https://doi.org/10.1007/s11704-025-41297-1

REVIEW ARTICLE

# A survey on threshold digital signature schemes

Yu PENG, Qi FENG✉, De-Biao HE, Min LUO

School of Cyber Science and Engineering, Wuhan University, Wuhan 430072, China

Received November 29, 2024; accepted April 9, 2025

E-mail: fengqi.whu@whu.edu.cn



**Abstract**

Threshold signature, as a privacy-preserving distributed signature, has become the underlying technology in various fields over the last decade. It is useful to protect against a single point of failure and can effectively ensure key security. In recent years, many different digital signatures have been thresholded and many new techniques, algorithms, and protocols have been proposed. This paper introduces the mainstream threshold signature schemes based on the signatures by several standards. We comprehensively investigate various aspects of these threshold signature schemes for comparison and evaluation, and provide the relevant applications and more potential directions for threshold signature.

**Keywords**

threshold digital signature; key protection; secure multi-party computation

## 1 Introduction

*Digital Signature* (DS) developed from public key cryptography, is a cryptographic technology with a signature function, which can guarantee the data integrity, non-repudiation, and anonymity of digital information, and plays an important role in the field of information security. With the current development in the digital era, threshold signatures have gained prominence in the public sphere because of their effective guarantee of the security of keys, and thus have attracted many researchers to carry out thresholding studies on digital signatures.

In the area of information security, the threshold signature maintains the security of keys in many applications as the underlying technology, which allows multiple participants to collaborate on signatures. Even if some of the holders' keys are stolen or tampered with, the signature can normally be completed as long as the specified threshold can be reached. However, threshold signature schemes still have limitations in performance or security. According to the survey, there have been several threshold signature schemes with security flaws.

Since Desmedt and Frankel first proposed the threshold signature scheme [1,2], threshold signatures have attracted extensive attention and research in the cryptographic community. Many government agencies are also looking at the development of threshold signatures, such as National Institute of Standards and Technology (NIST), which called attention to multi-party collaborative schemes in 2023 [3]. In the industry, threshold signatures have become a core technology for blockchain platforms and fintech companies, safeguarding the execution security of multi-party custodial wallets and smart contracts.

Over the past five years, threshold signature schemes have seen an increased presence in top cryptography conferences. Many studies have surveyed and summarized threshold signature schemes. Ergezer et al. [4] conducted a comparative analysis of threshold signature schemes based on ECDSA, Schnorr, and BLS signatures, highlighting their differences and applications. Aumasson et al. [5] provided a detailed survey of threshold ECDSA schemes, dissecting their components and benchmarking their performance and properties. Sedghighadikolaei et al. [6] comprehensively reviewed various threshold signature schemes, including post-quantum threshold signatures and exotic signature variants, along with their application. Jing et al. [7] aligned the survey with foundational signatures in NIST standard, cataloguing common threshold signature types while exploring threshold signcryption techniques, applications, and standardization progress. However, critical gaps persist in these works: 1) incomplete coverage of signature types (e.g., omitting ECDSA-based threshold designs); 2) vague evaluation metrics for scheme comparison; 3) lack of clear developmental roadmaps for emerging scenarios.

To address these limitations, we investigate the evolution of threshold signatures and review innovative and representative studies from the past decade. Following the framework of basic digital signature types outlined in International Organization of Standardization) (ISO) reports [8], we select a variety of mainstream signatures based on different algorithmic structures and recent literature, which can be broadly classified into Schnorr, EdDSA, ECDSA, pairing-based, and SM2 signatures. We systematically

organize their theoretical advancements and engineering challenges, analyze differences in thresholding designs, and evaluate their applicability across multiple dimensions.

This survey aims to introduce the mainstream threshold digital signature schemes and differ in various aspects including security, functionality, technology, and performance. Our contributions are focused on the following:

- We examine the similarities and differences between threshold Schnorr and EdDSA signatures, investigating the feature of scalability and determinism of these schemes.
- We provide a comprehensive description of threshold ECDSA and SM2 signatures, presenting the relationship between each scheme and improvements, and then categorize the schemes by the technology and functionality.
- We analyze and investigate a variety of pairing-based threshold signatures, including BLS, SM9, and identity-based signatures in IEEE p1363.
- We compile current applications of threshold signatures and difficulties encountered in thresholding digital signatures, and propose thoughts for future directions.

The rest of this paper are organized as follows. Section 2 illustrates the notations and relevant knowledge of threshold signature. In Section 3, we review some of the main threshold Schnorr digital signature schemes and difficulties. Section 4 and Section 5 discuss the construction of the Schnorr signature variants, which is the thresholding of EdDSA and ECDSA schemes. In Section 6, we examine the threshold pairing-based digital signature. We present the SM2-based threshold signature schemes in Section 7. Finally, we investigate the applications in Section 8 and directions of several threshold signatures in Section 9.

## 2 Preliminaries

In this section, we introduce the fundamentals and related techniques of threshold signature schemes. First, we present the notations in this paper, then we introduce definitions, assumptions and security of threshold digital signature. We also show some cryptographic primitives used in designing threshold schemes. Finally, we give the evaluation criteria for different schemes.

### 2.1 Notations

Let $\kappa$ be the security parameter and $n$ be the number of parties. Thus $t$ represents the threshold of signature and $t_c$ is the number of maximally corrupt participants. We denote $P_i$ as the $i$th participant in the scheme. $sk$ and $pk$ represent private and public key, $m$ represent the message. We denote $\mathsf{H}$ as a secure cryptographic hash function, and we use $\mathsf{H}_s, \mathsf{H}_d, \mathsf{H}_e$, and $\mathsf{H}_b$ represent the hash function in Schnorr, EdDSA, ECDSA, and BLS signature, respectively. $\mathcal{A}$ represents the adversary in schemes. Many notations will be described in different signatures.

### 2.2 Digital signature

Digital signature is a cryptographic technology with a signature function. In DS, a signer $S$ holds a pair of key $(sk, pk)$ and uses private key $sk$ to sign a message $m$. Then verifier can check on the integrity of the signature $\sigma$ by public key $pk$ to verify whether the signature is valid. The definition of DS is as follows:

**Definition 1** (Digital Signature). A digital signature scheme $(DSS)$ involves four algorithms $(Setup, KG, Sign, Ver)$, which is defined as follows:

- $\mathbb{PP} \leftarrow DSS.Setup(\kappa)$: On input the security parameter $\kappa$, this algorithm outputs the system domain parameters.
- $(sk, pk) \leftarrow DSS.KG(\mathbb{PP})$: This is a randomized algorithm to initialize the parameters in the scheme and generate a key pair $(sk, pk)$ with $\mathbb{PP}$ input.
- $\sigma \leftarrow DSS.Sign(m, sk)$: On input the message $m$ and private key $sk$, this randomized algorithm can generate and outputs the signature $\sigma$.
- $b \leftarrow DSS.Ver(m, \sigma, pk)$: This is a deterministic algorithm to output bit $b \in \{0,1\}$ on input of the pair $(m, \sigma)$ and $pk$, which represent the signature is valid iff $b = 1$.

### 2.3 Threshold digital signature

**Definition 2** (Threshold Digital Signature). A $(t, n)$ threshold digital signature scheme $(TDSS)$ accomplishes a valid signature iff the number of participants $k \geqslant t$, which consist of four algorithms $(Setup, KG, Sign, Ver)$:

- $\mathbb{PP} \leftarrow TDSS.Setup(\kappa)$: On input the security parameter $\kappa$, this algorithm outputs the system domain parameters.
- $(\langle sk_1, sk_2, ..., sk_n\rangle, PK) \leftarrow TDSS.KG(\mathbb{PP})$: Given the parameter $\kappa, n, t$, this randomized algorithm outputs the public key $PK$ and a set of secret keys $sk_i$.
- $\sigma \leftarrow TDSS.Sign(m, \{sk_i\})$: This randomized algorithm can generate and output the signautre $\sigma$ with the message $m$ and keys $(|\{sk_i\}| \geqslant t)$ input.
- $b \leftarrow TDSS.Ver(m, \sigma, PK)$: This deterministic algorithm outputs a bit $b \in \{0,1\}$ with the pair of signature $(m, \sigma)$ and the $PK$ input, where represent the signature is valid iff $b = 1$.

### 2.4 Difficult problems and security definitions

**Definition 3** (Discrete Logarithm Problem). Given a group $G$ and generator $g$, choose an element $h \in G$, it is hard to find an integer $a$ such that $h = g^a$.

**Definition 4** (Elliptic Curve Discrete Logarithm Problem). Given the elliptic curve $E$ and the points $P, Q \in E(\mathbb{F}_p)$, it is hard to find an integer $a$ such that $Q = aP$.

**Definition 5** (Elliptic Curve Decisional Diffie-Hellman Problem). Given an elliptic curve group $\mathbb{G}$ with prime order $q$ and generator $G$, compute $A = a \cdot G, B = b \cdot G$ and randomly choose $C \in \mathbb{G}$, where $a, b \leftarrow_R \mathbb{Z}_q$ and $C \in \mathbb{G}$. For two tuples $(A, B, a \cdot b \cdot G)$ and $(A, B, C)$, the value of $C$ and $a \cdot b \cdot G$ are computationally indistinguishable.

