Journal of Network and Computer Applications 127 (2019) 43–58

Contents lists available at ScienceDirect

Journal of Network and Computer Applications

journal homepage: www.elsevier.com/locate/jnca

Cryptographic primitives in blockchains

Licheng Wang[a], Xiaoying Shen[a], Jing Li[b,*], Jun Shao[c], Yixian Yang[a]

[a] State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing 100876, China
[b] Guangzhou University, China
[c] Department of Information Security, Zhejiang Gongshang University, China

ARTICLE INFO

Keywords:
Blockchain
Hash function
Accumulator
Commitment
Zero-knowledge proofs

ABSTRACT

Blockchain, as one of the crypto-intensive creatures, has become a very hot topic recently. Although many surveys have recently been dedicated to the security and privacy issues of blockchains, there still lacks a systematic examination on the cryptographic primitives in blockchains. To this end, we in this paper conduct a systematic study on the cryptographic primitives in blockchains by comprehensive analysis on top-30 mainstream cryptocurrencies, in terms of the usages, functionalities, and evolutions of these primitives. We hope that it would be helpful for cryptographers who are going to devote themselves to the blockchain research, and the financial engineers/managers who want to evaluate cryptographic solutions for blockchain-based projects.

1. Introduction

Since its introduction in the early 1980s (Chaum, 1982), the design of e-cash has always been one of the main research topics in the field of cryptography. However, the one without any trusted third party remained an open problem till Bitcoin (Nakamoto) launched in 2009. Due to its decentralization, unforgeability, double-spending resistance and pseudonymity, this brand new e-cash system has brought a remarkable culmination of cryptocurrency research and its applications. Based on its main framework, many new cryptocurrencies including decentralized (such as (Litecoin), Nxtcoin (Nxt)) and centralized ones (such as RScoin (Danezis and Meiklejohn, 2016)) have been proposed. The market value of these cryptocurrencies has increased more than 30 times during 2017 (from about $17 billion on 1st Jan. to $591 billion on 31st Dec.) (Coinmarketcap). As the core technology behind Bitcoin, the blockchain has demonstrated its capability of innovation and infiltration in many domains, including finance, insurance, industry, healthcare, agriculture and so on (Romano and Schmid, 2017; Tasca et al., 2017). For example, the blockchain technology, combined with the cloud computing and Intel® SGX (Software Guard Extensions), would give us an opportunity for alleviating cost and risk caused by trust third parties (Romano and Schmid, 2017), which will have a pervasive impact on the future of our society (Tasca et al., 2017). However, Bitcoin is not yet an ideal e-cash system. The privacy leakage is one of the main problems. For instance, anyone can see the payer's bitcoin address, payee's bitcoin address, and the content of each transaction in the bitcoin blockchain. To solve this issue, many advanced cryptographic primitives, such as ring signature (van Saberhagen, 2013), zero-knowledge proof (Miers et al., 2013; Ben-Sasson et al., 2014a) have been adopted in blockchains.

There are many recent surveys have been dedicated to the security and privacy issues of blockchains (Androulaki et al., 2013; Tschorsch and Scheuermann, 2016; Zhu et al., 2017; Li et al., 2018; Meng et al., 2018; Tao et al., 2018). Androulaki et al. (2013) evaluated the privacy in Bitcoin by simulating the use of Bitcoin in the university. Tschorsch and Scheuermann (2016) discussed security risks and security implications of the Bitcoin protocol. Zhu et al. (2017) described the attack method against privacy in the existing blockchains in detail and introduced the privacy protection mechanism. Li et al. (2018) conducted a systematic study on the security threats to blockchain and surveyed the corresponding real attacks by examining popular blockchain systems. Meng et al. (2018) discussed the applicability of blockchain to intrusion detection. Tao et al. (2018) proposed a lightweight protocol based on capacity-based secure access authentication, furthermore, provided reliable operation for data security and privacy in V2G networks. As mentioned above, none of them perform a systematic examination of the underlying cryptographic primitives and their functionalities in blockchains systems. This situation does not match with the fact that the blockchain is a crypto-intensive technology. Furthermore, a survey

* Corresponding author.
E-mail address: lijing@gzhu.edu.cn (J. Li).

https://doi.org/10.1016/j.jnca.2018.11.003
Received 17 June 2018; Received in revised form 12 October 2018; Accepted 9 November 2018
Available online 22 November 2018

Table 1
Summary of cryptographic primtives/algorithms in blockchains.

	Hashes						Signatures						Com/Acc		Proofs	
	SHA256	Ethash	SCrypt	X11	Equihash	RIPEMD160	ECDSA	EdDSA	Ring	One-Time	Borromean	Multi-signature	Commitment	Accumulator	ZK-SNARK	Bulletproofs
Bitcoin (Nakamoto, 2008)	✓					✓	✓					✓				✓
Ethereum (Ethereum)	✓	✓				✓	✓									
Dash (Dash)	✓			✓			✓					✓				
Litecoin (Litecoin)	✓		✓			✓	✓					✓				
Zcash (Ben-Sasson et al., 2014a)	✓				✓		✓			✓			✓	✓	✓	✓
Zcoin (Miers et al., 2013)	✓						✓					✓	✓	✓		
ZILLIQA (Zilliqa)		✓					EC-Schonrr					✓				
Monero (van Saberhagen, 2013)	✓	Keccak, blake256				✓		✓	✓	✓	✓		✓			✓
Ripple (Ripple)	✓					✓	✓					✓				
Nxt (Nxt)	✓		✓			✓	EC-KCDSA									
Blackcoin (Vasin, 2014)	✓		✓			✓	✓									
NEM (Nem, 2015)	Keccak256, Keccak512					✓		✓				✓				
Siacoin (Vorick and Champine, 2014)	blake2b							✓				✓				
Verge (Verge)	SCrypt, X17, blake2smyr-groestl, Lyra2rev2						✓					✓				
Qtum (Qtum)	✓	✓				✓	✓					✓				
BitConnect (Bitconnect, 2016)			✓				✓					✓	✓	✓		
Stratis (Khatwani, 2018)	✓		✓	X13			✓					✓				
Hshare (Hshare)	✓	X13,X14				✓	✓					✓				
Bytecoin (Bytecoin)	✓	Keccak, blake256				✓		✓	✓	✓	✓		✓			
Komodo (Komodo)	✓				✓		✓			✓			✓	✓	✓	
Dogecoin (Markus et al., 2013)	✓		✓			✓	✓					✓				
DigiByte (Digibyte)	SHA256, SCryptGroestl, Skein, Qubit						✓					✓				
RaiBlocks (LeMahieu, 2016)	blake2b						✓									
Ark (Thoorens et al., 2016)	✓						✓					✓				
MonaCoin (Monacoinproject, 2013)	✓	Lyra2rev2					✓					✓				
Byteball (Byteball)	✓						✓					✓				
Electroneum (van Saberhagen, 2013)	✓	Keccak, blake256				✓		✓	✓	✓	✓		✓			
Naivecoin (Naivecoin)	✓							✓								
RScoin (Danezis and Meiklejohn, 2016)	✓						✓									
IOTA (Iota)	Curl, Keccak384									✓		✓				

on the cryptographic primitives in blockchains would be helpful for cryptographers who are going to devote themselves to the blockchain research, and for financial engineers/managers who want to evaluate cryptographic solutions for blockchain-based projects. To this end, this paper will focus on the cryptographic primitives in blockchains from a comprehensive perspective, in terms of the usages, functionalities, and evolutions of these primitives. In particular, the highlights of this paper can be summarized as follows.

- We present an overview of cryptographic primitives/algorithms used in top-30 mainstream cryptocurrencies in Table 1, which can be regarded as a quick manual for identifying which kinds of cryptographic primitives/algorithms are involved in interested cryptocurrencies.
- We also classify cryptographic primitives in blockchains into two categories: primary and optional. The former category includes cryptographic hashes and standard digital signatures that are essential for ensuring the blockchain as a globe ledger with tamper-proof, public verifiability and achievable consensus. While the latter category, mainly used for enhancing the privacy and anonymity of blockchain-based transactions, covers some special signatures (such as ring signatures), commitments, accumulators, zero-knowledge proofs and so on. Furthermore, other cryptographic primitives, such as secret sharing and oblivious transfer, are also indirectly used in constructions of commitments and zero-knowledge proofs. To the best of our knowledge, the standard encryption has never been directly used in blockchains, although it can be used in constructions of ring signatures, or tailored into commitment protocols.
- Finally, we give comprehensive explorations on the cryptographic primitives, including their functionalities and usages in blockchains, and their evolutions.

We hope these explorations are beneficial to readers who want to identify proper references for further study on cryptocurrencies or blockchain-based applications.

ROADMAP. The rest of this paper is organized as follows. The usages of hash functions in blockchains are summarized in Section 2, and two standard signature algorithms (ECDSA and EdDSA) used in many cryptocurrencies are introduced in Section 3. Cryptographic primitives for enhancing the privacy and anonymity of blockchains, including special signatures, homomorphic commitment, cryptographic accumulator, and typical zero-knowledge proofs such as ZK-SNARKs and Bulletproofs, are explored in sections 4, 5, 6 and 7, respectively. Finally, the conclusion of this paper is given in Section 8, and some future research directions are also given in this section.

2. Hash function in blockchains

A cryptographic hash function is an algorithm that maps data of arbitrary size to a fixed size string. Two security requirements named one-wayness and collision-resistance are usually required for hash functions. The former ensures that the underlying hash function is not invertible, while the latter implies that it is not easy to find two inputs having the same hash value. For a hash function with n-bit length output, the complexities of breaking one-wayness and finding a collision are respectively bounded by $\mathcal{O}(2^n)$ brute force attack and $\mathcal{O}(2^{\frac{n}{2}})$ birth-

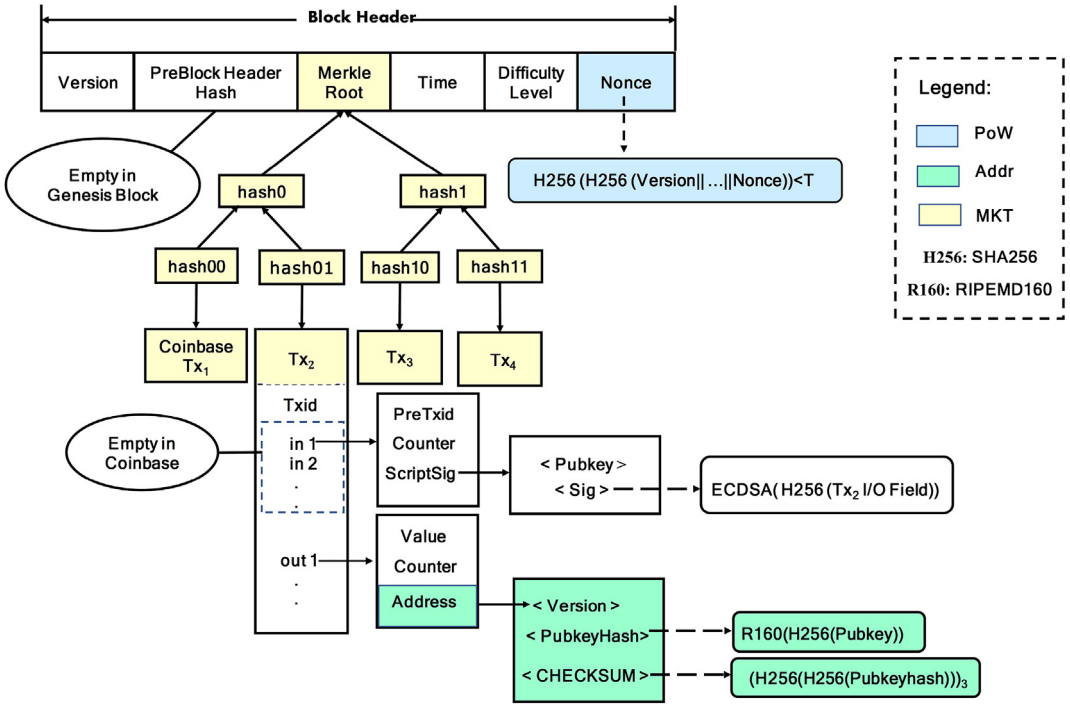


Fig. 1. The usages of hash function in Bitcoin.

day attack. Therefore, for ensuring at least 80-bit security, the output length of hash functions should be at least 160 bits.

The most popular hash function used in blockchains is SHA256, which is one of the algorithms from a family of cryptographic hash functions named SHA (Secure Hash Algorithms). SHA is a U.S. Federal Information Processing Standard, and most of the algorithms in this family, including SHA0 (published in 1993), SHA1 (published in 1995), SHA2 (published in 2001) are designed by the United States National Security Agency (NSA). While SHA3 (published in 2014) is original from Keccak proposed by (Bertoni et al., 2010), and only the padding method is modified by the National Institute of Standards and Technology (NIST). Keccak is the winner of the NIST hash function competition launched by the NIST in 2008 and ended in 2012. To satisfy the current security requirement, SHA2 and SHA3 are recommended for using in blockchains and cryptocurrencies.

2.1. The usages of hash function in blockchains

Roughly, we can divide usages of hash function in blockchains into the following six categories: proof-of-work (a.k.a., coin mining) (PoW), address generation (Addr for short), block generation (as a part of Merkle-tree paradigm, MKT for short), message digest in signatures (MDS), pseudorandom number generation (PNG), and bridge components (as that in the well-known Fiat-Shamir mechanism,[1] FSM for short). The last four usages are quite typical and popular even before the birth of blockchain, while the first two became hot recently due to the emergence of cryptocurrencies and blockchain.

