimes. Assuming that the semiprime used is difficult to decompose and its decomposition is unknown to the public, a collision-free accumulator is obtained. In addition, a cryptographic accumulator scheme is proposed, which allows arbitrary integers to be accumulated and supports batch updates of witnesses. However, the scheme was eventually broken.

5.2. Known Order Group. Nguyen proposed a dynamic accumulator scheme, which is suitable for paired-friendly groups with prime p. It is secure under the t-SDH assumption and allows up to t values to be accumulated from domain Z_p. Later, Damgard, Triandopoulos, and Au et al. extended the scheme of Nguyen with general functions. Recently, Acar and Nguyen [38] removed the upper limit t for the number of elements accumulated by the t-SDH accumulator. To do this, they used a set of cryptographic accumulators, each of which contained a subset of the entire set to be accumulated. Camenisch et al. introduced another cryptographic accumulator scheme for pairing-friendly prime arrays. It supports public updates of witnesses and witnesses, and its security depends on the t-DHE assumption.

Table 5 shows the development of cryptographic accumulator schemes.

6. Cryptographic Accumulator Applications

6.1. Application of the Cryptographic Accumulator in Digital Signature

6.1.1. Ring Signature. In anonymous authentication on trusted platform, the length of ring signature is positively related to the number of ring members, while large members lead to low efficiency. Therefore, Xu et al. [40] proposed a ring signature anonymous authentication method based on the one-way accumulator and constructed its solution in detail. In the signature phase, the length of the ring is determined by a one-way accumulator, which accumulates the information of all members so that the ring is not too large for a considerable number of members. During the verification period, the efficiency is improved, and the hash computing time, encryption computing time, and decryption computing time are reduced. Compared with the typical ring signature, it is shown that the new solution has lower time complexity and space complexity. At the same time, the new solution ensures anonymity and validity, which not only makes up for the weakness of traditional ring signature but also has high efficiency under the premise of security.

6.1.2. Group Signature. Based on the knowledge of an accumulative composite dynamic accumulator and an effective protocol to prove that the factorization of a submitted value develops a novel, efficient, and provably secure group signature scheme [37], it allows authorization and ownership proof at the same time as factorization based on cumulative synthesis. It enables a group member to perform lightweight authorization proof so that the complexity of proof and verification is independent of the number of current or all deleted members. Using a dynamic accumulator to facilitate authorization, it is required that the group manager propagate certain information such as the value deleted from the cryptographic accumulator whenever a member (or group of members) joins or leaves the group.

6.2. Encrypted Search. The dynamic accumulator is introduced into the encrypted search scheme [41, 42], and the existing search scheme of decentralized storage based on block chain is improved. The new scheme takes advantage of the efficient verifiability of the witness in the dynamic accumulator and the dynamic addition and deletion of elements in the accumulated value and takes into account both efficiency and flexibility. In the encryption search scheme based on CCS'14 Hahn in [43], a dynamic accumulator is introduced and improved for the decentralized storage application scenario based on blockchain.

6.3. Revoking Anonymous Credentials. The dynamic accumulator can be used to revoke normal credentials (and certificates): First, add a unique value to each credential. Then, the accumulator value of the unique value of all valid credentials is truly published [44]. Now, users can convince the verifier that the credential is still valid by providing a witness for the unique value contained in their credential. Therefore, to check the credential, the verifier must check the publisher's signature to obtain the current accumulator value and use the witness provided by the user to verify that the unique value contained in the credential is included in the accumulator value.

For anonymous credentials, the same method can be used. However, the witnesses and values contained in the cryptographic accumulator can no longer be disclosed to the validator because this completely endangers anonymity. Instead, the user can apply zero-knowledge proof to convince the verifier that the values contained in its credentials are also included in the cryptographic accumulator. Therefore, if a valid protocol is found to prove that the values contained in the commitment are also included in the certificate, any anonymous certificate scheme can be effectively revoked.

6.4. Cryptographic Accumulator in Vector Commitment. Catalano and Fiore [45] proposed a black box construction of cryptographic accumulator based on vector commitment.

Table 5: Cryptographic accumulator schemes.