**Definition 6** (EUF-CMA). Given a digital signature scheme $DSS$=$(Setup, KG, Sign, Ver)$, which is considered a security model

under the Existential Unforgeability against chosen-message attack (EUF-CMA). It can be described as follows:

- **Setup** The challenger runs $DSS.KG$ to get $(sk, pk)$ with system public parameter $\mathbb{PP}$ input and send $pk$ to adversary $\mathcal{A}$.
- **Query** The challenger responds multiple times to the message $m_i$ and signature queries sent by the adversary $\mathcal{A}$ and runs the signature algorithm and $sk$ to generate the signature $\sigma_{m_i}$ and send it to the adversary $\mathcal{A}$.
- **Forgery** The adversary $\mathcal{A}$ output a forged signature $\sigma^*$ about $m^*$. $\mathcal{A}$ win the game when satisfies both $Ver(m^*, \sigma^*, pk) = 1$ and $m^* \notin \{m_1, \ldots, m_i\}$. We denote this probability as advantage $\epsilon$.

A digital signature scheme $DSS$ is secure if the probability $\epsilon$ is negligible after adversary $\mathcal{A}$ making $i$ signature queries in polynomial time $t$.

**Definition 7** (Robustness). A threshold signature scheme is robust if $t$ participants comply with the protocol and generate a valid signature, even if $t-1$ participants are corrupted.

### 2.5 Secure multi-party computation primitive

The concept of secure multi-party computation (MPC) was born out of "The Millionaire" problem for two parties by Yao [9] in 1982. These technologies can compute the private input distributed in function for all participants and obtain the output without any leak. They are often integrated with technologies from various other domains [10−13], and many primitives are used in the schemes to optimize communication and computation.

Threshold signatures are a concrete application of MPC. Leveraging underlying MPC techniques can significantly enhance the applicability, security, and robustness of signature schemes in distributed scenarios. This includes cryptographic primitives such as secret sharing, oblivious transfer (OT), and homomorphic encryption (HE). We will briefly introduce some of the important MPC technologies for threshold schemes.

#### 2.5.1 Secret sharing

Secret sharing is a technology of dividing the secret into multiple sub-secrets and distributing them to different participants to protect the security and availability. The first mention of secret sharing is proposed by Shamir [14] and Blakley [15] in 1979. In the design of schemes for threshold signatures, the most commonly used is Shamir secret sharing (SSS) based on the Lagrange polynomials interpolation.

**Definition 8** (Shamir secret sharing). Given the secret value $s$, a $(t, n)$ SSS scheme over $\mathbb{F}_p$ denotes that split the $s$ to $n$ participants and at least $t$ participants can jointly reconstruct $s$. Choose a set of parameters $\{a_1, a_2, \ldots, a_{t-1}\}$ and generate a polynomial with order $t-1$, $f(x) = s + \sum_{i=1}^{t-1} a_i \cdot x_i$, the $s_i = f(x_i)$ denotes the share of $s$ and pair $(x_i, s_i)$ are sent to parties, where $x_i \leftarrow_R \mathbb{F}_p$. Thus any $t$ participants can jointly compute

$$f(x) = \sum_{i=1}^{t} \sum_{j=1, j\neq i}^{t} \frac{x - x_j}{x_i - x_j}.$$

The SSS scheme only considers the Dealer to be trustworthy. However, if the Dealer does something evil, such as assigning random values to parts of the secret that are inconsistent across participants, the results can be disastrous. The verifiable secret sharing (VSS) was proposed in 1985 by Choc et al. [16] to solve this problem, and then many VSS protocols were invented.

Feldman's VSS [17] provides an efficient method to ensure the accuracy and security of secret distribution for environments that require synchronized communication. Pedersen's VSS [18] is based on a commitment scheme for securing the verification of secrets and polynomial coefficients. It makes the verification of secret shares more secure and unbiased by exploiting homomorphic cryptographic properties. A distinctive feature of Pedersen's VSS is that it resists interference from active attackers and provides higher security.

Publicly verifiable secret sharing (PVSS) [19] is a publicly VSS scheme that allows anyone (not limited to the participants) to verify the correctness of a secret share. This public verification mechanism enhances the transparency and trust of the system and is particularly suitable for scenarios that require public auditing and verification.

Asynchronous verifiable secret sharing (AVSS) [20] is a verifiable secret sharing protocol implemented in an asynchronous communication environment. It does not rely on synchronous assumptions, allowing messages to arrive at any time and dealing with message delays and sequential uncertainty. AVSS enhances system robustness and resilience by providing security and accuracy guarantees in asynchronous environments suitable for distributed systems, blockchains, and other asynchronous network environments.

Proactive secret sharing (PSS) [21] is designed to address long-term security threats by periodically refreshing the secret shares held by participants without changing the original secret. This scheme resists static attacks by constantly refreshing secret shares, increasing the security and robustness of the system for long-running distributed systems and critical infrastructures.

#### 2.5.2 Homomorphic encryption

Homomorphic encryption (HE), one of the fundamental technologies in privacy-preserving computation, was first conceptualized by Rivest et al. in 1978 [22]. As a cryptographic technique, HE enables direct computations on encrypted data, allowing mathematical operations to be performed on ciphertexts such that the decrypted result matches the outcome of equivalent operations executed on plaintext.

HE is classified into Partially Homomorphic Encryption (PHE) and Fully Homomorphic Encryption (FHE): PHE supports a single type of operation (e.g., addition or multiplication), whereas FHE permits arbitrary combinations of addition and multiplication. Since Craig Gentry's groundbreaking work in 2009 [23], which introduced the first practical FHE scheme based on ideal lattices, numerous FHE variants—such as BFV [24], GSW [25], and CKKS [26] have been proposed. Subsequent advancements have focused on algorithmic improvement and hardware acceleration to enhance the computational efficiency of homomorphic encryption systems.

#### 2.5.3 Oblivious transfer

Oblivious transfer (OT) is a cryptographic protocol designed to protect privacy in communication between two parties, ensuring selective privacy during message exchange. OT guarantees that the sender cannot determine which messages the receiver has selected and the receiver only obtains the specific information they chose and gains no knowledge about unselected messages.

The first OT protocol was proposed by Rabin in 1981 [27], but it achieved message delivery with only a $50\%$ success probability. In 1985, Even et al. [28] introduced the 1-out-of-2 OT protocol, laying the foundation for modern OT frameworks. Subsequent research expanded OT protocols based on diverse cryptographic primitives and mathematical hardness assumptions, such as the development of 1-out-of-n OT [29], Random OT (ROT) [30], and OT extension [31] techniques. Today, OT serves as a foundational component of MPC, enabling its widespread adoption in cryptographic schemes and real-world applications—from privacy-preserving data analysis to secure federated learning.

### 2.6 Evaluation criteria

In this paper, we investigate two aspects of the threshold signature scheme, namely performance and security.

#### 2.6.1 Performance

The performance criteria for threshold signatures are contained in the following categories:

- Communication rounds. In a multi-party collaborative task of completing a threshold signature scheme, each participant needs to compute and send intermediate variables, and receive the information sent by the other participants. Communication round refers to the number of times information is exchanged between participants during the execution of the protocol, and each round of communication typically involves all participants sending and receiving a specific type of information. Until now, many algorithms have supported the pre-signing phase, i.e., participants can execute some steps in the offline phase that do not require the involvement of message $m$, thus reducing the number of rounds of online interactions between participants.
- Communication overhead. Communication overhead refers to the amount of resources consumed and data transmitted by participants to exchange information for signature generation during protocol execution. In addition to the computational intermediate variables of signature, there are commitments and zero-knowledge proofs for interactions in the interaction process.
- Computation complexity. This criterion contains the computation overhead of local computation by each participant. Participant local computation includes the intermediate variables for signatures and the steps to implement the various functionalities and the cryptographic primitives.

#### 2.6.2 Security

Security criteria cover various aspects, including threshold optimality, security assumption, and adversary behaviour:

- Threshold optimality. This criterion means that the total number of participants in a $(t,n)$ threshold signature algorithm need only satisfy $n \geqslant t$. Earlier threshold signature algorithms [32] require $n \geqslant 2t-1$, i.e., if $t$ participants can cooperate to produce a complete signature, the private key needs to be split to at least $2t-1$ nodes, which increases the operation cost and the risk of private key leakage.
- Security assumption. In security proofs, the process of solving some difficult problem is statistically represented as an attack on the scheme, i.e., the security of a cryptographic scheme is reduced to the difficulty of the assumptions. The algorithm is secure if the problem is recognized as difficult or unsolvable. Weaker security assumptions indicate that the algorithm is more secure and has fewer security dependencies. The threshold signature schemes usually involve strong RSA assumptions and standard ECDSA assumptions, etc.
- Adversary behaviour. Adversary behaviour can be described in several ways:
  1) The *semi-honest* adversary tries to obtain as much information and corrupt other parties but follows the protocols. The *malicious* adversary will corrupt parties and cause them to deviate from the protocols.
  2) When there are more than half the number of honest participants in the model, it is called *honest majority*. Conversely, when the adversary corrupts more than half of the participants, it is called *dishonest majority*.
  3) Based on the ability of the adversary to corrupt, they are categorized into *Static* and *Adaptive* models. The first model can be called *Static* when the adversary selects parties to corrupt before the start of the protocol and keeps it unchanged during the protocol process. The *Adaptive* model means that the adversary can corrupt honest participants at any time.