The usages PoW, Addr and MKT in Bitcoin are depicted in Fig. 1. The core functionality of PoW is to enable a decentralized group without pre-established trust to agree on a consistent transaction history and prevent from double-spending attacks (Nakamoto, 2008). Although the idea of using the hash as a PoW tool firstly appeared in 1997 when Back (Back, 1997) put forward the concept of Hashcash to resist DDoS attacks, it became a popular topic about 10 years later when Bitcoin came to us. Given an input X, to find a nonce Y such that the first ℓ bits of the hash value $h(h(X, Y))$ are all zero. In Bitcoin, ℓ is adjusted, whenever 2016 new blocks are produced out, to keep the average interval of block generations being around 10 min. Specifically, the difficulty of finding such Y, d_i $(i = 1, 2, \cdots)$ is adjusted by the following formula

$$d_i = d_{i-1} \times \frac{T_{2015}}{1209600}, \tag{1}$$

where the constant 1209600 is the ideal total time (in seconds) for generating 2016 blocks, while T_{2015} is the entire real time (in seconds) for producing the previous 2015 blocks. Note that d_0 can be different for various ways to measure the initial difficulty. In engineering practice of Bitcoin, the difficulty value is not directly stored in blocks. Instead, each block stores a 32-bit packed number T for representing its actual hexadecimal target. Given T, the corresponding mining difficulty d can be derived by

$$d = A \times 256^{B-3}, \tag{2}$$

where $A = \mathrm{LSB}_{24}(T)$ and $B = \mathrm{MSB}_8(T)$, i.e. the least 24 significant bits of T and the most 8 significant bits of T, respectively.[2]

2.2. Contest between mining techniques and new hash algorithms

There exists an interesting contest between mining techniques and designs of new hash functions. The development of mining techniques is a thrilling thing for miners, while it may be a bad news for blockchain itself. In particular, advanced mining techniques enable some entity to have a higher mining speed than others, and if abused, the well-known 51% attack would happen with a higher probability.

During the past decade, we have witnessed that the mining speed has been increased from 100 MHash/s on Intel® CPUs to 2 GHash/s on

[1] On one hand, FSM can be viewed as a special case of PNG in the sense that FSM is often used for generating random challenges in interactive protocols. On the other hand, FSM is always modeled as a random oracle model in the security proof, while PNG is not the same case.

[2] See https://en.bitcoin.it/wiki/Difficulty and https://github.com/bitcoin/bitcoin/blob/master/src/pow.cpp#L49 for more details.

ECDSA specified as `secp256k1`:

KeyGen: $(\mathbb{E}, q, a, b, G, n, h; d, Q)$

$\mathbb{E}$: an elliptic curve $y^2 = x^3 + ax + b$ over $\mathbb{F}_q$

q: a prime $2^{256} - 2^{32} - 2^9 - 2^8 - 2^7 - 2^6 - 2^4 - 1$

a, b: $a = 0,\ b = 7$

G, n: a random base point in $\mathbb{E}$ with prime order n

h: a hash, instantiated with SHA1

Signing key: $d \xleftarrow{\$} [1, n-1]$

Verification key: $Q = dG \in \mathbb{E}$

Sign$(d; m)$: $\langle r, s\rangle \in \mathbb{F}_q^2$ where

r: the non-zero x-coordinator of point kG for some $k \xleftarrow{\$} [1, n-1]$

s: $s = k^{-1}(h(m) + d \cdot r) \bmod n$

Verify$(Q; r, s)$: $(r, s \stackrel{?}{\in} [1, n-1])$ and $(v \stackrel{?}{=} r)$, where

v: the x-coordinator of point $s^{-1}(h(m)G + rQ) \in \mathbb{E}$.

Fig. 2. The description of ECDSA.

AMD® GPUs to 25 GHash/s on FPGAs to 14 THash/s on ASICs nowadays. To fight the development of the mining techniques, many ASIC-resistant and memory-hard hash functions have been proposed, such as Ethash (Ethash), SCrypt (Percival, 2009; Boolberry, 2014), X11 (X11) and Equihash (Biryukov and Khovratovich, 2016).

- Ethash, original from Keccak256 and Keccak512 and used in Ethereum-based cryptocurrencies (Buterin, 2013), is considered as a so-called ASIC-resistant hash function. However, it has been reported on 15th Dec. 2017 that the highest average hash-rate is 140 THash/s (Etherscan, 2018).
- SCrypt, proposed by Percival and published by IETF as RFC 7914 and used in many cryptocurrencies such as Tenebrix, Fairbrix and Litecoin, is considered a so-called memory-hard hash function. It is claimed that SCrypt asymptotically requires more memory (say 128 KB) than the one used in Bitcoin. However, InnoSilicon has developed A4+ LTC Master miners for SCrypt with the hash-rate 620MHash/s (Innosilicon, 2017a).
- X11 is another memory-hard hash function proposed by Duffield, who sequentially combined 11 hash functions chosen from the SHA3 candidates to compose of X11. These hash functions include Blake, Grostl, JH, Keccak, Skein, ECHO, Luffa, BMW, CubeHash, SHAvite and SMID. X11 is now used in Darkcoin. However, InnoSilicon also claimed that they had developed ASIC miners for X11 with the hash-rate 32.5 GHash/s (Innosilicon, 2017b).
- Equihash, proposed in ZCash (Biryukov and Khovratovich, 2016), is a new memory-hard hash function specified by three parameters n, k and d. Executing Equihash is to find 2^k numbers $i_1, i_2, \ldots, i_{2^k}$ satisfying the following conditions:

$$i_j < 2^{n/(k+1)+1},\ j = 1, \ldots, 2^k, \tag{3a}$$

$$h(i_1) \oplus h(i_2) \cdots \oplus h(i_{2^k}) = 0, \tag{3b}$$

and

$$h(i_1 \parallel i_2 \parallel \cdots \parallel i_{2^k}) < 2^{512-d} \tag{3c}$$

where h is the Blake2b hash function (Aumasson et al., 2012). The best algorithm to solve the Equihash puzzle is Wagner's algorithm which requires $\mathcal{O}(2^{\frac{n}{k+1}})$ memory, and any memory reduction will increase the time complexity. The authors of (Biryukov and Khovratovich, 2016) claimed that their 700 MB-proof is 120 bytes long and can be found in 15 s on a single-thread laptop with a 2.1 GHz CPU. An adversary with 250 MB memory would pay 1000-fold in computations when the best tradeoff strategy is applied, while a memoryless adversary would require prohibitive 2^{75} hash function calls.

3. Digital signatures

Besides the hash function, the digital signature is another inevitable cryptographic primitive in blockchains. The concept of the digital signature was put forward by Diffie and Hellman in 1976 when they opened the gate of public key cryptography (Diffie and Hellman, 1976). As a basic primitive of cryptography, digital signature is used for ensuring the source authentication (Lin et al., 2018), source non-repudiation and integrity. The standard security of the digital signature is existential unforgeability against adaptively chosen messages attacks (EUF-CMA), which guarantees that the adversary cannot forge a valid signature on a new message, even if it can access the signing oracle that could provide the signing service.

ECDSA (Certicom-Research, 2000) and EdDSA (Bernstein et al., 2011) are the two digital signature schemes frequently used in blockchains. In principle, both of them are based on the hardness of the elliptic curve version of discrete logarithm problem. ECDSA works over a general elliptic curve and now is used in Bitcoin and Ethereum, while EdDSA works over a (twisted) Edward curve and now is used in Naivecoin and Monero. The Edward curve is a plane model of an elliptic curve and has better efficiency and security than a general elliptic curve. Thus, it has been already selected as the next elliptic curve generation of TLS by Internet Research Professional Working Group.

For the completeness, we describe ECDSA and EdDSA in Figs. 2 and 3, respectively. Meanwhile, we simplify them by omitting technical details on encoding/decoding aspects. Engineers with the purpose to

EdDSA specified as `ed25519`:

KeyGen: $(\mathbb{E}, q, d, G, l, h; a, A)$

$\mathbb{E}$: an Edward elliptic curve $-x^2 + y^2 = 1 + dx^2y^2$ over $\mathbb{F}_q$

q: a prime $2^{255} - 19$

d: $-\frac{121665}{121666} \in \mathbb{F}_q$

G, l: a random base point in $\mathbb{E}$ with prime order l

h: a hash, instantiated with SHA512

Signing key: $a \xleftarrow{\$} [1, l-1]$

Verification key: $A = aB \in \mathbb{E}$

Sign$(a; m)$: $\langle R = rB, s = r + a \cdot h(A \parallel R \parallel m) \rangle \in \mathbb{E} \times \mathbb{Z}_l$

where $r \xleftarrow{\$} [1, l-1]$.

Verify$(A; R, s)$: $sB \stackrel{?}{=} R + h(A \parallel R \parallel m)A$

Fig. 3. The description of EdDSA.

implement ECDSA and EdDSA are suggested to refer to (Certicom-Research, 2000) and (Bernstein et al., 2011), respectively.

4. Special signature primitives for blockchains

To enhance the privacy and anonymity of transactions, some advanced signature primitives such as ring signature and multi-signature are also widely applied in blockchains.

4.1. Ring signatures

Anonymity is always required in information systems (Shen et al., 2018), especially in the e-cash system. However, Bitcoin can only provide pseudonymity due to the linkability of transactions. Therefore, many new alternative cryptocurrencies have been proposed to address this problem. From a perspective of cryptography, there are many kinds of signatures for achieving anonymity, such as blind signature (Chaum, 1982), ring signature (Rivest et al., 2001), group signature (Chaum and van Heyst, 1991) and DC-nets (Chaum, 1988). However, only ring signature and its variants have been used in blockchains for anonymity.

The concept of ring signature was proposed in 2001 by Rivest, Shamir and Tauman (Rivest et al., 2001). One can use a ring signature scheme to sign messages on behalf of a group including himself/herself without revealing himself/herself, while he/she can compose this group without other group members' permission. Besides the existential unforgeability, the unconditional anonymity is another important security requirement for ring signature. This new property can be divided into two sub-properties: untraceability and unlinkability. The former means that one cannot identify the signer, while the latter says that no one can decide whether two signatures are generated by the same signer. The unconditional anonymity is a strong security notion that would be a double-edged sword: On one hand, it provides perfect privacy protection towards individual signing behavior. On the other hand, it could be abused for some illegal purpose such as wash trading. Therefore, some restrictions on anonymity should be taken into consideration. In fact, even ten years before the concept of ring signature, Chaum (Chaum and van Heyst, 1991) proposed the concept of group signature, which allows a group member to anonymously sign a message on behalf of the group, with the restriction that a designated group manager is able to identify the signer whenever it is necessary. One of the main differences between group signature and ring signature lies in that the ring structure is an ad hoc group that can be formed in an on-the-fly manner, while the group structure is formed under the control of the group manager. Furthermore, anyone who wants to adjoin the group has to at first perform a registration process — either online or offline.

In 2004, Liu et al. (Liu et al., 2004) proposed a linkable spontaneous anonymous *group* (LSAG) signature scheme, which is essentially a linkable *ring* signature considering the spontaneous group formation and no group manager (Sun et al., 2017). Recently, Liu et al.'s idea was adopted by Back (Back, 2015) in designing Ring-Coin with necessary improvements in efficiency. Along with another line, Fujisaki et al. (Fujisaki and Suzuki, 2007) in 2007 extended the concept of ring signature into the so-called traceable ring signature by adding an issue-related tag into the signature. In this case, anyone in the ring, pretending to be another person to sign the same message, would face the risk of revealing his/her identity immediately. This idea was adopted to prevent double-spending and now becomes the basis of CryptoNote (van Saberhagen, 2013) with a slight modification.

However, either CryptoNote or Ring-Coin suffers from the possible attack based on the observation and analysis of the amounts sent in a given transaction (Noether, 2015). To hide amounts for any transaction, Maxwell (Maxwell, 2017) proposed the concept of the confidential transaction by using homomorphic commitment protocol. Shortly afterward, Noether (Noether, 2015) offered a modification to the Monero protocol by coupling three techniques: Maxwell's confidential transaction, ring signature and multilayered linkable spontaneous, anonymous group signature (MLSAG). Noether's idea is now named as Ring Confidential Transactions for Monero (RingCT for short).

In 2017, Sun et al. (Sun et al., 2017) proposed a non-trivial upgraded version towards RingCT, named as RingCT 2.0. Besides the rigorous formalization of the syntax of RingCT and formal security models, RingCT 2.0 also applies some well-known cryptographic primitives, including Pedersen commitment, the accumulator with one-way domain and signature of knowledge, to obtain the significant storage and communication cost saving. More specifically, the signature size is reduced from $\mathcal{O}(nm)$ to $\mathcal{O}(m)$, where n and m are the numbers of groups and accounts in one group, respectively. In other words, the transaction size

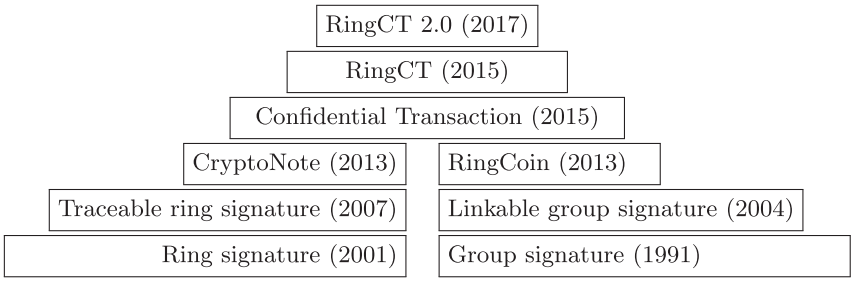


Fig. 4. The evolutionary process from ring signature to RingCT.

in RingCT 2.0 is independent of the number of groups in the ring, and this enables each block to process more transactions (Sun et al., 2017).

In summary, the evolutionary process from ring signature to ring confidential transactions is depicted in Fig. 4.

Remark 1. *The efficiency is always one of the important considerations to choose the particular cryptographic schemes. However, the existing linkable ring signature schemes with $\mathcal{O}(1)$ signature size, such as Tsang-Wei scheme (Tsang and Wei, 2005), Au-Chow scheme (Au et al., 2006), Chow-Susilo scheme (Chow et al., 2006), and Au-Liu scheme (Au et al., 2013), are not applied in blockchains. The main reason of this situation, as claimed in (Sun et al., 2017), is the "trusted setup" problem. In particular, it is required a trusted third party to distribute the system parameter, which is against the decentralized concept of blockchains. However, we believe that this situation would be improved gradually, especially Goyal and Goyal (Goyal and Goyal, 2017) showed that the "trusted setup" in cryptography can be achieved using blockchains.*

4.2. One-time (ring) signatures

Lamport in 1979 (Lamport, 1979) proposed the concept of one-time signature (OTS), where the signing key can be used *securely but only once*, and the signing key would be revealed if it is used twice or more. OTS is frequently used as a building block in constructions of encryptions and authenticated key agreements.