	Known order group
2005 [22]	A dynamic accumulator scheme is proposed, which is suitable for paired-friendly groups with prime p. It is secure under the t-SDH assumption and allows up to t values to be accumulated from the domain.
2008 [32]	Extended 2005 scheme with general functions.
	Hidden order group
	The accumulative domain is limited to primes
1997 [5]	Improved the original RSA scheme in 1993 and strengthened the original concept of collision-free safety.
1999 [21]	It is recommended to use unknown decomposed RSA modules to construct trapdoor-free accumulators.
2002 [2]	The scheme in 1997 is extended to have the ability to dynamically add/delete values to the cryptographic accumulator, and the first dynamic accumulator is constructed.
2007 [24]	In 2002, support for nonmembership witnesses was increased, so a universal dynamic accumulator was obtained, and an optimization scheme was proposed to update the documents of nonmembership witnesses more effectively.
2012 [34]	The RSA accumulator is broadly defined as a module over a Euclidean ring.
	The accumulative domain is limited to semiprimes
2003 [37]	It is allowed to accumulate semiprimes.
2007 [29]	The cryptographic accumulator scheme allows arbitrary integers to be accumulated and supports batch updates of witnesses.
2019 [39]	Dynamic accumulator based on hash greatly reduces storage space.
2011 [38]	The upper limit t for accumulating elements of the t-SDH accumulator is canceled.

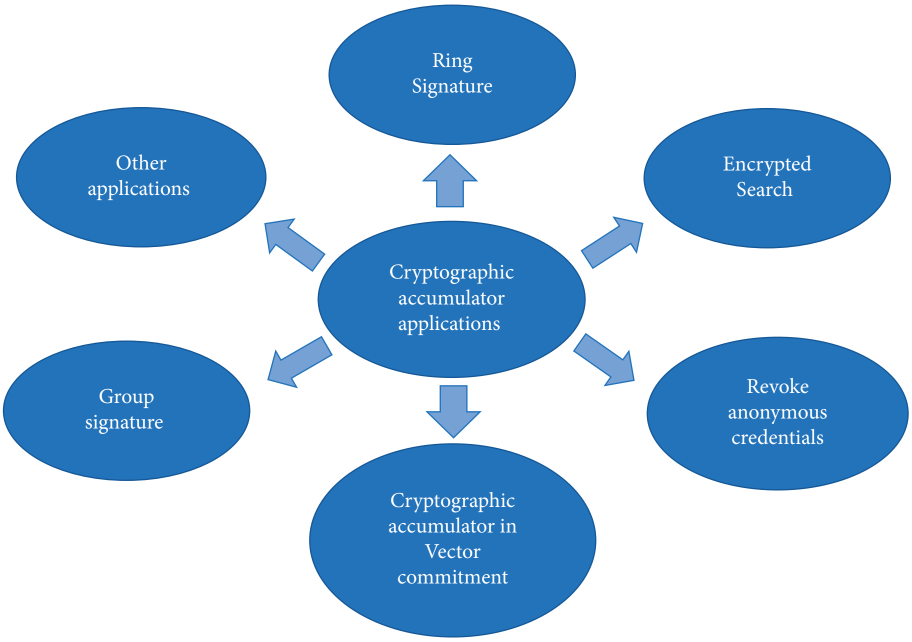
Ring Signature
Other applications
Encrypted Search
Cryptographic accumulator applications
Group signature
Revoke anonymous credentials
Cryptographic accumulator in Vector commitment

Figure 1: Cryptographic accumulator applications.

Vector commitment allows concise commitment C to be formed for vector $X = \{x_1, \ldots, x_n\}$. Here, it is not computationally feasible to open position i of C to a value x_i' different from that of x_i. The accumulative domain in the black box construction is set $D = \{1, \ldots, t\}$. The cryptographic accumulator is modeled as a commitment to a binary vector of length t; that is, each bit i represents the existence or nonexistence of element $i \in D$ in the cryptographic accumulator. Then, the (non)membership of value i can be proved by opening position i that is committed to 1 or 0, respectively.

6.5. Other Applications. The applications of the cryptographic accumulator are shown in Figure 1.

Cryptographic accumulators can be applied to membership testing, distributed signatures, responsible certificate management, and authenticated dictionaries and can also be used as editable, sanitary processing [46, 47], homomorphic signatures [48, 49], and privacy protection data outsourcing building blocks as for authenticated data structures [50, 51]. In addition, the cryptographic accumulator scheme can be used to prove the zero knowledge of (nonmembership) witnesses [52, 53], and undisclosed values are now widely used to revoke group signatures and anonymous credentials [54, 55]. Recently, cryptographic accumulators are also used in Zerocoin [56, 57], and Zerocoin is an anonymous extension of bitcoin cryptocurrencies. Therefore, the cryptographic accumulator can be applied to many aspects, and readers can understand the specific applications of the cryptographic accumulator in these aspects by consulting the above literature.