## 3 Threshold schnorr digital signature

Based on the threshold signatures mentioned in the ISO's report [8], we first investigate the Schnorr signature [33] and its threshold schemes based on the Discrete Logarithm Problem (DLP). Variants of Schnorr signatures include EC-Schnorr, EdDSA, and ECDSA over the elliptic curve. We describe Schnorr signatures on integer groups and their associated definitions below.

**Definition 9** (Schnorr Digital Signature Scheme). The Schnorr digital signature scheme [33] consists of four algorithms, the definition is described as follows:

1. $\mathbb{PP} \leftarrow Schnorr.Setup(\kappa)$: On input the security parameter $\kappa$, this algorithm outputs public parameters $\mathbb{PP} = ((p,q,\mathbb{Z}_p^{\times}, g), \mathsf{H}_s)$, where $p,q$ are two primes, $\mathbb{Z}_p^{\times}$ is Schnorr group with prime order $q$ and generator $g$, $\mathsf{H}_s$ is hash function.
2. $(d,Q) \leftarrow Schnorr.KG(\mathbb{PP})$: Choose a private key $d \leftarrow_R \mathbb{Z}_q$ and compute the public key $Q = g^d$.
3. $\sigma \leftarrow Schnorr.Sign(m,d)$: The signer choose a nonce $k$ and

compute $r = g^k$, then compute the signature pair $\sigma = (e, s)$ to output for the message $m$, where $e = \mathsf{H}_s(r, m), s = d \cdot e + k$.

4. $b \leftarrow Schnorr.Ver(m, \sigma, Q)$: The verifier computes $r' = g^s Q^{-e}$ by $m$ and the signature $\sigma$, and checks whether $\mathsf{H}_s(r', m) = e$. If the equation holds, the verifier outputs $b = 1$.

The linearization advantage of the Schnorr signature structure makes it easy to aggregate signature splits. The security of Schnorr signatures in multi-party environments is a challenge, such as whether the scheme can successfully output legitimate signatures. During the signature aggregation phase, it is important to prevent *Rogue Key Attacks* by adversaries and avoid forging illegal signatures. In addition, how to increase the number of maximum corrupted participants to improve security has become the focus of scholars' research.

### 3.1 Robust and non-robust threshold schnorr signature

Early thresholding of Schnorr signatures did not attract much attention from researchers, whose studies focused on the design of Schnorr signatures. Therefore, we denote the scheme in which valid signatures can be accomplished by any $t$ participants correctly following the protocol as *robust* and the scheme in which the deviation of the participants from the protocol is detected as *non-robust*.

#### 3.1.1 Robust schemes

In 2001, Stinson et al. [34] directly utilized the Distributed Key Generation (DKG) protocol in [35] to perform key generation and distribution in a multi-participants environment. This scheme requires at least four rounds of communication by the participants, of which the DKG during the signing process accounts for three rounds. Similarly, Gennaro et al. [36] improved the DKG protocol of Pedersen [37] with only two rounds of operation in 2003, which differs from [34] in that the random number $k$ will be generated in the signature phase. They further proposes a method involving a trusted third party to perform signature aggregation, which was later adopted in subsequent work [38].

However, the above schemes have some limitations, such as not being able to achieve threshold optimality, i.e., not being able to satisfy the setting that only requires $n > t$. Also, these schemes are unable to distinguish participant failure behaviour, and the other signers will reconstruct the key shares even if a signer has a benign failure instead of a malicious failure.

Joshi et al. [39] proposed an asynchronous scheme ATSSIA with non-interactivity based on problems with FROST, which reduces security to the DLP problem while supporting dishonest majorities. Their protocol performs twice DKG per signature and is robust and non-interactive in the actual signature phase, but still can not avoid the high cost of many non-robust and synchronous steps involving multiple rounds in the pre-signing phase.

Ruffing et al. [40] proposed methods to provide robustness in the sign phase. The paper constructs a wrapper that only needs one preprocessing and one online signing round to complete the signature, making the protocol robust and asynchronous. This wrapper can be applied to FROST [38] while satisfies the condition of non-robustness, provides robustness by having concurrent signing sessions, and provides the functionality of identifiable aborts. Finally, this paper points out that their wrapper can be combined with the scheme proposed by González et al. [41], which guarantees robustness in the key generation phase in [38], to achieve a fully robust threshold Schnorr signature scheme.

Benhamouda et al. [42] proposed a new threshold Schnorr scheme called SPRINT for a large number of participants scenario that guarantees robustness on asynchronous networks with high throughput. This paper considers protocol requirements based on a blockchain scenario and aims to generate multiple Schnorr signatures at once to spread the protocol cost. Their signature phase also uses an optimized DKG method to generate message-independent random numbers in two rounds, and then generates multiple signatures in a non-interaction theory. To ensure that the protocol performance is not affected in the presence of network latency of the participants, they forgo complete secret sharing [43] and use an early negotiation protocol. To maximize the signature efficiency, they proposed a component combining packed secret sharing [44] with super-invertible matrix [45] bonding called extreme packing, and proposed to share long-term keys in the packing vector so that each participant can perform multiplication and randomization locally. The cost, however, makes it possible to reduce the number of maximally corrupt participants $t_c$ to $n - 2a + 1$, where $a$ is an efficiency parameter. The security of this paper only relies on the DL problem under the programmable Random Oracle Model (ROM) model, and Shoup [46] points out that this scheme can be combined with FROST [38] to improve security.

In 2024, Groth et al. [47] also proposed another asynchronous robustness protocol. Unlike SPRINT, this protocol has optimal resilience that $t_c < n/3$. In this paper, the authors proposed a GoAVSS protocol and a new super-invertible matrix algorithm to support their MPC engine. The GoAVSS protocol is responsible for the distribution and verification of secrets and is constructed based on the protocol of [48], which uses error-correcting codes to avoid the overhead of polynomial commitments. The super-invertible matrix efficiently combines temporary public keys to generate fewer public keys. Thus their scheme supports an efficient pre-signature generation process, which generates a large number of pre-signatures in the offline phase through batch processing techniques, thus reducing the computational and communication cost in the online signature phase. This design greatly improves the throughput of signature generation and supports the generation of a large number of signatures per second.

#### 3.1.2 Non-robust schemes

In 2021, Komlo and Goldberg proposed a non-robust scheme called the FROST protocol [38]. The program suspends the protocol and restarts when any participant is detected to have deviated from the protocol and can identify malicious participants. The Keygen protocol in FROST is a variant of Pedersen's DKG [37], which introduces zero-knowledge proofs of secret values to prevent Rogue Key Attacks [49] in $t \geqslant n/2$ case. The FROST signature protocol is a

two-round interaction protocol. Since the first round of operations does not involve the message $m$ to be signed, it can be executed in the pre-processing phase. The FROST protocol has a Signature Aggregator role mentioned in [36], which collects the intermediate variables computed by all the participants and composes a list $B=(i,D_i,E_i)$, where secret share $k_i=d_i+e_i\cdot\rho_i$ and $D_i=g^{d_i},E_i=g^{e_i}$. Then, the aggregator sends the list and the message $m$ to the selected participants for the following computation and verification, which is resistant to attack in [50]. Each signature participant can replace this role in the implementation so that messages previously sent to the signature aggregator are instead sent to all the other participants.

### 3.2 Security assumption

After FROST's birth, many researchers began to invest more attention to security and safety assumptions.

To study the security of Schnorr threshold signatures more uniformly, in 2022, Bellare et al. [51,52] proposed a new security-definition structure called TS-UF-$i$, for $i\in[0,4]$, and designed the FROST2 algorithm in [51]. In the static corruption model, FROST and FROST2 can statute security proofs to the One-More Discrete Logarithm (OMDL) [53] assumption under the ROM. The scheme devises a variant of Pedersen's DKG protocol [37] in the key generation phase by adding proofs of possession, called PedPoP. Schnorr Knowledge of Exponent Assumption (Schnorr-KOE) is introduced to guarantee the operation of PedPoP in the dishonest majority case, and the proof of infers that the security of this signature can be statistically up to the DL assumption under the Algebraic Group Model (AGM). Finally, this paper gives the proof of TS-UF-3 secure for FROST and TS-UF-2 secure for FROST2.