By combining the ideas of OTS and ring signature, Saberhagen (van Saberhagen, 2013) proposed a new signature scheme where the private key can be used only once for signing on behalf of a group. Suppose that Bob's public key is (A, B), and Alice wants to send a payment to Bob. Then, Alice can pick a random number $r \in \mathbb{F}_q$ and compute the transaction public key R and the destination key P as follows:

$$R = rG \quad \text{and} \quad P = \mathcal{H}_s(rA)G + B, \tag{4}$$

where $\mathcal{H}_s : \{0,1\}^* \to \mathbb{F}_q$ is a cryptographic hash function and G is the public base point of the elliptic curve $E(\mathbb{F}_q)$. Then, Bob can locate Alice's payment via checking every past transaction on the blockchain with his private key pairs (a, b)[3] to see if

$$P = \mathcal{H}_s(aR)G + B \tag{5}$$

holds. After locating Alice's payment, Bob can recover the corresponding one-time private key

$$x = \mathcal{H}_s(aR) + b \tag{6}$$

and spend this output at any time by signing a transaction with x.

Remark 2. *Unlike the traditional one-time signature, the one-time ring signature (van Saberhagen, 2013) only allows to link two valid signatures σ_1 and σ_2 if they are signed by the same private key. However, this linkability is enough for the double-spending-detection in blockchains.*

[3] That is, $A = aG$ and $B = bG$.

4.3. Borromean (ring) signatures

Another interesting primitive related to ring signature and blockchain is the so-called Borromean (ring) signature (BRS), proposed by Maxwell and Poelstra in 2015 (Maxwell and Poelstra, 2015). Poelstra (Poelstra, 2017) claimed that BRS is now used in Elements (Element, 2015), Liquid (Liquid) and Monero. Moreover, all of those projects are now being transited from the BRS-based range proofs to Bulletproofs (Bünz et al., 2017) that will be reviewed in Section 7.3.

In an abstract view, a ring signature is nothing than a signature that the signer knows one of secret keys for *a given group*, say

$$x_1 \vee x_2 \vee \cdots \vee x_n, \tag{7}$$

while a Borromean ring signature extends this idea to the scenario where the signer knows one of secrets for *each given group*, say

$$(x_1 \vee x_2 \vee \cdots) \wedge (y_1 \vee y_2 \vee \cdots) \wedge \cdots \wedge (z_1 \vee z_2 \vee \cdots). \tag{8}$$

Apparently, this idea gains the capability to express knowledge of any monotone boolean function of the signing keys (Maxwell and Poelstra, 2015).

Although the primitive of attribute-based signature (ABS) (Maji et al., 2011) can also realize the formula (8) by considering signing keys $x_1, x_2, \ldots, y_1, y_2, \ldots, z_1, z_2, \cdots$ as attributes and modelling the signing capability as a tree-like access structure corresponding to (8), there exists an essential difference between ABS and BRS. In particular, ABS focuses on who can generate a valid signature, while BRS focuses on how to aggregate multiple ring signatures anonymously. That is, the validations of all involved ring signatures in a BRS scheme are intertwined. If one of the ring signatures involved in the joint Borromean signature is invalid, then the entire signature is invalid, and you cannot tell which one is invalid (Poelstra, 2017). This is the very reason for the name Borromean ring signature. In topological, *Borromean rings* is a style of interlocking rings such that each ring goes through each other ring (Poelstra, 2017; Cromwell et al., 1998).

The construction of BRS scheme in reference (Maxwell and Poelstra, 2015) is based on an elegant combination of several efficient techniques, including Schnoor authentication (Schnorr, 1991), AOS ring signature (Abe et al., 2002), and the newly developed "half-chameleon hash" and "multiple-chameleon hash". Interested readers are suggested to refer (Maxwell and Poelstra, 2015) for more details.

4.4. Multi-signatures

The primitive of multi-signature allows a single signature to work as several ordinary signatures on the same message. One of the critical requirements of multi-signature is that the single signature has the same size as one regular signature. This primitive was introduced by (Itakura and Nakamura, 1983) in 1983 and has been studied over the past decades (Ohta and Okamoto, 1999; Okamoto, 1988; Boldyreva, 2002; Micali et al., 2001).

Very recently, ZILLIQA team (Zilliqa) proposed the next generation high throughput blockchain platform by using an EC-Schnorr

multi-signature protocol as one of its innovative ingredients. More specifically, the protocol in ZILLIQA consists of the following steps:

- The standard Schnorr signature scheme (Schnorr, 1991) is instantiated over the elliptic curve specified by `secp256k1` (Certicom-Research, 2000).
- The above EC-Schnorr signature scheme for a single user is extended to an EC-Schnorr multi-signature scheme for multiple users based on the idea in reference (Micali et al., 2001).
- The above EC-Schnorr multi-signature is tweaked for PBFT (practical Byzantine fault tolerance) settings, where the message is required to be properly signed by at least $\frac{2}{3}n+1$ nodes in the committee.

Remark 3. *Aggregate signature is a primitive tightly related to multi-signature and introduced by Boneh* et al. *at Eurocrypt 2003* (Boneh et al., 2003). *In such a signature, only one signature can be used for k signatures on k distinct messages from k different signers. It is easy to see that aggregate signature is a non-trivial generalization of multi-signature, and it is useful for saving storage and bandwidth.*

5. Homomorphic commitments

Commitment protocol is also very useful in blockchains. For instance, Pedersen commitment (Pedersen, 1991) and its vector version have already been used for building a blockchain-oriented range proof system – Bulletproof (Bootle et al., 2016; Bünz et al., 2017), and its elliptic curve version is also successfully used in Monero (Maxwell and Poelstra, 2015) and the newly proposed homomorphic mini-blockchain (França, 2015).

A cryptographic commitment protocol is a digital analog of a *sealed envelope* (Richelson, 2014), where two parties (Alice and Bob) play the following two-phase game. In the first phase, called commit phase, Alice sends Bob a commitment $\mathtt{Com}(m,r)$ for committing some valuable secret m in the sent envelop. In the second phase, called the open phase, Alice opens $\mathtt{Com}(m,r)$ to prove that Alice did not cheat Bob in the commit phase. $\mathtt{Com}(m,r)$ should satisfy two security requirements: *hiding and binding*. Informally, hiding ensures that Bob cannot tear the envelope for seeing m before the open phase,[4] while binding guarantees that Alice cannot alter what was sealed in the sent envelop after the commit phase. As we can see in the existing commitment schemes, these two requirements can be achieved computationally, statistically or perfectly (Richelson, 2014). It seems that the terminology "commitment" is from the remote coin-flipping protocol due to Blum in 1981; however, earlier to that, commitment via one-way hash functions had already been considered by Lamport as part of the original one-time one-bit signature scheme in 1979 (Damgård, 1998).

As a useful building block, cryptographic commitment manifests numerous important applications. For instance, the well-known commitment protocol, Pedersen commitment (Pedersen, 1991) and its variants (Derler et al., 2015) are used in blockchains. The Pedersen commitment and its vector version are depicted in Figs. 5 and 6, respectively. When $\mathbb{G}$ is instantiated as the additive group of a rational point of the elliptic curve $\mathbb{E}$ over a finite field $\mathbb{F}_q$, we have the elliptic curve Pedersen commitment as shown in Fig. 7.

It is easy to see that Pedersen (vector) commitments have the so-called additively homomorphic property in the sense that

$$\mathtt{Com}(x_1;r_1) \uplus \mathtt{Com}(x_2;r_2) = \mathtt{Com}(x_1+x_2;r_1+r_2) \tag{9}$$

[4] In fact, this analog merely captures the one-wayness of the hiding property. More precisely, hiding property says that Bob can learn *nothing* about m. Or equivalently, $\mathtt{Com}(m,r)$ should be indistinguishable from random values, regardless of m (Richelson, 2014).

Pedersen Commitment

$$\mathtt{Setup}: g, h \xleftarrow{\$} \mathbb{G}$$

$$\mathtt{Com}(x;r): z = g^x h^r \in \mathbb{G}$$

Fig. 5. The description of Pedersen commitment.

Pedersen Vector Commitment

$$\mathtt{Setup}: \vec{g} = (g_1, \cdots, g_n), h \xleftarrow{\$} \mathbb{G}$$

$$\mathtt{Com}(\vec{x};r): z = h^r \cdot \vec{g}^{\vec{x}} = h^r \prod_{i=1}^{n} g_i^{x_i} \in \mathbb{G}$$

Fig. 6. The description of vector version of Pedersen commitment.

where homomorphic combination operator $\uplus$ should be instantiated as the basic group operation over $\mathbb{G}$.

In Zerocoin, users can utilize Pedersen commitment to hide their coins. In particular, hiding serial number s and secret number r by using z, where $z = g^s h^r \in \mathbb{G}$. After that, to spend the committed coin, the user can generate a signature of knowledge about z to prove that he knows r and $z = g^s h^r$.

On the other hand, Monero uses EC Pedersen commitment to hide transaction amounts. Suppose that in a given transaction, one wants to hide an input amount b_{in} and two output amounts $b_{out,1}$ and $b_{out,2}$. Then, he can pick three blind factors a_{in}, $a_{out,1}$ and $a_{out,2}$ at random, and make an input commitment C_{in} and two output commitments $C_{out,1}$ and $C_{out,2}$ as follows:

$$C_{in} = a_{in}G + b_{in}H, \tag{10a}$$

$$C_{out,1} = a_{out,1}G + b_{out,1}H, \tag{10b}$$

$$C_{out,2} = a_{out,2}G + b_{out,2}H. \tag{10c}$$

To further prove that he indeed obeys the amount balance restriction $b_{in} = b_{out,1} + b_{out,2}$ in making the above commitments, he also needs to output a signature σ that is generated with knowing the temporary private key $t = a_{in} - a_{out,1} - a_{out,2}$, and publicly verifiable with only knowing the corresponding temporary public key

$$tG = C_{in} - C_{out,1} - C_{out,2}. \tag{11}$$

Recently, the concept of commitment has been extended to polynomial commitment (Kate et al., 2010) and functional commitment (Libert et al., 2016a). We believe that they can be applied in blockchain-based systems as useful tools. Here, we just provide a tailored description on the setup algorithm and the commitment algorithm of these two new cryptographic primitives. More details can be found in (Kate et al., 2010) and (Libert et al., 2016a). (see Fig. 8).

6. Accumulator

The accumulator is a one-way function that can provide membership proofs without revealing any individual member in the underlying set (Derler et al., 2015). It is a useful building block in many other cryptographic primitives used in blockchains, such as commitments and (ring)

ELLIPTIC CURVE PEDERSEN COMMITMENT

$\texttt{Setup}: G, H \xleftarrow{\$} \mathbb{E}(\mathbb{F}_q)$

$\texttt{Com}(x; r) : Z = xG + rH \in \mathbb{E}(\mathbb{F}_q)$

Addition: $\texttt{Com}(x_1; r_1) + \texttt{Com}(x_2; r_2) = \texttt{Com}(x_1 + x_2; r_1 + r_2)$

Scale multiplication: $\texttt{Com}(k \cdot x; k \cdot r) = k \cdot \texttt{Com}(x; r)$

Fig. 7. The description of Pedersen commitment over elliptic curve.

POLYNOMIAL COMMITMENT

$\texttt{Setup}(t)$: Bilinear map $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$

$(g, g^{\alpha}, \cdots, g^{\alpha^t}, h, h^{\alpha}, \cdots, h^{\alpha^t}) \xleftarrow{\$} \mathbb{G}^{2t+2}$

$\texttt{Com}\,(p(x); r(x)) : z = \left(\prod_{j=0}^{t}(g^{\alpha^j})^{p_j}\right) \cdot \left(\prod_{j=0}^{t}(h^{\alpha^j})^{r_j}\right)$

where $p(x) = \prod_{j=0}^{t} p_j x^j$, $r(x) = \prod_{j=0}^{t} r_j x^j$.

Fig. 8. The description of polynomial commitment.

signatures. For example, the linkable ring signature in RingCT 2.0 (Sun et al., 2017) is based on an accumulator scheme. Furthermore, the accumulator can also be directly applied in blockchains for implementing range proofs. For instance, a typical range proof in Zerocoin of

$$(c = c_1) \vee (c = c_2) \vee \cdots \vee (c = c_n) \tag{12}$$

is implemented by a member witness of the accumulator for the member set $\{c_1, c_2, \ldots, c_n\}$.

Generally speaking, there are three functionalities for the accumulator, i.e., membership witness, non-membership witness and dynamics. The functionality of membership witness allows a prover to efficiently generate the membership witness for any element in the set. While the functionality of non-membership witness further enables the prover to efficiently compute the non-membership witness for any element not in the set. The accumulator with property non-membership witness is called universal accumulator. The last functionality of dynamics empowers the prover to dynamically update the elements in the set and the corresponding witnesses.

Regarding the security requirements of the accumulator, we have one-wayness, indistinguishability, collision-resistance and undeniability. The first two security requirements are related to the information leak from witnesses, and the rest requirements are associated with the generation of witnesses. In particular, the one-wayness guarantees that the witness cannot reveal any information about the members in the set. The indistinguishability assures that anyone cannot tell a member x from set Q_0 or Q_1 with the witness $x \in Q_0 \cup Q_1$, where Q_0 and Q_1 are two disjoint sets. While the collision-resistance guarantees that anyone cannot generate a membership witness for a member not in the set, and the undeniability ensures that anyone cannot generate a membership witness and a non-membership witness for the same member (see Fig. 9).

Furthermore, there exist another two properties for the accumulator, i.e., trust-setup-free and trapdoor-free, which can be easily understood by their names. In Table 2, we summarize the existing accumulator schemes according to the above properties and requirements. The existing accumulator schemes can also be classified into three categories according to the underlying assumption, i.e., RSA-based, pairing-based and hash-based. Fig. 10 depicts a summary of techniques for building accumulators, and more details are given in the following subsections.