7. Conclusion

Cryptographic accumulator is a basic and important tool in the field of cryptography, which has been widely used in many aspects. This paper firstly introduces the types of cryptographic accumulators. Secondly, in the asymmetric accumulators, three different cryptographic accumulators schemes are classified through three security assumptions. Thirdly, several cryptographic accumulators based on security assumptions are introduced. Fourthly, this paper presents the cryptographic accumulator scheme under different characteristics. Finally the applications of cryptographic accumulators in different aspects are summarized. With the rapid development of big data security and blockchain, cryptographic accumulators are used more and more widely, and there is still much development space in the future.

Data Availability

All data supporting the findings of this study are included within the article.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by Open Project Foundation of Information Security Evaluation Center of Civil Aviation, Civil Aviation University of China (CAAC-ISECCA-202001), and it was also supported by the NSFC (62072249) and the PAPD Fund from NUIST.

References

[1] J. Benaloh and M. D. Mare, "One-way accumulators: a decentralized alternative to digital signatures," in *Proceedings of the Advances in Cryptology — EUROCRYPT '93*, pp. 274–285, Lofthus, Norway, May 1993.

[2] J. Camenisch and A. Lysyanskaya, "Dynamic accumulators and application to efficient revocation of anonymous credentials," in *Proceedings of the Advances in Cryptology - CRYPTO 2002*, pp. 61–76, CA, USA, August 2002.

[3] A. Biryukov, A. Udovenko, and G. Vitto, "Cryptanalysis of Au et al. Dynamic universal accumulator," *IACR Cryptol. ePrint Arch*, vol. 2020, no. 2020, p. 598, 2020.

[4] K. Nyberg, "Commutativity in cryptography," *Journal of Progressive Human Services*, vol. 13, no. 1, pp. 186–188, 1996.

[5] N. Baric and B. Pfitzmann, "Collision-free accumulators and fail-stop signature schemes without trees," in *Proceedings of the Advances in Cryptology — EUROCRYPT*, pp. 480–494, Konstanz, Germany, May 1997.

[6] I. Ozcelik, S. Medury, J. Broaddus, and S. Anthony, "An overview of cryptographic accumulators," 2021, https://arxiv.org/abs/2103.04330.

[7] N. Fazio and A. Nicolosi, "Cryptographic accumulators: definitions, constructions and applications," 2002.

[8] A. Kumar, P. Lafourcade, and C. Lauradoux, "Performances of cryptographic accumulators," in *Proceedings of the 39th Annual IEEE Conference on Local Computer Networks*, pp. 366–369, Edmonton, AB, Canada, September 2014.

[9] G. Vitto and A. Biryukov, "Dynamic universal accumulator with batch update over bilinear groups," *IACR Cryptol*, vol. 2020, no. 2020, p. 777, 2020.

[10] L. T. Tang, M. Xu, and Y. R. Jing, "Research on efficiency optimization method of identity authentication mechanism based on blockchains," *Application Research of Computers*, vol. 36, no. 6, pp. 2783–2787+2791, 2019.

[11] Y. J. Ren, F. J. Zhu, T. Wang et al., "Tolba Amr. Data query mechanism based on hash computing power of blockchain in Internet of Things," *Sensors*, vol. 20, no. 1, pp. 1–22, 2020.

[12] T. Dryja, "Utreexo: a dynamic hash-based accumulator optimized for the Bitcoin UTXO set," *IACR Cryptol. ePrint Arch*, vol. 611, 2019.

[13] M. P. Jhanwar and P. R. Tiwari, "Trading accumulation size for witness size: a merkle tree based universal accumulator via subset differences," *IACR Cryptol. ePrint Arch*, vol. 2019, p. 1186, 2019.

[14] Y. Ren, J. Qi, Y. Liu, J. Wang, and G. J. Kim, "Integrity verification mechanism of sensor data based on bilinear map accumulator," *ACM Transactions on Internet Technology*, vol. 21, no. 1, pp. 1–19, 2021.

[15] K. Nyberg, "Fast accumulated hashing," *Fast Software Encryption*, vol. 1039, pp. 83–87, 1996.

[16] B. H. Bloom, "Space/time trade-offs in hash coding with allowable errors," *Communications of the ACM*, vol. 13, no. 7, pp. 422–426, 1970.

[17] D. H. Yum, J. W. Seo, and P. J. Lee, "Generalized combinatoric accumulator," *IEICE - Transactions on Info and Systems*, vol. E91-D, no. 5, pp. 1489–1491, 2008.

[18] A. Buldas, P. Laud, and H. Lipmaa, "Accountable certificate management using undeniable attestations," in *Proceedings of the 7th ACM Conference on Computer and Communication Security 2000*, pp. 9–17, Athens Greece, November 2000.