Lindell proposed a new three-round interaction threshold Schnorr algorithm [54] in 2022, which similarly avoids the reuse of DKG in each signature session and guarantees point-to-point encrypted transmissions between participants under the PKI system. Unlike FROST, the security of this paper only relies on the DL problem under the ROM model and is resistant to Random inhomogeneities in a Overdetermined Solvable system of linear equations (ROS) attacks. Furthermore, this paper states that the scheme is UC-secure when the zero-knowledge proof and commitment functions used are UC-secure. There are two issues with the paper to be addressed in subsequent work, one is that the use of zero-knowledge proofs results in poor performance, and the other is that the protocol is only secure in the static model. It also points out the conflict of Schnorr's non-deterministic EdDSA signatures.

In 2023, Crites et al. [55] proposed a three-round interactive Sparkle+ algorithm, which is the first scheme to fully implement adaptive security relying on strong interactive assumptions. This paper continues Lindell's approach [54], again under the ROM assumption and the DLP assumption in being statically secure, but with no need for zero-knowledge proofs online, which is more efficient. Moreover, the security of the algorithm can be generalized to the Algebraic One-More Discrete Logarithm Assumption (AOMDL) problem [56] under the AGM and the stochastic predicate machine model, but the number of corrupting parties needs to be set up to satisfy $t_c\leqslant t/2$ under the adaptive security condition.

In 2024, Bacho et al. [57] proposed HARTS with high-threshold, adaptive security, and robustness. This scheme ensures the generation of valid signatures in asynchronous networks and maintains security at most $t_c<n/3$ malicious participants. Based on the DKG idea of [35], HARTS uses asynchronous verifiable secret sharing (AVSS), MVBA [58], and Superinvertible matrix [47] to construct Packed Asynchronous Distributed Key Generation technique that ensures security and efficiency in the case of malicious behaviour and network delays. This scheme balances the efficiency and security with an amortized communication cost of $O(\lambda n^2\log n)$ per signature and a constant number of interaction rounds. This scheme relies on the AGM and the OMDL to ensure security under adaptive decay.

Bacho et al. [59] also proposed Twinkle in the same year, a new scheme for achieving comprehensive adaptive security that uses Tagged Linear Function Families and Decisional Diffie-Hellman (DDH) assumptions to provide non-interactive security proofs. It also relies on the five-move identification to ensure the security and efficiency of the signature process. Twinkle simplifies the implementation complexity compared to Sparkle+ [55] by not relying on AGM, supporting arbitrary decay thresholds $t_c<n$, and avoiding rollback techniques. This paper points out that there is a gap in Sparkle's security proofs, i.e., an adversary may send inconsistent sets of promises to different honest parties, resulting in invalid signatures. For this reason, Twinkle introduces new techniques such as the equivalence class to solve this problem. Twinkle's signature size is at most three times the size of an ordinary Schnorr signature and can be accomplished in only three rounds of interactions.

### 3.3 Deterministic

In 2021, Garillot et al. [60] proposed a stateless deterministic signature scheme. This paper focuses on the implementation of stateless deterministic signatures in threshold signature schemes. The traditional Schnorr signature scheme needs to generate a new random number every time a signature is made, which may cause serious security problems in practical applications because the random number generator may suffer from rollback attacks and is vulnerable to factors such as system crashes and malicious attacks. This paper proposes a stateless threshold Schnorr signature scheme, which causes each participant to generate a nonce as the private key, and calculates the random number share through a deterministic algorithm to achieve deterministic random number derivation, avoiding the unreliability of the random number generator. By introducing a UC commitment-based mechanism, the paper implements and constructs a commitment scheme based on Zero-Knowledge from the Garbled Circuit (ZKGC) [61] with unobtrusive transmissions, which allows the verifier to submit a secret value once and prove its correctness several times afterwards. Also, the paper proposes a cryptographic device for converting intermediate cryptographic circuit line labels into arithmetic codes, thus simplifying the process of zero-knowledge proofs. This scheme possesses fewer rounds and improves the performance compared to the scheme using MPC protocol [62].

The first two-party Schnorr threshold scheme with statelessness and determinism was presented by Kondi et al. [63]. This paper uses Pseudorandom Correlation Functions (PCF) to generate unpredictable pseudo-random numbers. Two instantiations are constructed based on Vector Oblivious Linear Evaluation (VOLE), one is a protocol based on [64] that satisfies covert security [65], and the other is an active malicious security protocol based on [66]. The first protocol achieves security based on PRF and OT assumptions, while the second achieves security based on Decisional Composite Residuosity (DCR) and strong RSA assumptions, but it is still not possible to achieve both covert and malicious security. The stateless design of this scheme eliminates the state dependency in the signature process and prevents the security risk of state reuse while completing the signature in two rounds of interaction. The limitation of this scheme is that the constructed PCF only supports two-party collaboration, and the extension to $t$-party is still a problem to be realized.

### 3.4 Other functionality

Boneh et al. [67] proposed a new scheme TAPS in 2022. This paper points out that both the present private threshold signature (PTS) and accountable threshold signature (ATS) schemes do not balance the characteristics of both, and thus TAPS is proposed to guarantee the tracking of the signer if it is necessary in some cases, but maintain privacy in the ordinary case. The main feature of the scheme is to protect the privacy of the signer through public key encryption and zero-knowledge proofs while allowing the signer to be traced when necessary. In the construction of the scheme, they use ElGamal encryption for encrypting part of the information of the combined signature to ensure the privacy of the set of signers and $\Sigma$ protocol or Bulletproofs [68,69] for proving the validity of the encrypted signatures without revealing the information of the set of signers. The scheme requires three rounds and one round of interactions in the signature generation and verification process respectively, with relatively low communication and computation overheads, especially when Bulletproofs are used, and the length of the signatures and proofs are shorter, which makes it suitable for large-scale signer collections. Combiner collects the signature shares of the signers, generates the ATS signatures, encrypts them as $ct$, and generates the zero-knowledge proofs $\pi$, and finally outputs the signature $\sigma' = (ct, \pi)$. This scheme allows only the person who owns the traceability key to track down the signer's information when necessary, providing a secure and efficient solution for a variety of application scenarios.

Finally, we give Tables 1 and 2 to summarize some of the schemes in recent years, which contain the network of the scheme, the maximum corrupted parameter $t_c$, the number of signature rounds, the attack model, the robustness, and the security assumptions.

**Table 1** Summary of threshold Schnorr protocols for evaluation

| Scheme | Corruption | Rounds | Communication |
|---|---|---|---|
| SS01 [34] | $t_c \leqslant t < n/2$ | 5 | $O(\lambda n^2)$ |
| Lin22 [54] | $t_c \leqslant n$ | 3 | $O(\lambda n^2)$ |
| FROST [38] | $t_c \leqslant t$ | 2 | $O(\lambda n^2)$ |
| FROST2 [51] | $t_c \leqslant t$ | 2 | $O(\lambda n^2)$ |
| ROAST [40] | $t_c < n$ | 2 | $O(\lambda n^3 + n^4)$ |
| Sparkle [55] | $t_c \leqslant t/2$ | 3 | $O(\lambda n^2)$ |
| SPRINT [42] | $t_c < n/3$ | 3 | $O(\lambda n^2)$ |
| GS23 [47] | $t_c < n/3$ | $O(1)$ | $O(\lambda n)$ |
| HARTS [57] | $t_c < n/3$ | $O(1)$ | $O(\lambda n^2)$ |
| Twinkle [59] | $t_c \leqslant t$ | 3 | $O(\lambda n^2)$ |

**Table 2** Summary of threshold Schnorr protocols in functionality

| Scheme | Network | Adaptive | Robust | Assumptions |
|---|---|---|---|---|
| SS01 [34] | sync | × | × | DLP+ROM |
| Lin22 [54] | async | × | × | DLP+ROM |
| FROST [38] | sync | × | × | DLP |
| FROST2 [51] | async | × | × | AOMDL+ROM |
| ROAST [40] | async | × | √ | AOMDL+ROM |
| Sparkle [55] | sync | √ | × | AOMDL+ROM |
| SPRINT [42] | async | × | √ | DLP+ROM |
| GS23 [47] | async | × | √ | ROM |
| HARTS [57] | async | √ | × | OMDL+ROM |
| Twinkle [59] | sync | √ | × | AOMCDH+ROM |

⋆ sync means synchronous network and async means asynchronous network.

⋆ DLP denotes the discrete logarithm problem, ROM denotes the random oracle model, AOMDL denotes algebraic one-more discrete logarithm assumption, AGM denotes algebraic group model, OMDL denotes the one-more discrete logarithm assumption.