6.1. Accumulators based on RSA related assumptions

Due to the popularity of RSA cryptosystem, RSA related assumptions are used as the underlying complexity assumptions of many cryptographic primitives. Based on the trapdoor in the RSA problem, (Benaloh and de Mare, 1993) in 1993 proposed the first accumulator scheme that only satisfies one-wayness security requirement. However, the one-wayness is not sufficient for a hostile environment where an adversary can actively select cumulative values. To solve this problem, (Barić and Pfitzmann, 1997) suggested the collision-freeness as another security requirement. Furthermore, Benaloh and de Mare's scheme requires a trusted party to set up the RSA modulus. To remove this trust-setup process, (Sander, 1999) proposed a new accumulator scheme based on the strong RSA assumption but without using any trapdoor. Meanwhile, dynamics is usually required in many applications of the accumulator, while all the above accumulator schemes are static. To this end, (Camenisch and Lysyanskaya, 2002) put forward the concept of the dynamic accumulator, and a concrete dynamic accumulator scheme where the update cost is independent of the size of the underlying set. With the development of the accumulator, the non-membership witness was also considered as a desired requirement. The first accumulator scheme supporting non-membership witnesses was proposed by (Li et al., 2007). However, their proposal requires the trust-setup as the scheme in (Benaloh and de Mare, 1993). Later on, to remove the trust-setup process, (Lipmaa, 2012) proposed a new accumulator scheme called root accumulator based on the root assumption, while its efficiency is still comparable to Li et al.'s scheme. Furthermore, based on Li et al.'s scheme, (Mashatan and Vaudenay, 2013) proposed a fully dynamic universal accumulator scheme, which can generate a new witness for an added/deleted member.

6.2. Accumulators based on pairings

The first pairing-based accumulator scheme was proposed by (Nguyen, 2005), and it can be used to build the first identity-based ring signature scheme with constant signature size. However, it cannot provide the non-membership witness, and it requires the knowledge of the master secret key to update the accumulator. These two disadvantages could impede the use of the accumulator in the ring signature. For the first problem, (Damgård and Triandopoulos, 2008) proposed an improvement to support non-membership witness. Regarding the second problem, (Au et al., 2009) proposed a dynamic universal accumulator scheme by using the same technique used in (Li et al., 2007). However, the witness update in schemes mentioned above is not as efficient as expected. To this problem, (Camenisch et al., 2009) proposed an improved dynamic accumulator scheme. (Acar and Nguyen, 2011) presented another accumulator scheme with delegated non-membership proofs without the limitation on the number of accumulated elements in (Nguyen, 2005).

LINEAR FUNCTIONAL COMMITMENT

$\texttt{Setup}(n)$: Bilinear map $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_t$ with $|\mathbb{G}| = p_1p_2p_3$

Commit key $\{g^{\alpha^j}\}_{j=1}^{n}, \{U_j = u^{\alpha^j}\}_{j=1, j\neq n+1}^{2n}$ for some $g, u \in \mathbb{G}_{p_1}$

Trapdoor key $U_{n+1} = u^{\alpha^{n+1}}$

$\texttt{Com}(\vec{m}, r)$: $z = g^r \cdot \prod_{j=1}^{n}(g^{\alpha^j})^{m_j}$

Witness for $\langle \vec{x}, \vec{m} \rangle = y$: $W_y = \prod_{i=1}^{n}\left((u^{\alpha^{n-i+1}})^r \cdot \prod_{j=1}^{n}(u^{\alpha^{n+1+j-i}})^{m_j}\right)$

Fig. 9. The description of linear functional commitment.

Table 2
Features of different accumulators.

Contributors.	Functionalities[a]			Security Properties[b]				Building Features[c]	
	MW	NMW	DYN	OW	CF	UND	IND	TSF	TDF
Benaloh et al. (Benaloh and de Mare, 1993)	✓	–	–	✓	–	–	–	–	–
Nyberg et al. (Nyberg, 1996)	✓	–	–	✓	✓	–	–	✓	✓
Baric et al. (Barić and Pfitzmann, 1997)	✓	–	–	✓	✓	–	✓	–	–
Sander et al. (Sander, 1999)	✓	–	–	✓	–	–	–	✓	✓
Camenisch et al. (Camenisch and Lysyanskaya, 2002)	✓	–	✓	✓	✓	–	–	–	✓
Nguyen et al. (Nguyen, 2005)	✓	–	✓	✓	✓	–	–	–	✓
Li et al. (Li et al., 2007)	✓	✓	✓	✓	–	–	–	–	–
Damgrd et al. (Damgård and Triandopoulos, 2008)	✓	✓	✓	✓	✓	✓	–	–	–
Camacho et al. (Camacho et al., 2008)	✓	✓	✓	✓	✓	–	–	–	–
Camenisch et al. (Camenisch et al., 2009)	✓	–	✓	✓	–	–	–	✓	✓
Au et al. (Au et al., 2009)	✓	✓	✓	✓	–	✓	–	–	–
Mashatan et al. (Mashatan and Vaudenay, 2013)	✓	✓	✓	✓	–	–	–	–	–
Acar et al. (Acar and Nguyen, 2011)	✓	✓	✓	✓	✓	–	–	–	–
Lipmaa et al. (Lipmaa, 2012)	✓	✓	–	✓	✓	✓	–	✓	✓
Buldas et al. (Buldas et al., 2000; Buldas et al., 2002)	✓	✓	✓	✓	✓	✓	–	✓	✓
Boneh et al. (Boneh and Corrigan-Gibbs, 2014)	✓	–	–	✓	✓	–	–	–	–
Libert et al. (Libert et al., 2016b)	✓	✓	✓	✓	✓	–	–	–	✓

[a] MW = Membership Witness, NMW=Non-Membership Witness, DYN = Dynamic.
[b] OW=One-Wayness, CF=Collision-Free, UND=Undeniable, IND=Indistinguishable.
[c] TSF = Trust-Setup-Free, TDF = Trapdoor-Free.

6.3. Accumulators based on hash function

The first hash-based accumulator scheme is proposed by (Nyberg, 1996). It is easy to see that the resultant accumulator scheme is trapdoor-free and more efficient than one-way accumulator schemes based on RSA-related assumptions. In the certificate management (one of the applications of the accumulator), it is essential that no one can prove that a certificate is valid and invalid at the same time. This leads to the security requirement named undeniability. By using hash functions and hash-trees, (Buldas et al., 2000; Buldas et al., 2002) proposed several undeniable universal dynamic accumulator schemes but with a trusted party to update the accumulator. To solve the problem, (Camacho et al., 2008) proposed a strong universal accumulator scheme. Recently, the indistinguishability requirement of hash-based accumulator scheme has been obtained considerable attention. Hermann et al. (de Meer et al., 2012) showed that the scheme in (Nyberg, 1996) cannot satisfy the indistinguishability, while that in (Barić and Pfitzmann, 1997) does. In 2014, Boneh et al. (Boneh and Corrigan-Gibbs, 2014) proposed another hash-based accumulator scheme with a new merit that the involved hash function is *algebraically deduced* from a simple bivariate Zagier polynomial over an RSA modular space. Furthermore, (Derler et al., 2015) proposed a simple, light-weight generic transformation for adding indistinguishability to existing accumulators. Recently, (Libert et al., 2016b) described a new construction, which is based on the hash tree and supports the zero-knowledge argument of knowledge.

7. Zero-knowledge (range) proofs

A natural idea to protect the privacy and anonymity of the transaction is to make transactions unlinkable. However, e-cash systems need to verify whether the spender has the secret corresponding to the address the money comes from. Fortunately, the zero-knowledge proof is born to solve this dilemma.

7.1. From proofs to ZK-SNARKs

In computer science, a *proof* for an NP-complete language L is a protocol enabling a prover to convince a verifier that $x \in L$. Two basic security requirements *completeness and soundness* are usually desired in a proof. The former is similar to a sufficient condition. If $x \in L$, then this proof should be accepted after the execution of the protocol. The latter is similar to a necessary condition. If $x \notin L$, then the probability of this proof being accepted is negligible. Usually, these two basic security requirements can be achieved computationally, statistically or perfectly (Richelson, 2014). However, from a mathematical point of view, a computational soundness proof is not a proof, but an *argument* (Brassard et al., 1988). It is because that the proof might become false with the breakthrough of computational technology or the invention of new algorithms. Furthermore, a proof (or argument) is said

- *zero-knowledge* if after accepting the proof, the verifier cannot learn anything beyond $x \in L$ (Goldwasser et al., 1985);

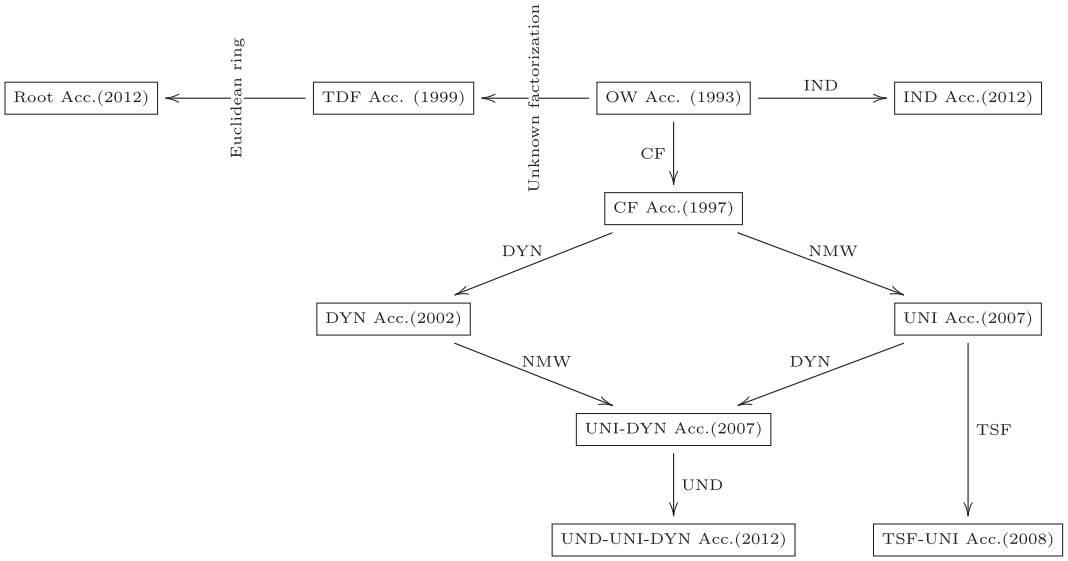


Fig. 10. The evolution of the underlying techniques in accumulators.

- *succinct* if both the length of the proof and the verification time are bounded by polylog functions with respect to the size of the circuit C representing $x \in L$ (Kilian, 1992; Micali, 1994);
- *proof of knowledge (POK for short)* or *argument of knowledge (AOK for short)* if given an acceptable proof/argument, anyone can extract a witness for supporting this proof/argument in polynomial time (Blum et al., 1988; Naor and Yung, 1990; Blum et al., 1991);
- *non-interactive* if there is only one pass communication from the prover to the verifier.

A succinct non-interactive argument is usually called *SNARG* for short (Kilian, 1992; Micali, 1994), and a SNARG with the property of AOK is called *SNARK*. As aforementioned in Section 2.1, one can derive a non-interactive proof from an interactive one by using the well-known Fiat-Shamir mechanism (Fiat and Shamir, 1986). For instance, Micali (Micali, 2000) introduced a method to construct a publicly-verifiable SNARG for NP in the random oracle model based on Kilian's protocol (Barak and Goldreich, 2002). Recently, several researches also showed how to construct SNARK via extractable collision-resistant hash functions (Bitansky et al., 2012; Damgård et al., 2012; Goldwasser et al., 2011) or fully-homomorphic encryption with an extractable homomorphism property (Bitansky and Chiesa, 2012).

A SNARK with the property of zero-knowledge is called *zk-SNARK*. Various techniques on building zk-SNARKs have been proposed in the recent years (Groth, 2010; Lipmaa, 2012; Bitansky et al., 2013a; Gennaro et al., 2013; Parno et al., 2013; Lipmaa, 2013; Ben-Sasson et al., 2014b; Ben-Sasson et al., 2014c; Braun et al., 2013; Bitansky et al., 2012; Bitansky et al., 2013b; Parno et al., 2016; Ben-Sasson et al., 2013; Groth, 2016; Bowe et al., 2017a; Abdolmaleki et al., 2017). We would like to summarize them into five levels as shown in Fig. 11. From the top level to the middle level, the proof targets, described by high level languages, are transformed into quadratic arithmetic programs (QAP), quadratic spanning programs (QSP) and square spanning programs (SSP), respectively, while from the bottom level to the middle level, the satisfiability problems, encoding in QAP, QSP or SSP, are verified by the techniques such as PCP, homomorphic encryption (Xu et al., 2018), pairing, etc, based on the hardness assumptions such as q-PKE, q-PDH, q-SDH, or q-TSDH. In brief, the middle level, QAP, QSP and SSP are the core for implementing zk-SNARKs.

A more detailed exploration is given below:

- The first transformation is from proof targets to the circuit satisfiability. The critical component in this transformation is the circuit generator (Ben-Sasson et al., 2014c) that can translate the correctness of suitably-bounded program executions into arithmetic circuits or boolean circuits, where suitably-bounded program executions stand for proof targets.
- The second transformation is to convert the circuit satisfiability to satisfiability encoding. In particular, converting arithmetic circuits and boolean circuits into a comparably sized *Quadratic Arithmetic Programs* (QAPs) (Gennaro et al., 2013), and *Quadratic Span Programs* (QSPs) (Gennaro et al., 2013) or *Square Span Programs* (SSPs) (Danezis et al., 2014), respectively.
- With satisfiability encoding and verification protocols, we can have zk-SNARKs. In most of existing zk-SNARKs, bilinear maps (Boneh and Franklin, 2001) play a critical role to allow a prover to prove the knowledge of secret inputs without revealing the inputs and to allow the private verifiability of the circuit satisfiability. The probabilistically checkable proof (PCP) (Babai et al.,1991; Feige et al., 1991; Arora et al., 1998) and homomorphic encryption are another two crucial tools for proof verifications in zk-SNARKs. Although PCPs and homomorphic encryptions are usually time-consuming, many zk-SNARKs are based on them (Bitansky and Chiesa, 2012; Goldwasser et al., 2011; Bitansky et al., 2012; Damgård et al., 2012; Bitansky et al., 2013a; Ben-Sasson et al., 2013; Ben-Sasson et al., 2017c). For instance, the schemes in (Ben-Sasson et al., 2013) and (Bitansky et al., 2013a) used a linear PCP and linear homomorphic encryption to verify satisfiability of circuits. Bitansky et al. (Bitansky and Chiesa, 2012) utilized the fully-homomorphic encryption to commit to a vector of functions succinctly.
- At last, verification protocols are usually related to some hardness assumptions, including q-power knowledge of exponent assumption (q-PKE) (Damgård, 1991), q-power Diffie-Hellman assumption (q-PDH) (Groth, 2010), q-strong Diffie-Hellman assumption (q-SDH) (Groth, 2010), and $q(\lambda)$ target group strong Diffie-Hellman assumption (q-TSDH) (Boneh and Boyen, 2008).