[19] A. Buldas, P. Laud, and H. Lipmaa, "Eliminating counter-evidence with applications to accountable certificate management1," *Journal of Computer Security*, vol. 10, no. 3, pp. 273–296, 2002.

[20] P. Camacho, A. Hevia, M. A. Kiwi, and R. Opazo, "Strong accumulators from collision-resistant hashing," in *Proceedings of the Information Security, 11th International Conference, ISC 2008*, pp. 471–486, Taipei, Taiwan, September 2008.

[21] T. Sander, "Efficient accumulators without trapdoor extended abstract," in *Proceedings of the International Conference on Information and Communications Security*, pp. 252–262, Sydney, NSW, Australia, November 1999.

[22] L. Nguyen, "Accumulators from bilinear pairings and applications," in *Proceedings of the Cryptographers' Track at the RSA Conference*, pp. 275–292, San Francisco, CA, USA, February 2005.

[23] C. Karlof, N. Sastry, Y. Li, J. D. Tyger, and P. Adrian, "Distillation codes and applications to DoS resistant multicast authentication," in *Proceedings of the Network and Distributed*

System Security Symposium - NDSS, pp. 37–56, CA, USA, February 2004.

[24] J. Li, N. Li, and R. Xue, "Universal accumulators with efficient nonmembership proofs," in *Proceedings of the International Conference on Applied Cryptography and Network Security*, pp. 253–269, Zhuhai, China, June 2007.

[25] J. Camenisch, M. Kohlweiss, and C. Soriente, "An accumulator based on bilinear maps and efficient revocation for anonymous credentials," in *Proceedings of the International workshop on public key cryptography*, pp. 481–500, CA, USA, March 2009.

[26] D. Boneh, X. Boyen, and E.-J. Goh, "Hierarchical identity based encryption with constant size ciphertext," in *Proceedings of the Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pp. 440–456, Aarhus, Denmark, May 2005.

[27] W. I. Khedr, H. M. Khater, and E. R. Mohamed, "Cryptographic accumulator-based scheme for critical data integrity verification in cloud storage," *IEEE Access*, vol. 7, Article ID 65651, 2019.

[28] C. Tartary, S. Zhou, D. Lin, H. Wang, and J. Pieprzyk, "Analysis of bilinear pairing-based accumulator for identity escrowing," *IET Information Security*, vol. 2, no. 4, pp. 99–107, 2008.

[29] P. Wang, H. Wang, and J. Pieprzyk, "A new dynamic accumulator for batch updates," *Information and Communications Security*, vol. 4861, pp. 98–112, 2007.

[30] P. Camacho and A. Hevia, "On the impossibility of batch update for cryptographic accumulators," *Lecture Notes in Computer Science*, vol. 6212, pp. 178–188, 2010.

[31] M. H. Au, P. P. Tsang, W. Susilo, and Y. Mu, "Dynamic universal accumulators for DDH groups and their application to attribute-based anonymous credential systems," in *Proceedings of the Cryptographers' Track at the RSA Conference*, pp. 295–308, CA, USA, February 2009.

[32] I. Damgard and N. Triandopoulos, "Supporting non-membership proofs with bilinear-map accumulators," 2008.

[33] G. Schul-Ganz and G. Segev, "Accumulators in (and beyond) generic groups: non-trivial batch verification requires interaction," *Theory of Cryptography*, vol. 1106, pp. 77–107, 2020.

[34] A. Mashatan and S. Vaudenay, "A fully dynamic universal accumulator," *Proceedings of the Romanian Academy*, vol. 14, pp. 269–285, 2013.

[35] K. Peng and F. Bao, "Vulnerability of a non-membership proof scheme," in *Proceedings of the International Conference on Security and Cryptography*, pp. 1–4, Athens, Greece, July 2010.

[36] H. Lipmaa, "Secure accumulators from euclidean rings without trusted setup," *Applied Cryptography and Network Security*, vol. 7341, pp. 224–240, 2012.

[37] G. Tsudik and S. Xu, "Accumulating composites and improved group signing," in *Proceedings of the Advances in Cryptology - ASIACRYPT 2003*, pp. 269–286, Taipei, Taiwan, December 2003.

[38] T. Acar and L. Nguyen, "Revocation for delegatable anonymous credentials," in *Proceedings of the Public Key Cryptography - PKC 2011*, pp. 423–440, Taormina, Italy, March 2011.

[39] D. Boneh, B. Bünz, and B. Fisch, "Batching techniques for accumulators with applications to iops and stateless blockchains," in *Proceedings of the Annual International Cryptology Conference*, pp. 561–586, Santa Barbara, USA, August 2019.