## 4 Threshold EdDSA digital signature

The edwards-curve digital signature algorithm (EdDSA) [70,71] is the variant of Schnorr signature on the twisted Edwards curves, and its security is built based on Elliptic Curve Discrete Logarithm Problem (ECDLP). The relevant definitions of EdDSA are described as follows:

**Definition 10** (EdDSA Digital Signature Scheme). Given a twisted elliptic curve $E(\mathbb{F}_p)$, the algorithms of EdDSA signature are described as follows:

1. $\mathbb{PP} \leftarrow EdDSA.Setup(\kappa)$: On input the security parameter $\kappa$, this algorithm output the system domain parameters $\mathbb{PP} = (\mathbb{G}, q, G, E(\mathbb{F}_p), l, \mathsf{H}_d)$, where $\mathbb{G}$ is an additive cycle group with order $q$ and generator $G$, and $\mathsf{H}_d$ is a hash function.
2. $(sk, s, pk) \leftarrow EdDSA.KG(\mathbb{PP})$: Select a secret key $sk \leftarrow_R \{0,1\}^l$ and compute $\mathsf{H}_d(sk) = (h_0, h_1, \ldots, h_{2l-1})$. Then set $h_0 = h_1 = h_2 = h_{l-1} = 0$ and $h_{l-2} = 1$, thus define the secret scalar $s = \sum_{i=0}^{l_1-1} h_i \cdot 2^i$ and compute $pk = s \cdot G$, where $s \in \mathbb{Z}_q$

3. $\sigma \leftarrow EdDSA.Sign(m,sk,pk)$: Compute a randomness $r = \mathsf{H}_d(\mathsf{H}_{d_R}(sk),m)$, where $\mathsf{H}_{d_R}$ is the right half of hash value. Then compute $R = r \cdot G, c = \mathsf{H}_d(r,pk,m)$ and $S = r + c \cdot s$. Finally output the signature $\sigma = (R,S)$.
4. $b \leftarrow EdDSA.Ver(m,pk,\sigma)$: Verifier first computes $c' = \mathsf{H}(R,pk,m)$ and check whether $(2^3S)G = 2^3R + (2^3c')pk$ holds, if it is then output $b = 1$, otherwise output $b = 0$.

The choice of parameters in EdDSA has a significant impact on security, and the specific parameter selection can be found in [72]. EdDSA is a deterministic algorithm with random numbers that can be derived from a pseudorandom function PRF, thus there is a risk of disclosing the private key $sk$.

In 2020, Feng et al. [73] proposed the first two-party EdDSA scheme, which generates the signature with interactive protocol securely. This paper implements the protocol on mobile devices and gives the security proof in ROM.

In 2021, Bonte et al. [62] proposed a threshold HashEdDSA signature scheme to cope with the high value and potential risks of generating valid signatures in application scenarios such as blockchain. Their contributions are the demonstration of efficient threshold HashEdDSA without modifying the behaviour of the signature algorithm through the use of a doubly-authenticated bit (daBit) generation protocol suitable for $Q_2$ access structures, and the proposed use of a Rescue hash function suitable for MPC to improve performance. On the technical side, the paper cites and optimizes a daBit generation protocol for MPC that allows efficient conversion between binary and large prime fields. Security is based on the active adversary model and the discrete logarithm problem to ensure secret key security. Performance evaluation shows that the use of Rescue hash functions significantly reduces the computational overhead, especially in the case of long messages. Compared with other schemes, this method does not require modification of standard algorithms and is suitable for existing systems.

Feng et al. [74] presented a new multi-party EdDSA signature protocol with stateless and deterministic works in a full threshold setting and can tolerate all but one malicious corruption. The stateless and deterministic generation of random numbers is verified via Multi-Verifier Zero-Knowledge proof (MVZK) by generating and converting Information-Theoretic Message Authenticated Codes (IT-MACs) via PCF and mv-edaBits, and generalized to the multi-verifier case. The stateless and deterministic generation of random numbers is ensured through PCF and mv-edaBits, and their correctness is verified. In the two-party case, this paper stays secure based on the LPN assumption and takes only two rounds as in [63], but the random numbers in [63] are derived using customized functions instead of PCF, and thus cannot be adapted to EdDSA. whereas, in the multi-party case, the new scheme reduces the communication cost by a factor of 56 at the expense of an increase in the computational cost as compared to [60], and is also done in three rounds. The scheme significantly improves efficiency and especially stands out in reducing the cost of communication.

In 2024, Xie et al. [75] proposed an EdDSA-based responsibility threshold signature protocol called TAPS-PR, which addresses the lack of comprehensive implementation of responsibility, privacy, and key protection in blockchain systems in existing research. TAPS-PR utilizes zero-knowledge proof to guarantee the correctness and privacy of update keys and introduces the ElGamal encryption to hide the threshold value and the set of signers, thus improving privacy protection. In addition, the protocol introduces an entity called Tracer that enables liability tracking by using a secret tracking key to track the specific signer that generates the threshold signature in the event of a fraud event. For security purposes, the Tracer verifies the legitimacy of the signature using zero-knowledge proof techniques and ensures that the tracking process does not reveal specific information about the signer. In addition, the paper designs a more efficient protocol, ATS-PR, which reduces the communication and computation overhead. Through experiments and theoretical analysis, the superiority of the proposed protocol in terms of communication and computation overhead is proved, while its compatibility and effectiveness are demonstrated in real applications.

Finally, we give Table 3 to summarise the threshold EdDSA protocols on deterministic property.

## 5 Threshold ECDSA digital signature

The Elliptic Curve Discrete Logarithm Problem (ECDLP)-based threshold Elliptic Curve Digital Signature Algorithm (ECDSA) signature has been receiving a lot of attention lately, mainly because it is the signature scheme supported by the vast majority of cryptocurrencies recently. The details of the ECDSA definition are as follows:

**Definition 11** (ECDSA Digital Signature Scheme). Given a elliptic curve $E(\mathbb{F}_p)$, a base point $G$ over additive cycle group $\mathbb{G}$ with order $q$, and a hash function $\mathsf{H}_e$, the algorithm of ECDSA signature are as follows:

1. $\mathbb{PP} \leftarrow ECDSA.Setup(\kappa)$: On input the security parameter $\kappa$, the algorithm output system public parameters $\mathbb{PP} = (\mathbb{G},G,q)$.
2. $(d,Q) \leftarrow ECDSA.KG(\mathbb{PP})$: Choose a private key $d \leftarrow_R \mathbb{Z}_q$, compute public key $Q = d \cdot G$.
3. $\sigma \leftarrow ECDSA.Sign(d,m)$: For the message $m$, choose a randomness $k \leftarrow_R \mathbb{Z}_q$, compute $R = (r_x,r_y) = k \cdot G$ and $e = \mathsf{H}_e(m)$, then compute part of signature $s = k^{-1}(e + d \cdot r_x)$ and output $\sigma = (r_x,s)$.
4. $b \leftarrow ECDSA.Ver(m,Q,\sigma)$: On input of message $m$, signa-

**Table 3** Summary of threshold EdDSA protocols

| Scheme | PRF | Rounds | Communication | Assumptions |
|---|---|---|---|---|
| GKMN21 [60] | √ | 3 | $O((\lvert C\rvert\lambda + \lvert q\rvert\lambda)n^2)$ | PRF+CRHF |
| KOR23 [63] | × | 1 | $O((m + \lvert q\rvert + \lambda)n^2)$ | DCR+SRSA |
| FYZW24 [74] | √ | 2 | $O((\lvert C\rvert + \log(\lvert C\rvert)\lambda + l\lambda)n^2)$ | LPN |

⋆ $|C|$ denotes the size of PRF and $l$ is the bit length of PRF outputs and $m$ is a large parameter used in [63]

⋆ CRHF denotes the collision-resistant hash function, SRSA denotes the strong RSA assumption

ture $\sigma = (r_x, s)$ and public key $Q$, verifier check the range of signature value. Then verifier computes $R' = s^{-1}(\mathrm{H}_e(m)\cdot G + r_x \cdot Q) = (r'_x, r'_y)$ and output 1 iff $r_x = r'_x$.

Many researches aim to optimize the thresholding of ECDSA, which have different design focuses, technical approaches, and cryptographic techniques. There is a certain degree of technology inheritance, but they also has the characteristics of technology independence. The main design difficulty of ECDSA is the sharing of its random number $k$, the private key $sk$, and the computation during joint signing. We notice that the structure of ECDSA is nonlinear and cannot be directly linearly aggregated. It also needs to collaborate to compute the inverse element of a shared random number in a multi-party environment. How to solve these difficult problems has become the main direction of research.

In 1996, Gennaro et al. proposed GJKR96 [32], which is the first ECDSA threshold signature scheme that does not satisfy the property of threshold optimality. In [32], the private key $sk$ and the signing random number $k$ exist in the form of $n$ secret shares between the two parties, with each $P_i$ holding its own share and computing $k_i \times sk_i$ directly during the signing process to obtain the share of $k \times sk$, which also leads to the secret-sharing polynomial number rises from $t-1$ to $2t-2$. This leads to a threshold $t$ and the number of participants $n$ that need to satisfy $n \geqslant 2t-1$, which is not threshold-optimal.

Based on [32], a series of technical schemes have emerged around the design goals of threshold optimization, efficiency, and various functionalities. Taking the technical way to realize the threshold optimization as the basic classification dimension, these schemes design MtA protocol by several techniques like distributed Homomorphic Encryption (HE), Oblivious Transfer (OT), etc.