There are many efficient zk-SNARKs based on QAPs (Gennaro et al., 2013; Bitansky et al., 2013a; Parno et al., 2013; Ben-Sasson et al., 2013; Ben-Sasson et al., 2014c; Parno et al., 2016; Bowe et al., 2017a; Abdolmaleki et al., 2017) or QSPs (Lipmaa, 2013; Gennaro et al., 2013); however, all of them suffer from the so-call trusted setup problem. Furthermore, their implementations are lack of scalability. In particular, they require computationally and lengthy intensive protocols (Bowe et al., 2017b) for generating highly complex common reference strings (CRS).

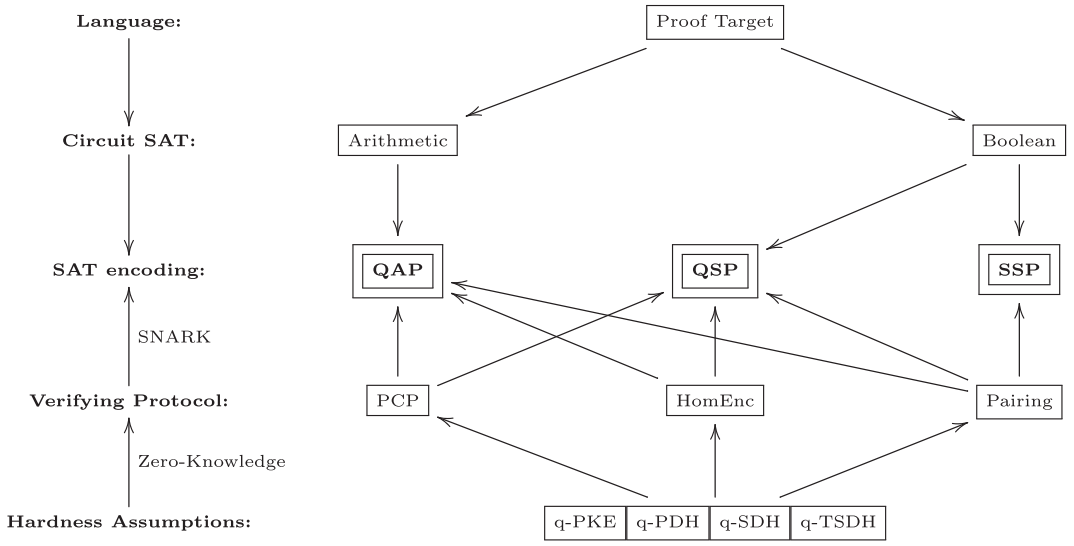


Fig. 11. The technical architectures of zk-SNARK.

Most recently, several new signs of progress on zk-SNARKs have been made. Ben-Sasson et al. (Ben-Sasson et al., 2017a; Ben-Sasson et al., 2017b) presented a proof system called *Scalable Computational Integrity* (SCI). With SCI, a proof system only requires a simple setup and only relies on collision-resistant hash functions. However, the proof system is not zero-knowledge and so efficient as that in (Ben-Sasson et al., 2013; Bootle et al., 2016). In the subsequent work, Ben-Sasson (Ben-Sasson et al., 2017c) presented a scalable, transparent argument-of-knowledge (STARK for short) that is zero-knowledge and more efficient than SCI. Additionally, Bootle et al. (Bootle et al., 2017) presented an argument for arithmetic circuit satisfiability based on collision-resistant hash functions. In particular, the cost of the prover scales linearly in the arithmetic circuit size, and the verifier is slightly sub-linear. Hence, it is more efficient than the schemes in (Bootle et al., 2016; Ben-Sasson et al., 2017a). Besides, Veeningen Meilof (Veeningen, 2017) proposed an *adaptive* zk-SNARKs, where multiple different computations can be performed on the committed data.

7.2. Zk-SNARKs in Zerocash

Zerocoin proposed by Miers et al. (Miers et al., 2013) aims at providing Bitcoin with anonymity by breaking the trace of coins. However, the resultant e-cash cannot support full-edged anonymous payments due to the following reasons. First, Zerocoin uses coins of a fixed denomination. Second, one has to transfer anonymous coins into non-anonymous coins before payment. Third, it does not hide the amount or other metadata in transactions. To address the above issues, Ben-Sasson et al. (Ben-Sasson et al., 2014a) proposed Zerocash. Specifically, Zerocash provides user anonymity and privacy of transaction data with anonymous coins. Moreover, Zerocash significantly reduces the size of transactions with one coin to less than one KB and the verification time of a transaction to less than 6 ms.

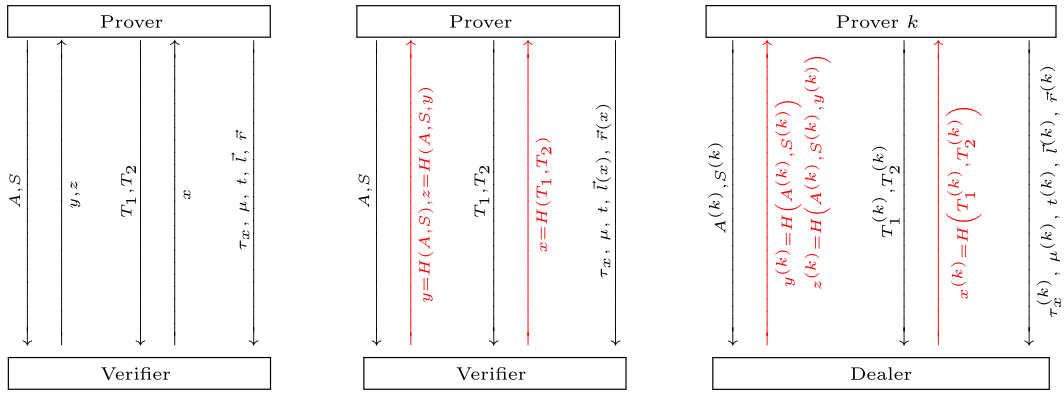
As a fact, Zerocash captures the functional requirements and security and privacy goals of a full-fledged decentralized e-cash system via zk-SNARKs (Ben-Sasson et al., 2014c). In Zerocash, transactions are divided into three types: basecoin transaction, mint transaction, and pour transaction. The key points of using zk-SNARKs in Zerocash for anonymity enhancement can be roughly summarized as follows:

- Basecoin transaction (without anonymity protection): The basecoin transaction is as the same as what in Bitcoin and thus user addresses and transaction amounts are public.
- Mint transaction (with committed coin values and public addresses): A mint transaction in Zerocash is actually a tuple $(cm, v, ^*)$, where cm is a coin commitment, v is a coin value, and * is used to record other necessary information. When a mint transaction is appended in the ledger, a certain number of coins are committed at the blockchain. Here, the key point is that the generator of the mint transaction can pay his committed values to others without revealing his address and the transferred value.
- Pour transaction (with enhanced anonymity protection): Given a mint transaction or a pour transaction, any user can generate a new pour transaction. Formally, a pour transaction is a tuple $(rt, sn_1, sn_2, cm_1, cm_2, \pi, ^*)$, where rt is a root of a Merkle tree of coin commitments, sn_1 and sn_2 are two coin serial numbers, cm_1 and cm_2 are two new coin commitments, π is a proof of zk-SNARK on secret inputs, and * is used to record other necessary information. When a pour transaction is appended to the blockchain, a certain number of coins are transferred from some users to others without revealing users' addresses and amounts of transferred coins.

In Zerocash protocol, the secret inputs of π are more complex. Here, we would like to give a simplified version for describing how to use secret inputs in Zerocash. For instance, the payer's address is $\{a_{pk}^{old}, a_{sk}^{old}\}$, where a_{pk}^{old} is public and a_{sk}^{old} is secret. Similarly, the payee has an address like $\{a_{pk}^{new}, a_{sk}^{new}\}$. The transferred coin value v and payer's secret address a_{sk}^{old} are hidden in the serial number sn, and the payee's public address a_{pk}^{new} is hidden in the commitment cm. In particular, $a_{pk}^{old} = \mathrm{PRF}(addr \| a_{sk}^{old} \| 0)$, $a_{pk}^{new} = \mathrm{PRF}(addr \| a_{sk}^{new} \| 0)$, $sn = \mathrm{PRF}(sn \| a_{sk}^{old} \| \rho^{old})$, $k = \mathtt{Com}(r \| a_{pk}^{new} \| \rho^{new})$ and $cm = \mathtt{Com}(s \| v \| k)$, where $\mathtt{Com}$ and PRF are instantiated with the hash function SHA256, while a_{sk}^{old}, a_{pk}^{new}, ρ^{old}, ρ^{new}, r, s and v are inputs of the hash functions. Therefore, when a pour transaction only contains the commitment the sn and the cm, it does not reveal v, a_{sk}^{old} and a_{pk}^{new}. With zk-SNARK, payer generates a proof π to prove that he/she knows correct a_{sk}^{old}, a_{pk}^{new}, ρ^{old}, ρ^{new}, r, s and v.

7.3. Bulletproofs: zk-Range Proofs and Aggregations

Recall equations (10) in Section 5, it is possible for a malicious user to divide a coin with face value 1 into two coins with face value 10 and face value −9, respectively. This will result in a serious confusion, and the corresponding system would lose the faith of honest users. Fortunately, this problem can be solved by the technique named range proof of value. For example, the BRS (one instance of range proof of value)



(a) Single Range Proof (b) Interactivity Removing (c) Aggregating

Fig. 12. Range Proofs and Aggregations.

is used in Menero and Elements to prove that the committed value is positive and in a range (See Section 4.3). Recently, (Bünz et al., 2017) proposed a more efficient scheme called Bulletproofs by aggregating multiple range proofs.

Bulletproofs is a non-interactive and aggregatable inner-product range proof protocol that allows the provers to prove that several committed values lie in given ranges with a combined and every short proof. Actually, Bulletproofs is original from the inner-product argument (IPA) protocol proposed by (Bootle et al., 2016), while (Bünz et al., 2017) improved Bootle et al.'s scheme by halving the size of the base vector for the commitment. In particular, they transformed an n-dimensional vector to a 1-dimensional vector by $\log_2 n$ times iterations, thus the communication complexity of the original inner-product argument is reduced remarkably.

By using the improved inner-product argument protocol, (Bünz et al., 2017) present an inner-product range proof protocol. In such a protocol, the prover aims to prove that v is a number ranging in $[0, 2^n - 1]$ by using a Pedersen commitment $V = \mathrm{Com}(v;\gamma) = g^v h^\gamma$ with a blind factor γ. Specifically, assume v's binary representation is $v = \sum_{i=1}^{n} a_i \cdot 2^{i-1}$ with $a_i \in \{0,1\}$ $(i = 1,\ldots,n)$. The prover sends two binding commitments $A = h^\alpha \vec{g}^{\vec{a}_L} \vec{h}^{\vec{a}_R}$ and $S = h^\rho \vec{g}^{\vec{s}_L} \vec{h}^{\vec{s}_R}$ to the verifier, where *binding* means that the prover cannot modify A and S after they are submitted, whereas $\vec{a}_L = (a_1,\ldots,a_n) \in \{0,1\}^n$ is specified by v's binary representation, but $\vec{a}_R = (a_1 - 1,\ldots,a_n - 1)$ takes value in $\mathbb{Z}_p^n$, g and h are two generators of $\mathbb{G}$, $\vec{g}$ and $\vec{h}$ are the part of system parameters chosen from $\mathbb{G}^n$, α and ρ are two blind factors chosen randomly from $\mathbb{Z}_p$, and $\vec{s}_L$ and $\vec{s}_R$ are chosen randomly from $\mathbb{Z}_p^n$. After receiving A and S from the prover, the verifier chooses y and z randomly from $\mathbb{Z}_p$ and sends them to the prover as the challenges. To response the challenges, the prover firstly constructs two degree 1 vector-polynomials $\vec{l}(X) = \vec{a}_L - z \cdot \vec{1}^n + \vec{s}_L \cdot X \in \mathbb{Z}_p^n[X]$ and $\vec{r}(X) = \vec{y}^n \circ (\vec{a}_R + z \cdot \vec{1}^n + \vec{s}_R \cdot X) + z^2 \cdot \vec{2}^n \in \mathbb{Z}_p^n[X]$, where $\vec{1}^n$ is the n-dimensional vector with $(1,\ldots,1)$, $\vec{2}^n = (2^0, 2^1, \ldots, 2^{n-1})$, $\vec{y}^n = (y^0, y^1, \cdots, y^{n-1})$, and the operator $\circ$ denotes the component-wise product, i.e. $(a_1,\ldots,a_n) \circ (b_1,\ldots,b_n) = (a_1 \cdot b_1,\ldots,a_n \cdot b_n)$. Secondly, the prover computes an inner-product polynomial $t(X) = \langle \vec{l}(X), \vec{r}(X) \rangle = t_0 + t_1 X + t_2 X^2 \in \mathbb{Z}_p[X]$, and then sends the commitments $T_i = \mathrm{Com}(t_i;\tau_i) = g^{t_i} h^{\tau_i}$ $(i = 1, 2)$ to the verifier. Upon receiving T_i, $(i = 1, 2)$, the verifier chooses a new challenge $x \in \mathbb{Z}_p$ and sends it to the prover. The prover can response the challenge by replying the values of $\tau_x, \mu, t(x), \vec{l}(x), \vec{r}(x)$ to the verifier, where $\tau_x = \tau_1 x + \tau_2 x^2 + z^2 \gamma$ and $\mu = \alpha + \rho x$, while $\gamma \in \mathbb{Z}_p$ is just the blind factor specified at the beginning of the range proof. The verifier can check v is a number in $[0, 2^n - 1]$ by taking the advantage of the fact that the constant term t_0 of $t(X)$ is independent of $\vec{a}_L$ if and only if $\vec{a}_L$ indeed contains the bits of v (See (Bünz et al., 2017) for detail). Note that the vectors $\vec{l}(x)$ and $\vec{r}(x)$ can be replaced with the aforementioned inner-product argument, therefore the corresponding communication cost is reduced from $2n$ to $2\lceil \log_2 n \rceil$. A high-level description of the above process is given in Fig. 12(a). Finally, the challenge-response process in the range proof can be transformed into a non-interactive process by using a hash function. More specifically, the transmission of challenges y, z and x can be saved, since the prover can generate them by himself/herself according to $y = H(A,S)$, $z = H(A,S,y)$, and $x = H(T_1, T_2)$. This process is shown in red color in Fig. 12(b). Note that by doing so, the prover in practice can send all his/her proof transcripts to the verifier in a single pass.