[40] Z. M. Xu, T. Hao, D. S. Liu, and J. Lin, "A ring-signature anonymous authentication method based on one-way accumulator," in *Proceedings of the 2010 Second International Conference on Communication Systems, Networks and Applications*, pp. 56–59, Hong Kong, China, June 2010.

[41] C. Ge, W. Susilo, Z. Liu, J. Xia, P. Szalachowski, and F. Liming, "Secure keyword search and data sharing mechanism for cloud computing," *IEEE Transactions on Dependable and Secure Computing*, vol. 20, no. 8, p. 1, 2020.

[42] Y. Ren, F. Zhu, J. Wang, P. K. Sharma, and U. Ghosh, "Novel vote scheme for decision-making feedback based on blockchain in internet of vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 8, pp. 1–10, 2021.

[43] Y. Zhang, J. F. Wang, Z. Y. Qi, R. Yang, and Y. Wang, "Decentralized encryption search scheme based on dynamic accumulator," *Journal of Network and Information Security*, vol. 5, no. 2, pp. 23–29, 2019.

[44] L. Xu, C. Xu, Z. Liu, Y. Wang, and J. Wang, "Enabling comparable search over encrypted data for IoT with privacy-preserving," *Computers, Materials & Continua*, vol. 60, no. 2, pp. 675–690, 2019.

[45] D. Catalano and D. Fiore, "Vector commitments and their applications," in *Proceedings of the Public-Key Cryptography - PKC 2013*, pp. 55–72, Nara, Japan, March 2013.

[46] S. Canard and A. Jambert, "On extended sanitizable signature schemes," in *Proceedings of the Topics in Cryptology - CT-RSA 2010*, pp. 179–194, CA, USA, March 2010.

[47] Y. Ren, J. Qi, Y. Cheng, J. Wang, and J. Xia, "Digital continuity guarantee approach of electronic record based on data quality theory," *Computers, Materials & Continua*, vol. 63, no. 3, pp. 1471–1483, 2020.

[48] J. H. Ahn, D. Boneh, J. Camenisch, S. Hohenberger, A. Shelat, and B. Waters, "Computing on authenticated data," *Theory of Cryptography*, vol. 7194, pp. 1–20, 2012.

[49] C. Ge, W. Susilo, J. Baek, Z. Liu, J. Xia, and L. Fang, "Revocable attribute-based encryption with data integrity in clouds," *IEEE Transactions on Dependable and Secure Computing*, vol. 21, no. 3, p. 1, 2021.

[50] L. Fang, Y. Li, X. Yun et al., "THP: a novel authentication scheme to prevent multiple attacks in SDN-based IoT network," *IEEE Internet of Things Journal*, vol. 7, no. 7, pp. 5745–5759, 2020.

[51] Y. Ren, Y. Leng, Y. Cheng, and J. Wang, "Secure data storage based on blockchain and coding in edge computing," *Mathematical Biosciences and Engineering*, vol. 16, no. 4, pp. 1874–1892, 2019.

[52] F. Baldimtsi, J. Camenisch, M. Dubovitskaya, A. Lysyanskaya, L. Reyzin, and K. Same, "Accumulators with applications to anonymity-preserving revocation," in *Proceedings of the 2017 IEEE European Symposium on Security and Privacy (EuroS&P)*, pp. 301–315, Paris, France, April 2017.

[53] Z. T. Li, J. W. Kang, R. Yu, D. Ye, Q. Deng, and Y. Zhang, "Consortium blockchain for secure energy trading in industrial Internet of Things," *IEEE Transactions on Industrial Informatics*, vol. 8, no. 14, pp. 3690–3700, 2018.

[54] C. Ge, Z. Liu, J. Xia, and L. Fang, "Revocable identity-based broadcast proxy re-encryption for data sharing in clouds," *IEEE Transactions on Dependable and Secure Computing*, vol. 18, no. 3, pp. 1214–1226, 2021.

[55] L. Fang, M. Li, Z. Liu et al., "A secure and authenticated mobile payment protocol against off-site attack strategy," *IEEE Transactions on Dependable and Secure Computing*, vol. 21, no. 8, pp. 77–90, 2021.

[56] I. Miers, C. Garman, M. Green, and A. D. Rubin, "Zerocoin: anonymous distributed e-cash from bitcoin," in *Proceedings of*

the IEEE Symposium on Security and Privacy, pp. 397–411, Berkeley, CA, USA, May 2013.

[57] C. Ge, W. Susilo, J. Baek, Z. Liu, J. Xia, and L. Fang, "A verifiable and fair attribute-based proxy re-encryption scheme for data sharing in clouds," *IEEE Transactions on Dependable and Secure Computing*, vol. 21, no. 7, p. 1, 2021.