In 2015, Gennaro et al. applied threshold ECDSA signatures to Bitcoin wallets in [76], and then proposed the concept of threshold optimality for the first time in [77], and designed the first ECDSA threshold optimal signature algorithm. This paper uses additive secret sharing for keys and random numbers, encrypted by an additive homomorphic encryption algorithm, the signature process is multiplication and inverse operation in the ciphertext state, and finally generates the ciphertext of the signature, and all the parties cooperate to decrypt to get the signed plaintext data. The key step is that all parties generate a set of additive homomorphic encryption algorithm parameters through a trusted third party or a public verifiable algorithm, the encryption key is public, and the decryption private key is distributed in the form of shares in each party. However, the computational overhead is high due to the use of many zero-knowledge proofs and commitments in the interaction process.

The design idea of [78] is similar to that of [77], and its improvement is to replace the encryption algorithm from an additive homomorphic encryption algorithm to a 1-order fully homomorphic encryption algorithm [79], so that any times of addition and multiplication of the encrypted parameter with the depth of 1 can guarantee the homomorphism, thus the number of node interactions can be reduced from six rounds to four rounds.

Although the ECDSA threshold signature scheme based on distributed homomorphic encryption realizes the nature of threshold optimality, there is a very fatal problem, i.e., the achievability/practicality is unknown, because when the number of nodes is more than 2, how to generate the key of a homomorphic encryption algorithm (e.g., Paillier Encryption) in a distributed manner is an uncertain problem.

ECDSA Threshold Signature based on Multiplicative to Additive (MtA) protocol is a mainstream way to realize ECDSA, which has been widely used in practical business scenarios. The core of this scheme is based on the MtA protocol, which realizes the computation process of $k \times sk$, and at the same time, it does not cause the elevation of the number of polynomials of secret sharing, which satisfies the nature of the threshold optimization. Earlier MtA protocol is based on Paillier homomorphic encryption algorithm implementation, the input of the protocol is multiplicative secret shares $a$ and $b$, which satisfies $T = ab$, and the output of the protocol is additive secret shares $\alpha$ and $\beta$, which satisfies $T = \alpha + \beta$. In subsequent works, many different MtA protocols are proposed based on Obvious Transfer (OT), Castagnos-Laguillaumie (CL) [80], etc.

Lindell et al. [81] proposed a two-party ECDSA signature scheme based on the Paillier in 2017, which does not need to execute the distributed Paillier key algorithm while directly utilize Paillier homomorphic attributes to complete two-party collaborative signatures, which improves the operational efficiency of two-party collaborative signatures. The signature scheme in this paper includes a DKG protocol and a distributed signature protocol. The key generation phase is more complex than the signature phase because it is necessary to prove that $P_1$ correctly generates and verifies the Paillier key pair. Moreover, key generation is run only once, so this more costly key generation phase is acceptable.

Gennaro et al. [82] proposed the first ECDSA threshold signature scheme called GG18 based on the MtAwc (MtA with check) protocol. In this paper, they used Feldman's VSS method with the two-party protocol originally proposed by Gilboa et al. [83], which enables the scheme to abort in the presence of a malicious adversary and drastically reduces the computational overhead in the signing process at the cost of increased rounds. In 2020, Gennaro et al. [84] improved GG18 and proposed GG20, which implements IA (Identifiable Abort) on the previous scheme and splits the signature process into the pre-processing phase and online phase, where offline preprocessing is used to perform computations that are not related to the signed message locally, and only one round of interactions is needed to complete the signature online.

In 2018, Lindell et al. [85] also proposed a scheme, LNR18, which is similar to GG18. However, the focus is different, the starting point of LNR18 is to question the practicality of the keygen phase of [77] and [78]. That is, it is difficult to generate a homomorphic encryption algorithm key in a distributed manner. Based on this realization, LNR18 will use an exponential form of the ElGamal algorithm instead of the Paillier homomorphic encryption algorithm in the secret sharing project. This is because the ElGamal algorithm is linear, so its distributed key generation is simple to implement, and it does not directly decrypt the plaintext to get the plaintext, but rather

gets the elliptic curve multiplication point of the plaintext, so as to complete the verification of the correctness of the calculation process.

Castagnos et al. proposed the CCL20 [86] in 2020. This scheme uses additive secret sharing for keys and random numbers, replacing the Paillier homomorphic encryption algorithm used in GG18 with the CL homomorphic encryption algorithm [80], thus avoiding the problem caused by the inconsistency in the order of magnitude of the Paillier modulus and the ECDSA modulus. Although the use of CL encryption solves the computational load of needing to introduce a large number of zero-knowledge proofs to complete range proofs, the efficiency of this scheme is in the same order of magnitude as GG18, since the computational effort of CL is much larger than that of Paillier when the message space is the same.

In 2020, Canetti et al. proposed a new scheme with identifiable abort called CGGMP20 [87]. This scheme focuses on the security guarantee of the computational process by introducing corresponding zero-knowledge proofs at each step, thus ensuring the security of the signature process and avoiding the risk of leaking sensitive information due to signature failure, which is an effective enhancement to the method of verifying the correctness of the signature result in GG18. The solution uses additive secret sharing for keys and random numbers, adds an identifiable abort function, and is resistant to adaptive attacks. The paper also gives two different solutions, i.e., one with four (3+1) rounds of interactions and $O(n^2)$ computational complexity and the other with seven (6+1) rounds of interactions and $O(n)$ computational complexity.

On the other hand, DKLs18 [88] first proposes ECDSA threshold signature schemes based on Correlated Oblivious Transfer (COT) in two-party, which are significantly different from the previous technology paths, but the COT is to implement the MtA functionality. Doerner et al. proposed DKLs20 [89] that improves on DKLs18 and extends it to the multi-party environment, while optimizing the communication and computation process to achieve a 40% performance improvement. The core component of the protocol of DKLs20 is 2-party Multiplication, whose function is similar to that of MtA. The core component of the DKLs20 protocol is 2-party multiplication, which is similar in function to the MtA protocol and is based on the COT implementation. Based on 2-party multiplication, $t$-party inverse-sampling protocol and 2-party multiplier protocol are implemented to complete the inverse operation and multiplication operation respectively, and finally realize the complete signature process.

In 2023, Doerner et al. proposed a new three-round protocol called DKLs23 [90] based on OT, which allows these nonlinear relations to be computed in parallel, thus reducing the number of interaction rounds. This paper proposes a two-round Vector Oblivious Linear Evaluation (VOLE) for implementing the multiplication operations required for ECDSA signatures, ensuring security and privacy during the computation process, and generating the required multiplication results by randomizing the input values, reducing the communication overhead in the specific implementation. The protocol design employs statistical consistency checking and no zero-knowledge proofs to verify the inputs and outputs of the parties to ensure security. The security assumptions are based on the ideal VOLE and commitment primitives and the OT implementation under the stochastic predicate machine model. The concrete implementation requires each participant to perform $6t-2$ scalar operations in the signature phase, and the bandwidth cost is significantly reduced. With this approach, the new protocol ensures malicious security and resistance to malicious attacks while reducing the number of interaction rounds and optimizing the communication/computation overhead, demonstrating a significant improvement in the threshold ECDSA protocol.

In 2021, Kondi et al. [91] proposed an offline refresh protocol for threshold cryptocurrency wallets that allows any $t$ online parties in a threshold wallet to actively refresh in the system at any time, and the remaining offline parties to participate non-interactively at their convenience. This paper focuses on defining the concept of offline refreshing, developing an efficient protocol for the $(2,n)$ threshold signature scheme, and constructing an active multiplication mechanism to refresh the state of the threshold ECDSA protocol, but requiring either Shamir's Secret Sharing (SSS) or additive secret sharing. Security assumptions are based on the mobile attacker model and it is shown that it is impossible to construct offline refresh protocols in the general $(t,n)$ case when there is a dishonest majority of online parties. Experimental results show that the scheme has good performance and security in small-scale decentralized applications, but has some limitations in large-scale systems with higher thresholds.

Abram et al. [92] proposed a new low bandwidth threshold ECDSA scheme. The scheme uses additive secret sharing for keys and random numbers and achieves silent preprocessing by utilizing a pseudo-random correlation generator (PCG) [93] to allow each participant to generate a large number of random numbers locally using seed, which requires only logarithmic-level communication complexity to generate ECDSA signatures, greatly reducing communication and storage requirements. It also employs Learning Parity With Noise (LPN) and Ring LPN assumptions, Distributed Point Functions (DPF), and other techniques to ensure that the system operates efficiently with active security against an arbitrary number of malicious corrupt parties, and realizes non-interactive signatures, which are well suited for use in large-scale financial applications.

Xue et al. [94] proposed an efficient two-party ECDSA signature scheme with additive secret sharing for keys and multiplicative secret sharing for random numbers, and the use of mask $r$ to protect $k_2$ in the random number phase, which reduces one MtA conversion and significantly reduces the online computation and communication overhead. The online phase of the scheme is almost non-interactive and the computational overhead is mainly focused on signature verification. Finally, this paper is optimized and implemented with MtA based on OT, CL, and Paillier.