With the help of a simple multi-party computation (MPC) protocol, multiple above inner-product range proofs can be aggregated into the Bulletproofs. For easy understanding, we assume that there exist m provers persuading the verifier that their committed values lie in given ranges. In this protocol, each prover $P_k (i = 1,\ldots,m)$ generates $(A^{(k)}, S^{(k)}, T_1^{(k)}, T_2^{(k)}, \tau_x^{(k)}, \mu^{(k)}, t^{(k)}, \vec{l}(x)^{(k)}, \vec{r}(x)^{(k)})$ as in the single range proof, and sends them to a dealer (which could be one of the provers). After receiving all the proofs from the provers, the dealer can generate the aggregated proof as follows.

$$A = \prod_{k=1}^{m} A^{(k)},\ S = \prod_{k=1}^{m} S^{(k)},\ T_1 = \prod_{k=1}^{m} T_1^{(k)},\ T_2 = \prod_{k=1}^{m} T_2^{(k)}, \tag{13}$$

$$\tau_x = \sum_{k=1}^{m} \tau_x^{(k)},\ \mu = \sum_{k=1}^{m} \mu^{(k)}, t = \sum_{k=1}^{m} t^{(k)},\ \vec{l}(x) = \coprod_{k=1}^{m} \vec{l}(x)^{(k)},\ \vec{r}(x) = \coprod_{k=1}^{m} \vec{r}(x)^{(k)}, \tag{14}$$

where $\coprod$ denotes the interleaved concatenation of multiple vectors. Specifically, if

$$\vec{g}^{(1)} = (g_1^{(1)}, g_2^{(1)}, \ldots,\ g_n^{(1)}),\ \cdots,\ \vec{g}^{(m)} = (g_1^{(m)}, g_2^{(m)}, \ldots, g_n^{(m)}), \tag{15}$$

we have

$$\coprod_{k=1}^{m} \vec{g}^{(k)} = (g_1^{(1)}, g_1^{(2)}, \cdots, g_1^{(m)}, g_2^{(1)}, g_2^{(2)}, \cdots, g_2^{(m)}, \cdots, g_n^{(1)}, g_n^{(2)}, \cdots, g_n^{(m)}). \tag{16}$$

Similar with the single inner-product range proof, $\vec{l}(x)$ and $\vec{r}(x)$ can also be replaced with the inner-product argument, and the resultant length is reduced from $2mn$ to $2\lceil \log_2 mn \rceil$. A high-level description of the above process is given in Fig. 12(c).

Table 3
Cryptographic Primitives vs. Privacy Concerns of Blockchains.

Primitives	Transaction Privacy			Efficiency
	Payer's Privacy	Payee's Privacy	Content Privacy	
Hash Function	○	◐	○	high
Standard Signature	○	○	○	medium
Ring Signature	●	○	○	low
One-Time Signature	○	●	○	medium
Borromean Signature	○	○	◐	low
Multi-signature	○	○	○	low
Commitment	○	○	●	medium
Accumulator	●	○	○	medium
ZK (Range) Proof	●	●	●	low

8. Conclusion

As a crypto-intensive thing, the blockchain is considered as one of the most exciting inventions in the information and communication technology community during the past decade. A comprehensive study on the underlying cryptographic primitives in blockchains would be helpful for a deep understanding of the security and privacy of blockchain-based systems. By studying the top-30 mainstream cryptocurrencies, we have classified the cryptographic primitives into two categories. For each primitive, we have also explained its functionality, usages and evolution. For easy guidance, we also give a brief evaluation on the extent for solving the related privacy issues in Table 3, where the solid circle means that the corresponding primitives provide good protection to the associated privacy issues, the empty circle indicates irrelevant, while the half-solid merely means partial protection. Note that a general evaluation on the efficiency of these cryptographic primitives is also listed in the rightmost column of Table 3.

The research of the blockchain and its applications is still in its infant stage, and many problems, especially the ones related to cryptography, remain unsolved. Some of them are as follows.

- The compatibility of lightweight cryptographic algorithms in blockchains: This is essential for bridging the technologies of the Internet of Things (IoT) and blockchains. Currently, except IOTA, no existing cryptocurrencies are based on lightweight cryptographic algorithms. This situation leaves a lot of rooms for future research.
- More efficient ways to resist unexpected computational power accumulation: Breakthroughs in computing capacity are usually expected; however, the current quick progress on hashrates brings some troubles to blockchain-based systems. People even began to use strange ways to overcome these new advances. For example, the so-called X11 and X17 are constructed by sequentially combining several existing hash functions, which is meaningless from the cryptographer's point of view. Indeed, this kind of PoW mechanism is merely a waste of electric power, without any positive outputs.
- Solutions for the paradox between the limited life-span of cryptographic algorithms and the claimed everlasting and tamper-proof property of blockchains. All the current cryptographic algorithms used in blockchains are computationally secure, which means it is possible that the underlying algorithms would become insecure in the future. Blockchains should have the ability to deal with this situation.

Acknowledgements

We thank Dr. Man Ho AU and Andrew Poelstra for the insightful discussion during the preparation of this paper. This work was supported by the National Key Research and Development Program of China (No. 2016YFB0800602), the National Natural Science Foundation of China (NSFC) (Nos. 61502048 and 61632012), and the Shandong Provincial Key Research and Development Program of China (2018CXGC0701).

References

Abdolmaleki, B., Baghery, K., Lipmaa, H., Zajac, M., 2017. A subversion-resistant SNARK. In: Advances in Cryptology - ASIACRYPT 2017 - 23rd International Conference on the Theory and Applications of Cryptology and Information Security, Hong Kong, China, December 3-7, 2017, Proceedings, Part III, Vol. 10626 of Lecture Notes in Computer Science. Springer, pp. 3–33.

Abe, M., Ohkubo, M., Suzuki, K., 2002. 1-out-of-n signatures from a variety of keys. In: Advances in Cryptology - ASIACRYPT 2002, 8th International Conference on the Theory and Application of Cryptology and Information Security, Queenstown, New Zealand, December 1-5, 2002, Proceedings, Vol. 2501 of Lecture Notes in Computer Science. Springer, pp. 415–432.

Acar, T., Nguyen, L., 2011. Revocation for delegatable anonymous credentials. In: Public Key Cryptography - PKC 2011 - 14th International Conference on Practice and Theory in Public Key Cryptography, Taormina, Italy, March 6-9, 2011. Proceedings, pp. 423–440.

Androulaki, E., Karame, G., Roeschlin, M., Scherer, T., Capkun, S., 2013. Evaluating user privacy in bitcoin. In: Financial Cryptography and Data Security -17th International Conference, FC 2013, Okinawa, Japan, April 1-5, 2013, Revised Selected Papers, Vol. 7859 of Lecture Notes in Computer Science. Springer, pp. 34–51.

Arora, S., Lund, C., Motwani, R., Sudan, M., Szegedy, M., 1998. Proof verification and the hardness of approximation problems. J. ACM 45 (3), 501–555.

Au, M.H., Chow, S.S.M., Susilo, W., Tsang, P.P., 2006. Short linkable ring signatures revisited. In: Public Key Infrastructure, Third European PKI Workshop: Theory and Practice, EuroPKI 2006, Turin, Italy, June 19-20, 2006, Proceedings, Vol. 4043 of Lecture Notes in Computer Science. Springer, pp. 101–115.

Au, M.H., Tsang, P.P., Susilo, W., Mu, Y., 2009. Dynamic universal accumulators for DDH groups and their application to attribute-based anonymous credential systems. In: Topics in Cryptology - CT-RSA 2009, the Cryptographers' Track at the RSA Conference 2009, San Francisco, CA, USA, April 20-24, 2009. Proceedings, Vol. 5473 of Lecture Notes in Computer Science. Springer, pp. 295–308.

Au, M.H., Liu, J.K., Susilo, W., Yuen, T.H., 2013. Secure id-based linkable and revocable-iff-linked ring signature with constant-size construction. Theor. Comput. Sci. 469, 1–14.

Aumasson, J.-P., Neves, S., Wilcox-OHearn, Z., Winnerlein, C., 2012.

Babai, L., Fortnow, L., Levin, L. A., Szegedy, M., 1991. Checking computations in polylogarithmic time. http://lance.fortnow.com/papers/files/check.pdf.

Back, A., 1997. Hashcash. http://www.hashcash.org/.

Back, A., 2015. Bitcoins with Homomorphic Value (Validatable but Encrypted). https://bitsharestalk.org/index.php/topic,16797.msg214814.html#msg214814.

Barak, B., Goldreich, O., 2002. Universal arguments and their applications. In: Proceedings of the 17th Annual IEEE Conference on Computational Complexity, Montréal, Québec, Canada, May 21-24, 2002. IEEE Computer Society, pp. 194–203.

Barić, N., Pfitzmann, B., 1997. Collision-free accumulators and fail-stop signature schemes without trees. In: Advances in Cryptology-EUROCRYPT '97, International Conference on the Theory and Application of Cryptographic Techniques, Konstanz, Germany, May 11-15, 1997, Proceeding, Vol. 1233 of Lecture Notes in Computer Science. Springer, pp. 480–494.

Ben-Sasson, E., Chiesa, A., Genkin, D., Tromer, E., Virza, M., 2013. Snarks for C: verifying program executions succinctly and in zero knowledge. In: Advances in Cryptology - CRYPTO 2013 - 33rd Annual Cryptology Conference, Santa Barbara, CA, USA, August 18-22, 2013. Proceedings, Part II, Vol. 8043 of Lecture Notes in Computer Science. Springer, pp. 90–108.

Ben-Sasson, E., Chiesa, A., Garman, C., Green, M., Miers, I., Tromer, E., Virza, M., 2014a. Zerocash: decentralized anonymous payments from bitcoin. In: 2014 IEEE Symposium on Security and Privacy, SP 2014, Berkeley, CA, USA, May 18-21, 2014. IEEE Computer Society, pp. 459–474.

Ben-Sasson, E., Chiesa, A., Tromer, E., Virza, M., 2014b. Scalable zero knowledge via cycles of elliptic curves. Lect. Notes Comput. Sci. 8617, 1–59.

Ben-Sasson, E., Chiesa, A., Tromer, E., Virza, M., 2014c. Succinct non-interactive zero knowledge for a von neumann architecture. In: Proceedings of the 23rd USENIX Security Symposium. USENIX Association, San Diego, CA, USA, pp. 781–796. August 20-22, 2014.

Ben-Sasson, E., Chiesa, A., Gabizon, A., Riabzev, M., Spooner, N., 2017a. Interactive oracle proofs with constant rate and query complexity. In: 44th International Colloquium on Automata, Languages, and Programming, ICALP 2017, July 10-14, 2017, Warsaw, Poland, Vol. 80 of LIPIcs, Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, pp. 40:1–40:15.

Ben-Sasson, E., Bentov, I., Chiesa, A., Gabizon, A., Genkin, D., Hamilis, M., Pergament, E., Riabzev, M., Silberstein, M., Tromer, E., Virza, M., 2017b. Computational integrity with a public random string from quasi-linear pcps. In: Advances in Cryptology - EUROCRYPT 2017 - 36th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Paris, France, April 30 - May 4, 2017, Proceedings, Part III, Vol. 10212 of Lecture Notes in Computer Science, pp. 551–579.

Ben-Sasson, E., Bentov, I., Horesh, Y., Riabzev, M., 2017c. Stark - scalable transparent arguments-of-knowledge. http://www.cs.technion.ac.il/RESEARCH_DAY_17/POSTERS/michael_riabzev.pdf.

Benaloh, J., de Mare, M., 1993. One-way accumulators: a decentralized alternative to digital signatures. In: Workshop on the Theory and Application of of Cryptographic Techniques EUROCRYPT 1993:Advances in Cryptology EUROCRYPT 93, Vol. 765 of Lecture Notes in Computer Science, pp. 274–285.

Bernstein, D.J., Duif, N., Lange, T., Schwabe, P., Yang, B.-Y., 2011. High-speed high-security signatures. In: Cryptographic Hardware and Embedded Systems - CHES 2011- 13th International Workshop, Nara, Japan, September 28 - October 1, 2011. Proceedings, Vol. 6917 of Lecture Notes in Computer Science. Springer, pp. 124–142.

Bertoni, G., Daemen, J., Peeters, M., Assche, G. V., 2010. Keccak sponge function family main document. Vol. 3, Citeseer.

Biryukov, A., Khovratovich, D., 2016. Equihash: asymmetric proof-of-work based on the generalized birthday problem. In: 23nd Annual Network and Distributed System Security Symposium, NDSS 2016, San Diego, California, USA, February 21-24, 2016. The Internet Society.

Bitansky, N., Chiesa, A., 2012. Succinct arguments from multi-prover interactive proofs and their efficiency benefits. In: Advances in Cryptology - CRYPTO 2012 - 32nd Annual Cryptology Conference, Santa Barbara, CA, USA, August 19-23, 2012. Proceedings, Vol. 7417 of Lecture Notes in Computer Science. Springer, pp. 255–272.

Bitansky, N., Canetti, R., Chiesa, A., Tromer, E., 2012. From extractable collision resistance to succinct non-interactive arguments of knowledge, and back again. In: Innovations in Theoretical Computer Science 2012. ACM, Cambridge, MA, USA, pp. 326–349. January 8-10, 2012.

Bitansky, N., Chiesa, A., Ishai, Y., Ostrovsky, R., Paneth, O., 2013a. Succinct non-interactive arguments via linear interactive proofs. In: Theory of Cryptography - 10th Theory of Cryptography Conference, TCC 2013, Tokyo, Japan, March 3-6, 2013. Proceedings, Vol. 7785 of Lecture Notes in Computer Science. Springer, pp. 315–333.