In 2023, Xue et al. [95] proposed an efficient MtA function based on the modified Joye-Libert (JL) cryptosystem [96] and applied it to a threshold ECDSA signature scheme. The new scheme achieves efficient zero-knowledge proof and encryption operations through the JL cryptosystem and JL-based commitment scheme, and proves the

security of this MtA scheme based on the strong RSA assumption with the k-QR assumption of the quadratic residue assumption under the k-QR assumption JL modulus). Compared with the existing MtA schemes, it has less message space than Paillier-based and less commitment overhead than CL-based. In batch mode, the communication overhead is reduced by a factor of 2.4 to 2.7 and the computational cost is reduced by a factor of 1.62 to 2.26. These technical optimizations give the new solution a significant advantage in application scenarios that require efficient and secure signatures, such as blockchain and distributed systems.

Wong et al. [97] presented a robust scheme that achieves $(t,n)$ threshold flexibility throughout the pre-signing and signing phases with self-healing via VSS-based Distributed Randomness Generation (DRG) and MtA protocol. The MtA protocol is constructed based on the scheme of GG18 [82], which can reduce range proof by replacing the Paillier homomorphic encryption therein using the CL of Castgnos et al. [86]. This scheme requires four rounds of interactions in the pre-signing phase and one round of interactions in the online signing phase, with $O(n)$ computation and communication costs. Compared to CGGMP20 [87], this scheme introduces zero-knowledge proofs and homomorphic encryption while reducing the number of communication rounds and the cost, ensuring the correctness and security of the parties' contributions to the protocol. Even without an honest majority, the scheme can continue the protocol without restarting it through a self-healing mechanism and cheating recognition capability, saving time and computational resources and improving security, robustness and efficiency.

Finally, we give Tables 4 and 5 to summarize some of the schemes in recent years, which contains the attack model of the scheme, the number of signature rounds, the message space size, the computational overhead, the UC functionality, IA functionality, and the security assumptions.

## 6 Threshold pairing-based digital signature

Pairing-based signature schemes utilize the properties of bilinear pairs to achieve efficient and secure signature verification for a wide range of advanced cryptographic applications. The definition of bilinear pairs and their signatures are as follows:

**Table 4** Summary of threshold ECDSA protocols for evaluation

| Scheme | Round | Communication | Computation |
|---|---|---|---|
| GGN16 [77] | 6 | $O(n)$ | $O(n)$ |
| BGG17 [78] | 4 | $O(n)$ | $O(n)$ |
| LNR18 [85] | 7 | $O(n^2)$ | $O(n^2)$ |
| GG18 [82] | 8 | $O(n^2)$ | $O(n^2)$ |
| DKL18 [88] | $\lceil \log t \rceil + 6$ | $O(n^2)$ | $O(n^2)$ |
| CGG20 [87] | 4 or 7 | $O(n^3)/O(n^2)$ | $O(n^3)/O(n^2)$ |
| GG20 [84] | 7 | $O(n^2)$ | $O(n^2)$ |
| CCL20 [86] | 8 | $O(n^2)$ | $O(n^2)$ |
| ANO22 [92] | 2 | $O(n^2)$ | $O(n^2)$ |
| DKL23 [90] | 3 | $O(n^2)$ | $O(n^2)$ |
| WMY23 [97] | 5 | $O(n^2)$ | $O(n)$ |

**Table 5** Summary of threshold ECDSA protocols in functionality

| Scheme | Adaptive | UC | IA | Security assumptions |
|---|---|---|---|---|
| GGN16 [77] | Static | × | × | Strong RSA,DCR |
| BGG17 [78] | Static | × | × | Strong RSA |
| LNR18 [85] | Static | × | × | DDH, DCR |
| GG18 [82] | Static | × | × | DDH, DCR,Strong RSA |
| DKL18 [88] | Static | √ | × | DH |
| CGG20 [87] | Adaptive | √ | √ | DDH, DCR, Strong RSA |
| GG20 [84] | Static | × | √ | DDH, Strong RSA |
| CCL20 [86] | Static | × | × | DDH, CL |
| ANO22 [92] | Static | × | × | Ring-LPN |
| DKL23 [90] | Static | √ | × | DH |
| WMY23 [97] | Static | × | √ | DDH, CL |

⋆ IA means identifiable abort.

⋆ DCR denotes the decisional composite residuosity assumption, DDH denotes the deterministic Diffie-Hellman assumption, OT denotes the black-box oblivious transfer, CL denotes the Castagnos-Laguillaumie cryptography, Ring-LPN denotes the learning parity with noise assumption over ring.

**Definition 12** (Bilinear Pairing). Given elliptic curve $E(\mathbb{F}_p)$ and two $q$-order multiplicative cyclic groups $\mathbb{G}_1, \mathbb{G}_2$ are generated by the weil pairing [98] or the tate pairing [99–101].

The tuple $(\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, q, e)$ can be called a bilinear group if it contains three cyclic groups of order $q$ and a bilinear map

$$e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T.$$

For all $u, v \in \mathbb{Z}_q$ and two generators $g_1$ and $g_2$ of $\mathbb{G}_1$ and $\mathbb{G}_2$, it follows the pairing:

$$e(g_1^u, g_2^v) = e(g_1, g_2)^{uv},$$

then we can define the generator of $\mathbb{G}_T$ as $e(g_1, g_2)$.

Thresholding of pairing-based signatures faces the challenge of high resource consumption as they rely on bilinear pairs. The design of threshold BLS signatures faces the verification problem of signature share and also has to consider dynamic membership management in multi-user scenarios. On the other hand, the design of a threshold identity-based signature scheme needs to consider two challenges of how to distribute master private keys or signatures.

### 6.1 Threshold BLS digital signature

There have been many thresholding studies of BLS signatures in pairing-based signatures. We introduce the definition below:

**Definition 13** (BLS Digital Signature Scheme). Given a non-degenerate pairing $e$ and a Hash function $\mathsf{H}_b$, the algorithms of BLS signature are as follows:

1. $\mathbb{PP} \leftarrow BLS.Setup(\kappa)$: On input the security parameter $\kappa$, the algorithm output system public parameters $\mathbb{PP} = (\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T)$.
2. $(d, Q) \leftarrow BLS.KG(\mathbb{PP})$: Choose a private key $d \leftarrow_R \mathbb{Z}_q$ and compute public key $Q = g_2^d$.
3. $\sigma \leftarrow BLS.Sign(m, d)$: For the message $m$, compute $h = \mathsf{H}_b(m)$ and the signature $\sigma = h^d$.
4. $b \leftarrow BLS.Ver(m, \sigma, Q)$: On input of signature $\sigma$ and public key $Q$, the verifier output $b = 1$ iff $e(\sigma, g_2) = e(\mathsf{H}_b(m), g_2^d)$ holds.

Boldyreva [102] was the first to propose a robust actively secure threshold BLS signature scheme. The scheme is based on the Gap Diffie-Hellman (GDH) group [103], which is simple for DDH and difficult for CDH. This paper uses additive secret sharing for private keys and implements key generation using the DKG protocol of Gennaro et al. [35]. The scheme's features include robustness and proactive security, generating valid signatures even with $t_c < n/2$ malicious participants, and preventing cumulative attacks by periodically updating shares. The signature generation process requires no interaction or zero-knowledge proofs, simplifying the protocol implementation.

Bacho et al. [104] presented adaptive security considerations for threshold BLS signatures. This paper aims to address the security problem in [102] in response to adaptive attacks, i.e., the lack of effective security proofs against adaptive attacks, leading to possible security vulnerabilities in practical applications. To address this problem, the thesis introduces a new DKG security concept, namely oracle-aided algebraic simulatability, which ensures that the key is maintained even if some of the participating nodes are attacked during the key generation process. Specifically, the paper enhances the protection against node corruption by improving the definition of security in the DKG protocol. It proposes a modular security proof method by combining the assumptions of AGM and OMDL for several DKG protocols with a new definition of security. The scheme has low communication and computation overheads in the signature generation and verification process, thus enhancing the efficiency and security of practical applications.

In 2020, Tomescu et al. [105] proposed a fast BLS threshold signature scheme based on the aggregability of BLS signatures. This paper uses polynomial secret sharing for private keys, reduces the aggregation time of BLS threshold signatures from $\Theta(t^2)$ to $\Theta(t\log^2 t)$ by using a fast Lagrange interpolation algorithm, and introduces the Authenticated Multipoint Evaluation Tree (AMT) technique, which reduces the computation time of evaluating proofs of KZG [106] polynomial commitments from $\Theta(nt)$ to $\Theta(n\log t)$, optimizes the VSS and DKG protocols and reducing the sharing phase time to $\Theta(n\log t)$ and the reconfiguration phase time to $O(t\log^2 t + n\log t)$, which significantly improves the efficiency of the system in the case of large-scale participants, despite an increase in the communication and verification time in the case of small-scale participants. The security of the paper is based on the strong bilinear Diffie-Hellman (SBDH) assumption and the polynomial Diffie-Hellman assumption (polyDH).