Bitansky, N., Canetti, R., Chiesa, A., Tromer, E., June 1-4, 2013b. Recursive composition and bootstrapping for SNARKS and proof-carrying data. In: Symposium on Theory of Computing Conference, STOC'13. ACM, Palo Alto, CA, USA.

Bitconnect, 2016. https://en.bitcoinwiki.org/wiki/BitConnect.

Blum, M., Feldman, P., Micali, S., 1988. Non-interactive zero-knowledge and its applications (extended abstract). In: Proceedings of the 20th Annual ACM Symposium on Theory of Computing, May 2-4, 1988. ACM, Chicago, Illinois, USA, pp. 103–112.

Blum, M., Santis, A.D., Micali, S., Persiano, G., 1991. Noninteractive zero-knowledge. SIAM J. Comput. 20 (6), 1084–1118.

Boldyreva, A., 2002. Efficient threshold signature, multisignature and blind signature schemes based on the gap-diffie-hellman-group signature scheme. IACR Cryptol. ePrint Archiv. 118.

Boneh, D., Boyen, X., 2008. Short signatures without random oracles and the SDH assumption in bilinear groups. J. Cryptol. 21, 149–177.

Boneh, D., Corrigan-Gibbs, H., 2014. Bivariate polynomials modulo composites and their applications. In: Advances in Cryptology-ASIACRYPT 2014 - 20th International Conference on the Theory and Application of Cryptology and Information Security, Kaoshiung, Taiwan, R.O.C., December 7-11, 2014. Proceedings, Part I, Vol. 8873 of Lecture Notes in Computer Science. Springer, pp. 42–62.

Boneh, D., Franklin, M.K., 2001. Identity-based encryption from the weil pairing. In: Advances in Cryptology - CRYPTO 2001, 21st Annual International Cryptology Conference, Santa Barbara, California, USA, August 19-23, 2001, Proceedings, Vol. 2139 of Lecture Notes in Computer Science. Springer, pp. 213–229.

Boneh, D., Gentry, C., Lynn, B., Shacham, H., 2003. Aggregate and Verifiably Encrypted Signatures from Bilinear Maps. Springer Berlin Heidelberg.

Boolberry, 2014. Block chain based proof-of-work hash and wild keccak as a reference implementation. http://boolberry.com/files/Block_Chain_Based_Proof_of_Work.pdf.

Bootle, J., Cerulli, A., Chaidos, P., Groth, J., Petit, C., 2016. Efficient zero-knowledge arguments for arithmetic circuits in the discrete log setting. In: Advances in Cryptology - EUROCRYPT 2016 - 35th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Vienna, Austria, May 8-12, 2016, Proceedings, Part II, Vol. 9666 of Lecture Notes in Computer Science. Springer, pp. 327–357.

Bootle, J., Cerulli, A., Ghadafi, E., Groth, J., Hajiabadi, M., Jakobsen, S.K., 2017. Linear-time zero-knowledge proofs for arithmetic circuit satisfiability. In: Advances in Cryptology - ASIACRYPT 2017 - 23rd International Conference on the Theory and Applications of Cryptology and Information Security, Hong Kong, China, December 3-7, 2017, Proceedings, Part III, Vol. 10626 of Lecture Notes in Computer Science. Springer, pp. 336–365.

Bowe, S., Gabizon, A., Miers, I., 2017. Scalable Multi-party Computation for Zk-snark Parameters in the Random Beacon Model. https://eprint.iacr.org/2017/1050.pdf.

Bowe, S., Gabizon, A., Green, M.D., 2017. A multi-party protocol for constructing the public parameters of the pinocchio zk-snark. IACR Cryptol. ePrint Archiv. 2017, 602.

Brassard, G., Chaum, D., Crépeau, C., 1988. Minimum disclosure proofs of knowledge. J. Comput. Syst. Sci. 37 (2), 156–189.

Braun, B., Feldman, A.J., Ren, Z., Setty, S.T.V., Blumberg, A.J., Walfish, M., 2013. Verifying computations with state. In: ACM SIGOPS 24th Symposium on Operating Systems Principles, SOSP '13. ACM, Farmington, PA, USA, pp. 341–357. November 3-6, 2013.

Buldas, A., Laud, P., Lipmaa, H., 2000. Accountable certificate management using undeniable attestations. In: CCS 2000, Proceedings of the 7th ACM Conference on Computer and Communications Security, Athens, Greece, November 1-4, 2000. ACM, pp. 9–17.

Buldas, A., Laud, P., Lipmaa, H., 2002. Eliminating counterevidence with applications to accountable certificate management. J. Comput. Secur. 10 (3), 273–296.

Buterin, V., 2013. Ethereum. https://ethereum.org/.

B. Bünz, J. Bootle, D. Boneh, A. Poelstra, P. Wuille, G. Maxwell, Bulletproofs: Efficient range proofs for confidential transactions. https://eprint.iacr.org/2017/1066.pdf.

Byteball. https://github.com/byteball/byteball.

Bytecoin. https://github.com/dogecoin/dogecoin.

Camacho, P., Hevia, A., Kiwi, M.A., Opazo, R., 2008. Strong accumulators from collision-resistant hashing. In: Information Security, 11th International Conference, ISC 2008, Taipei, Taiwan, September 15-18, 2008. Proceedings, Vol. 5222 of Lecture Notes in Computer Science. Springer, pp. 471–486.

Camenisch, J., Lysyanskaya, A., 2002. Dynamic accumulators and application to efficient revocation of anonymous credentials. In: Advances in Cryptology - CRYPTO 2002, 22nd Annual International Cryptology Conference, Santa Barbara, California, USA, August 18-22, 2002, Proceedings, Vol. 2442 of Lecture Notes in Computer Science. Springer, pp. 61–76.

Camenisch, J., Kohlweiss, M., Soriente, C., 2009. An accumulator based on bilinear maps and efficient revocation for anonymous credentials. In: Public Key Cryptography-PKC 2009, 12th International Conference on Practice and Theory in Public Key Cryptography, Irvine, CA, USA, March 18-20, 2009. Proceedings, Vol. 5443 of Lecture Notes in Computer Science. Springer, pp. 481–500.

Certicom-Research, 2000. Sec 2: Recommended Elliptic Curve Domain Parameters. http://www.secg.org/SEC2-Ver-1.0.pdf.

Chaum, D., 1982. Blind signatures for untraceable payments. In: Advances in Cryptology: Proceedings of CRYPTO'82, Santa Barbara, California, USA, August 23-25, 1982. Plenum Press, New York, pp. 199–203.

Chaum, D., 1988. The dining cryptographers problem: unconditional sender and recipient untraceability. J. Cryptol. 1 (1), 65–75.

Chaum, D., van Heyst, E., 1991. Group signatures. In: Advances in Cryptology - EUROCRYPT '91, Workshop on the Theory and Application of of Cryptographic Techniques, Brighton, UK, April 8-11, 1991, Proceedings, Vol. 547 of Lecture Notes in Computer Science. Springer, pp. 257–265.

Chow, S.S.M., Susilo, W., Yuen, T.H., 2006. Escrowed linkability of ring signatures and its applications. In: Progressin Cryptology - VIETCRYPT 2006, First International Conferenceon Cryptology in Vietnam, Hanoi, Vietnam, September 25-28, 2006, Revised Selected Papers, Vol. 4341 of Lecture Notes in Computer Science. Springer, pp. 175–192.

Coinmarketcap. https://coinmarketcap.com/.

Cromwell, P., Rampichini, M., Beltrami, E., 1998. The borromean rings. Math. Intell. 20, 53–62.

Damgård, I., 1991. Towards practical public key systems secure against chosen ciphertext attacks. In: Advances in Cryptology - CRYPTO '91, 11th Annual International Cryptology Conference, Santa Barbara, California, USA, August 11-15, 1991, Proceedings, Vol. 576 of Lecture Notes in Computer Science. Springer, pp. 445–456.

Damgård, I., 1998. Commitment schemes and zero-knowledge protocols. In: Lectures on Data Security, Modern Cryptology in Theory and Practice, Summer School, Aarhus, Denmark, pp. 63–86.

Damgård, I., Triandopoulos, N., 2008. Supporting non-membership proofs with bilinear-map accumulators. IACR Cryptol. ePrint Archiv. 2008, 538.

Damgård, I., Faust, S., Hazay, C., 2012. Secure two-party computation with low communication. In: Theory of Cryptography - 9th Theory of Cryptography Conference, TCC 2012, Taormina, Sicily, Italy, March 19-21, 2012. Proceedings, Vol. 7194 of Lecture Notes in Computer Science. Springer, pp. 54–74.

Danezis, G., Meiklejohn, S., 2016. Centrally Banked Cryptocurrencies, in: 23rd Annual Network and Distributed System Security Symposium, The Internet Society.

Danezis, G., Fournet, C., Groth, J., Kohlweiss, M., 2014. Square span programs with applications to succinct NIZK arguments. In: Advances in Cryptology - ASIACRYPT 2014 - 20th International Conference on the Theory and Application of Cryptology and Information Security, Kaoshiung, Taiwan, R.O.C., December 7-11, 2014. Proceedings, Part I, Vol. 8873 of Lecture Notes in Computer Science. Springer, pp. 532–550.

Dash. https://www.dash.org/wp-content/uploads/2015/04/Dash-WhitepaperV1.pdf.

de Meer, H., Liedel, M., Pöhls,H. C., Posegga, J., Samelin, K., 2012. Indistinguishability of one-way accumulators. https://www.fim.uni-passau.de/fileadmin/files/forschung/mip-berichte/MIP_1210.pdf.

Derler, D., Hanser, C., Slamanig, D., 2015. Revisiting cryptographic accumulators, additional properties and relations to other primitives. In: Topics in Cryptology-CT-RSA 2015, the Cryptographer's Track at the RSA Conference 2015, San Francisco, CA, USA, April 20-24, 2015. Proceedings, Vol. 9048 of Lecture Notes in Computer Science. Springer, pp. 127–144.

Diffie, W., Hellman, M.E., 1976. New directions in cryptography. IEEE Trans. Inf. Theor. 22 (6), 644–654.

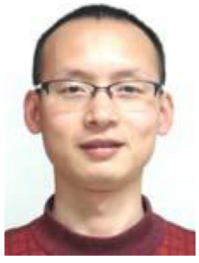
Digibyte. https://github.com/digibyte/digibyte.
Dogecoin. https://github.com/dogecoin/dogecoin.
Etherscan, 2018. Ethereum network hashrate growth rate. https://etherscan.io/chart/hashrate.
Element, 2015. https://github.com/ElementsProject/elements.
Ethash. https://github.com/ethereum/wiki/wiki/Ethash.
Feige, U., Goldwassert, S., Lovász, L., Safra, S., Szegedy, M., 1991. Approximating clique is almost np-complete (preliminary version). In: 32nd Annual Symposium on Foundations of Computer Science, San Juan, Puerto Rico, 1-4 October 1991. IEEE Computer Society, pp. 2–12.
Fiat, A., Shamir, A., 1986. How to prove yourself: practical solutions to identification and signature problems. In: Advances in Cryptology - CRYPTO '86, Santa Barbara, California, USA, 1986, Proceedings, Vol. 263 of Lecture Notes in Computer Science. Springer, pp. 186–194.
França, B., Homomorphic mini-blockchain scheme (2015). http://cryptonite.info/files/HMBC.pdf.
Fujisaki, E., Suzuki, K., 2007. Traceable ring signature. In: Public Key Cryptography-PKC 2007, 10th International Conference on Practice and Theory in Public-key Cryptography, Beijing, China, April 16-20, 2007, Proceedings, Vol. 4450 of Lecture Notes in Computer Science. Springer, pp. 181–200.
Gennaro, R., Gentry, C., Parno, B., Raykova, M., 2013. Quadratic span programs and succinct nizks without pcps. In: Advances in Cryptology - EUROCRYPT 2013, 32nd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Athens, Greece, May 26-30, 2013. Proceedings, Vol. 7881 of Lecture Notes in Computer Science. Springer, pp. 626–645.
Goldwasser, S., Micali, S., Rackoff, C., 1985. The knowledge complexity of interactive proof-systems (extended abstract). In: Proceedings of the 17th Annual ACM Symposium on Theory of Computing, May 6-8, 1985. ACM, Providence, Rhode Island, USA, pp. 291–304.
Goldwasser, S., Lin, H., Rubinstein, A., 2011. Delegation of computation without rejection problem from designated verifier cs-proofs. IACR Cryptol. ePrint Archiv. 2011, 456.
Goyal, R., Goyal, V., 2017. Overcoming cryptographic impossibility results using blockchains. In: Theory of Cryptography-15th International Conference, TCC 2017, Baltimore, MD, USA, November 12-15, 2017, Proceedings, Part I, Vol. 10677 of Lecture Notes in Computer Science. Springer, pp. 529–561.
Groth, J., 2010. Short pairing-based non-interactive zero-knowledge arguments. In: Advances in Cryptology - ASIACRYPT 2010 - 16th International Conference on the Theory and Application of Cryptology and Information Security, Singapore, December 5-9, 2010. Proceedings, Vol. 6477 of Lecture Notes in Computer Science. Springer, pp. 321–340.
Groth, J., 2016. On the size of pairing-based non-interactive arguments. In: Advances in Cryptology - EUROCRYPT 2016 - 35th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Vienna, Austria, May 8-12, 2016, Proceedings, Part II, Vol. 9666 of Lecture Notes in Computer Science. Springer, pp. 305–326.
Hshare. https://github.com/HcashOrg/Hshare.
Innosilicon, 2017a. Innosilicon a4+ ltcmaster. http://www.innosilicon.com/html/a4+-miner/index.html.
Innosilicon, 2017b. Innosilicon a5 dashmaster. http://www.innosilicon.com/html/a5-miner/index.html.
Iota, https://github.com/iotaledger.
Itakura, K., Nakamura, K., 1983. A Public-key Cryptosystem Suitable for Digital Multisignatures, vol. 71. Nec Research & Development, pp. 474–480.
Kate, A., Zaverucha, G.M., Goldberg, I., 2010. Constant-size commitments to polynomials and their applications. In: Advances in Cryptology - ASIACRYPT 2010 - 16th International Conference on the Theory and Application of Cryptology and Information Security, Singapore, December 5-9, 2010. Proceedings, Vol. 6477 of Lecture Notes in Computer Science. Springer, pp. 177–194.
Khatwani, S., 2018. Stratis cryptocurrency (strat) everything you need to know. https://coinsutra.com/stratis-cryptocurrency-strat/.
Kilian, J., 1992. A note on efficient zero-knowledge proofs and arguments (extended abstract). In: Proceedings of the 24th Annual ACM Symposium on Theory of Computing, May 4-6, 1992. ACM, Victoria, British Columbia, Canada, pp. 723–732.
Komodo. https://github.com/SuperNETorg/komodo.
Lamport, L., 1979. Constructing Digital Signatures from a One Way Function. http://lamport.azurewebsites.net/pubs/dig-sig.pdf.
LeMahieu, C., 2016. Raiblocks: a Feeless Distributed Cryptocurrency Network. https://raiblocks.net/media/RaiBlocks_Whitepaper_English.pdf.
Li, X., Jiang, P., Chen, T., Luo, X., Wen, Q., 2018. A Survey on the Security of Blockchain Systems, (CoRR abs/1802.06993).
Li, J., Li, N., Xue, R., 2007. Universal accumulators with efficient nonmembership proofs. In: Applied Cryptography and Network Security, 5th International Conference, ACNS 2007, Zhuhai, China, June 5-8, 2007, Proceedings, Vol. 4521 of Lecture Notes in Computer Science. Springer, pp. 253–269.
Libert, B., Ramanna, S.C., Yung, M., 2016a. Functional commitment schemes: from polynomial commitments to pairing-based accumulators from simple assumptions. In: 43rd International Colloquium on Automata, Languages, and Programming, ICALP 2016, July 11-15, 2016, Rome, Italy, Vol. 55 of LIPIcs, Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, pp. 30:1–30:14.
Libert, B., Ling, S., Nguyen, K., Wang, H., 2016b. Zero-knowledge arguments for lattice-based accumulators: logarithmic-size ring signatures and group signatures without trapdoors. In: International Conference on Advances in Cryptology — EUROCRYPT, pp. 1–31.
Lin, Q., Yan, H., Huang, Z., Chen, W., Shen, J., Tang, Y., 2018. An id-based linearly homomorphic signature scheme and its application in blockchain. IEEE Access PP (99), 1.
Lipmaa, H., 2012. Secure accumulators from euclidean rings without trusted setup. In: Applied Cryptography and Network Security - 10th International Conference, ACNS 2012, Singapore, June 26-29, 2012. Proceedings, Vol. 7341 of Lecture Notes in Computer Science. Springer, pp. 224–240.
Lipmaa, H., 2013. Succinct non-interactive zero knowledge arguments from span programs and linear error-correcting codes. In: Advances in Cryptology - ASIACRYPT 2013 - 19th International Conference on the Theory and Application of Cryptology and Information Security, Bengaluru, India, December 1-5, 2013, Proceedings, Part I, Vol. 8269 of Lecture Notes in Computer Science. Springer, pp. 41–60.
Liquid. https://en.wikipedia.org/wiki/Liquid.
Litecoin. https://litecoin.org/.
Liu, J.K., Wei, V.K., Wong, D.S., 2004. Linkable spontaneous anonymous group signature for ad hoc groups (extended abstract). In: Information Security and Privacy: 9th Australasian Conference, ACISP 2004, Sydney, Australia, July 13-15, 2004. Proceedings, Vol. 3108 of Lecture Notes in Computer Science. Springer, pp. 325–335.
Maji, H.K., Prabhakaran, M., Rosulek, M., 2011. Attribute-based signatures. In: Topics in Cryptology-CT-RSA 2011- the Cryptographers' Track at the RSA Conference 2011, San Francisco, CA, USA, February 14-18, 2011. Proceedings, Vol. 6558 of Lecture Notes in Computer Science. Springer, pp. 376–392.
Mashatan, A., Vaudenay, S., 2013. A fully dynamic universal accumulator. Proc. Rom. Acad. Math. Phys. Tech. Sci. Inf. Sci. 14, 269–285.
Maxwell, G., 2017. Confidential transactions. https://people.xiph.org/greg/confidential_values.txt.
Maxwell, G., Poelstra, A., 2015. Borromean Ring Signatures. https://pdfs.semanticscholar.org/4160/470c7f6cf05ffc81a98e8fd67fb0c84836ea.pdf.
Meng, W., Tischhauser, E., Wang, Q., Wang, Y., Han, J., 2018. When intrusion detection meets blockchain technology: a review. IEEE Access 6, 10179–10188.
Micali, S., 1994. CS PROOFS (extended abstracts). In: 35th Annual Symposium on Foundations of Computer Science. IEEE Computer Society, Santa Fe, New Mexico, USA, pp. 436–453. 20-22 November 1994.
Micali, S., 2000. Computationally sound proofs. SIAM J. Comput. 30 (4), 1253–1298.
Micali, S., Ohta, K., Reyzin, L., 2001. Accountable-subgroup multisignatures: extended abstract. In: CCS 2001, Proceedings of the 8th ACM Conference on Computer and Communications Security, Philadelphia, Pennsylvania, USA, November 6-8, 2001. ACM, pp. 245–254.
Miers, I., Garman, C., Green, M., Rubin, A.D., 2013. Zerocoin: anonymous distributed e-cash from bitcoin. In: 2013 IEEE Symposium on Security and Privacy, SP 2013, Berkeley, CA, USA, May 19-22, 2013. IEEE Computer Society, pp. 397–411.
Monacoin. https://github.com/monacoinproject/monacoin.
Naivecoin. https://github.com/conradoqg/naivecoin.
Nakamoto, S., 2018. Bitcoin: A peer-to-peer electronic cash system, Consulted. https://bitcoin.org/bitcoin.pdf.
Naor, M., Yung, M., 1990. Public-key cryptosystems provably secure against chosen ciphertext attacks. In: Proceedings of the 22nd Annual ACM Symposium on Theory of Computing, May 13-17, 1990. ACM, Baltimore, Maryland, USA, pp. 427–437.
Nem, 2015. https://blog.nem.io/nem-technical-report/.
Nguyen, L., 2005. Accumulators from bilinear pairings and applications. In: Topics in Cryptology - CT-RSA 2005, the Cryptographers' Track at the RSA Conference 2005, San Francisco, CA, USA, February 14-18, 2005, Proceedings, Vol. 3376 of Lecture Notes in Computer Science. Springer, pp. 275–292.
Noether, S., 2015. Ring Signature Confidential Transactions for Monero. IACR Cryptology ePrint Archive, p. 1098.
Nxt. https://bitbucket.org/JeanLucPicard/nxt/src/.
Nyberg, K., 1996. Fast accumulated hashing. In: Fast Software Encryption, Third International Workshop, Cambridge, UK, February 21-23, 1996, Proceedings, Vol. 1039 of Lecture Notes in Computer Science. Springer, pp. 83–87.
Ohta, K., Okamoto, T., 1999. Multi-signature Schemes Secure against Active Insider Attacks, vol. 82. {IEICE} Transactions on Fundamentals of Electronics Communications & Computer Sciences, pp. 21–31.
Okamoto, T., 1988. A Digital Multisignature Scheme Using Bijective Public-key Cryptosystems, vol. 6. ACM Transactions on Computer Systems, pp. 432–441.
Parno, B., Howell, J., Gentry, C., Raykova, M., 2013. Pinocchio: nearly practical verifiable computation. In: 2013 IEEE Symposium on Security and Privacy, SP 2013, Berkeley, CA, USA, May 19-22, 2013. IEEE Computer Society, pp. 238–252.
Parno, B., Howell, J., Gentry, C., Raykova, M., 2016. Pinocchio: nearly practical verifiable computation. Commun. ACM 59 (2), 103–112.
Pedersen, T.P., 1991. Non-interactive and information-theoretic secure verifiable secret sharing. In: Advances in Cryptology - CRYPTO '91, 11th Annual International Cryptology Conference, Santa Barbara, California, USA, August 11-15, 1991, Proceedings, Vol. 576 of Lecture Notes in Computer Science. Springer, pp. 129–140.
Percival, C., 2009. Stronger key derivation via sequntial menory-hard functions. http://www.tarsnap.com/scrypt/scrypt.pdf.
Poelstra, A., 2017. Questions about Borromean Ring Signatures, Personal Communication.
Qtum. https://qtum.org/wp-content/uploads/2017/01/Qtum-technical-white-paper-draft-version.pdf.
Richelson, S., 2014. Cryptographic Protocols with Strong Security: Non-malleable Commitments, Concurrent Zero-knowledge and Topology-hiding Multi-party Computation.
Ripple. https://github.com/ripple/rippled.

Rivest, R.L., Shamir, A., Tauman, Y., 2001. How to leak a secret. In: Advances in Cryptology - ASIACRYPT'2001, 7th International Conference on the Theory and Application of Cryptology and Information Security, Gold Coast, Australia, December 9-13, 2001, Proceedings, Vol. 2248 of Lecture Notes in Computer Science. Springer, pp. 552–565.
Romano, D., Schmid, G., 2017. Beyond bitcoin: a critical look at blockchain-based systems. Cryptography 1 (15), 1–31.
Sander, T., 1999. Efficient accumulators without trapdoor extended abstracts. In: Information and Communication Security, Second International Conference, ICICS'99, Sydney, Australia, November 9-11, 1999, Proceedings, Vol. 1726 of Lecture Notes in Computer Science. Springer, pp. 252–262.
Schnorr, C.-P., 1991. Efficient signature generation by smart cards. J. Cryptol. 4 (3), 161–174.
Shen, J., Gui, Z., Ji, S., Shen, J., Tan, H., Tang, Y., 2018. Cloud-aided lightweight certificateless authentication protocol with anonymity for wireless body area networks. J. Netw. Comput. Appl. 106, 117–123.
Sun, S.-F., Au, M.H., Liu, J.K., Yuen, T.H., 2017. Ringct2.0: a compact accumulator-based (linkable ring signature) protocol for blockchain cryptocurrency monero. In: Computer Security - ESORICS 2017- 22nd European Symposium on Research in Computer Security, Oslo, Norway, September 11-15, 2017, Proceedings, Part II, Vol. 10493 of Lecture Notes in Computer Science. Springer, pp. 456–474.
Tao, M., Ota, K., Dong, M., Qian, Z., 2018. Accessauth: capacity-aware security access authentication in federated-iot-enabled V2G networks. J. Parallel Distr. Comput. 118, 107–117.
Tasca, P., Thanabalasingham, T., Tessone, C. J., 2017. Ontology of Blockchain Technologies. Principles of Identification and Classification, (CoRR abs/1708.04872).
Thoorens, F., Povod, B., Nekrasov, P., Stupurac, S., Beddows, O., 2016. Ark-node. https://github.com/ArkEcosystem/ark-node.
Tsang, P.P., Wei, V.K., 2005. Short linkable ring signatures for e-voting, e-cash and attestation. In: Information Security Practice and Experience, First International Conference, ISPEC 2005, Singapore, April 11-14, 2005, Proceedings, Vol. 3439 of Lecture Notes in Computer Science. Springer, pp. 48–60.
Tschorsch, F., Scheuermann, B., 2016. Bitcoin and beyond: a technical survey on decentralized digital currencies. IEEE Commun. Surv. Tutor. 18 (3), 2084–2123.
van Saberhagen, N., 2013. Cryptonote V 2.0. https://cryptonote.org/whitepaper.pdf.
Vasin, P., 2014. Blackcoin's proof-of-stake protocol v2. https://blackcoin.co/blackcoin-pos-protocol-v2-whitepaper.pdf.
Veeningen, M., 2017. Pinocchio-based adaptive zk-snarks and secure/correct adaptive function evaluation. In: Progress in Cryptology - AFRICACRYPT 2017 - 9th International Conference on Cryptology in Africa, Dakar, Senegal, May 24-26, 2017, Proceedings, Vol. 10239 of Lecture Notes in Computer Science, pp. 21–39.
Verge. https://vergecurrency.com/assets/Verge-Anonymity-Centric-CryptoCurrency.pdf.
Vorick, D., Champine, L., Sia: Simple decentralized storage (2014). https://sia.tech/sia.pdf.
X11. http://wiki.darkcoin.eu/wiki/X11/.
Xu, J., Wei, L., Zhang, Y., Wang, A., Zhou, F., Gao, C., 2018. Dynamic fully homomorphic encryption-based merkle tree for lightweight streaming authenticated data structures. J. Netw. Comput. Appl. 107, 113–124.
Zilliqa, The zilliqa technical whitepaper. https://docs.zilliqa.com/whitepaper.pdf.
Zhu, L., Gao, F., Shen, M., Li, Y., Zheng, B., Mao, H., Wu, Z., 2017. Survey on privacy preserving techniques for blockchain technology. J. Comput. Res. Dev. 54, 2170–2186.

Licheng WANG received the Ph.D. degree from Shanghai Jiao Tong University in 2007. His current research interests include modern cryptography, network security, trust management, etc. He is an associate professor in Beijing University of Posts and Telecommunications, Beijing, China.

Xiaoying SHEN received the M.S. degree from Northwest Normal University in 2017, and she is currently a Ph.D candidate in Beijing University of Posts and Telecommunications. Her current research fields include network security, blockchain technology and isogeny-based cryptography.

Jin LI received the B.S. degree from Inner Mongol Normal University in 2010, the M.S. degree from Shananxi Normal University in 2013 and PhD degree in Beijing University of Posts and Telecommunications. Currently, she works at Guangzhou University. Her research interests include cloud computing, applied cryptography and privacy-preserving, etc.



Jun SHAO received the Ph.D. degree from Shanghai Jiao Tong University in 2008. His current research interests include applied cryptography, blockchain, and security of cloud/fog computing. He is a professor in Zhejiang Gongshang University, China.

Yixian YANG is a Professor of Computer Science and Technology at Beijing University of Posts and Telecommunications and also the director of the National Engineering Laboratory for Disaster Backup and Recovery of China. He is a fellow of China Institute of Communications (CIC), and a council member of Chinese Institute of Electronics (CIE) and Chinese Association for Cryptologic Research (CACR). His research interests include coding theory and cryptography, information security and network security, disaster backup and recovery, signal and information processing, etc.