In 2023, Garg et al. [107] proposed hinTS, which presents a Silent Threshold Signature (STS) that can change the threshold value $t$ based on the message $m$ without changing the public key, and is also able to support the weighted case. This scheme extends from BLS multi-signatures to threshold signatures by letting participants locally compute and send public "hints" in the signer's key during the setup phase, denoting the signing participants by $b = \{0,1\}^n$, proving that the number of participants reaches a threshold using the standard SNARK [108], and performing the private key aggregation by using the method in [109], given by the aggregator's computational correctness with respect to the public key $aPK$, thus avoiding the high overhead DKG protocol. In addition, the scheme supports weighted thresholds and dynamic selection of signers and thresholds to ensure security under the AGM and Common Reference String (CRS) model. In terms of performance, the signing and verification times are 1 ms and 17.5 ms, respectively, and the aggregation time for 1,000 signers is less than 0.5 s. Although the EVM Gas for verifying the signatures in a single pass on the chain is expensive, it can significantly reduce the overall cost and complexity in long-term use.

In 2023, Das et al. [110] proposed a scheme with weights. This paper aims to address the limitations of existing schemes in weighted distribution and multi-threshold environments, which cannot effectively handle signers with different weights and usually support only a single fixed threshold every time, which especially limits their application in decentralized systems such as PoS blockchains. The thought of this paper is similar to [107], extending from multiple signatures with weights to threshold signatures. To solve the inner product problem of $pk$ and $b$, it supports arbitrary weight distributions and multiple thresholds through the use of *inner-product argument* (IPA), and also maintains that the size of the signing and verifying keys is independent of the number of signers. The scheme uses efficient Lagrange polynomial computation and a non-interactive preprocessing method for efficient signature aggregation and verification. In terms of security, the scheme is proved to be secure under the algebraic group model and the random oracle model. Although the initial preprocessing requires $O(n^2)$ computation, the computational overheads of the actual signature and verification processes are constant time and $O(n)$, respectively, for 4096 signers with a signature size of 536 bytes, with a verification time of about 8.21ms, and an aggregation of signatures of 71 ms.

Finally, we give Table 6 to summarize some of the threshold BLS signature schemes in recent years. The table contains the attack model of the schemes, the size of partial and aggregate signatures, the security assumptions, and the contribution of these schemes.

### 6.2 Threshold identity-based digital signature

In addition to BLS signatures, some Identity-Based Signatures (IBS) are also constructed based on bilinear pairs, including the identity-based signatures in IEEE p1363 [111] and the SM9 signatures. To cope with the leakage of master secret key (MSK) in IBS, we briefly introduce the thresholding schemes for these signatures.

The SM9 standard for marking cryptographic algorithms was

**Table 6** Summary of threshold BLS protocols

| Scheme | Adaptive | Partial | Aggregate | Assumptions | Contribution |
|---|---|---|---|---|---|
| Boldyreva [102] | Static | W $G_1$ | 1 $G_1$ | GDH | – |
| BJ22 [104] | Adaptive | 1 $G_1$ | 1 $G_1$ | ROM+AGM+OMDL | New proof |
| hinTS [107] | Adaptive | 1 $G_1$ | 9 $G_1$, 5 $F$ | AGM | STS, proactive |
| DR24 [110] | Adaptive | 1 $G_1$,4 $F$ | 9 $G_1$, 5 $F$ | ROM+DDH+co-DDH | – |

⋆ W is the weight of per party, $G_1$ denote group element and $F$ denote field element

⋆ GDH denotes the Gap Diffie-Hellman, and AGM denotes the Algebraic Group Model

⋆ STS denotes silent threshold signatures

published by the State Cryptography Administration of China (SCA) on March 28, 2016, where the digital signature algorithm section can also be used for thresholding.

**Definition 14** (SM9 Digital Signature Scheme). The system initialization of SM9 digital signature scheme is given two $q$-order multiplicative cyclic groups $\mathbb{G}_1, \mathbb{G}_2$ with generator $g_1, g_2$ and bilinear map $e$, then choose the hash function $\mathsf{H}_1(\cdot)$ and $\mathsf{H}_2(\cdot)$, the algorithms of SM9 signature are as follows:

1. $\mathbb{PP} \leftarrow SM9.Setup(\kappa)$: On input the security parameter $\kappa$, the algorithm outputs system public parameter $\mathbb{PP} = (\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, g_1, g_2, \mathsf{H}_1, \mathsf{H}_2)$.
2. $(d, P_{pub}) \leftarrow SM9.KG(\mathbb{PP})$: The *key generation center* (KGC) choose a random master private key $d \leftarrow_R \mathbb{Z}_q^*$ and compute master public key $P_{pub} = d \cdot g_2$.
3. $d_A \leftarrow SM9.UKG(ID_A, d)$: For the user $A$ identity $ID_A$, KGC compute $h_1 = \mathsf{H}_1(ID_A) + d$, and the private key for user $A$ is $d_A = d \cdot h_1^{-1} g_1$.
4. $(\sigma_1, \sigma_2) \leftarrow SM9.Sign(m, d_A, P_pub)$: On input $d_A$, message $m$ and master public key $P_{pub}$, random choose $r \in \mathbb{Z}_q^*$, compute $\sigma_1 = \mathsf{H}_2(m \cdot e(g_1, P_{pub})^r)$ and $\sigma_2 = (r - \sigma_1) \cdot d_A$.
5. $b \leftarrow SM9.Ver(m, \sigma_1, \sigma_2, P_pub, ID_A)$: On input $P_{pub}$, $ID_A$, $m$ and $\sigma_1, \sigma_2$, the verifier checks whether $\sigma_1 \in \mathbb{Z}_q^*$, $\sigma_2 \in \mathbb{G}_1$, if not, then fail; and compute $h = (\mathsf{H}_1(ID_A) \cdot g_2) + P_{pub}$ to check $\sigma_1 = \mathsf{H}_2(me(\sigma_2, h)e(g_1, P_{pub})^{\sigma_1})$, output $b = 1$ if the equation holds, otherwise the signature is invalid.

In 2018, He et al. [112] disclosed a two-party signature based on SM9. The main idea of the scheme is private key splitting, where KGC generates two secret shares related to the private key and distributes them to two participants. This scheme uses additive homomorphism to compute the second part of the signature and uses zero-knowledge proofs to reduce the risk of data tampering.

In 2019, He et al. [113] disclosed a method for generating SM9 digital signatures jointly by multiple parties in an asymmetric environment. In this method, the user private key is an additive share $s_i$ and $P_1$ serves as the body of the signature, in the process of generating the digital signature, each participant selects a random number $r_i$, computes and broadcasts an intermediate variable $w_i = g^{r_i}$. Subsequently, each participant can generate a joint intermediate variable $w = \sum_{l=1}^{\tau} w_l$. Each participant computes the intermediate variable via the multiplication protocol $\pi_{mul}$ and sends it to $P_1$ so that $P_1$ can compute the second part $S$ of the signature. Similarly, He et al. [114] disclosed a method for SM9 digital signatures by multiple parties in a symmetric setting in 2019.

In 2020, Mu et al. [115] proposed a secure two-party SM9 signature scheme. Their protocol achieves strong protection of private keys by combining Paillier encryption and zero-knowledge proofs. However, the high overhead of the protocol in terms of performance limits its application in certain scenarios. Nevertheless, its enhancements in security and privacy provide a reliable solution for digital currency transactions.

In 2021, Zhang et al. [116] proposed a distributed key generation scheme based on SM9, they pointed out that the generation of a master key in IBC inevitably becomes a security issue that needs to be protected, to solve the single point of failure and the risk of leakage of master key in SM9 system, the authors proposed a $(t,n)$-threshold distributed key generation scheme. The scheme utilizes $\sigma$ and the share transformation algorithm in the *Pseudo-Random Secret Sharing* (PRSS) protocol for master key and user-signed private key generation, where $\sigma$ is a variable generated from the replicated secret sharing protocol for auxiliary key generation. The scheme realizes efficient private key generation with 1 round and meets the IND-ID-CCA security standard while ensuring compatibility with SM9, as well as high efficiency and scalability.

In 2020, Feng et al. [117] proposed a scheme based on IEEE p1363 identity-based signatures. They designed a new multiparty scalar multiplication making $\sum_{i=1}^{n} a_i = (\sum_{i=1}^{n} b_i) \cdot (\sum_{i=1}^{n} c_i)$ to realize distributed signatures. In the same year, He et al. [118] also proposed efficient two-party signatures based on the trusted KGC setup. To address the issues of key escrow and secure channels in identity-based signatures, Feng et al. [119] also proposed a new key generation protocol in 2020. They apply the scalar multiplication $\pi_{mul}$ proposed in the previous paper [117] to compute the intermediate variables and use the random number $r_{ID}$ to compute the blinding factor to participate in the key generation, thus addressing the need for a secure channel during key distribution. Subsequently, they extended the scheme from $(n,n)$ to $(t,n)$ threshold signature scheme based on the idea of DKLs19 [89].

The IBS in IEEE p1363 is based on the BLMQ [120] signature standardization. Jiang et al. [121] proposed a fully distributed threshold BLMQ signature scheme in 2023. This scheme was constructed using VSS-Elliptic Curve (VSS-EC) and VSS-Bilinear Group (VSS-BG) and Paillier encryption to design different distributed protocols for master private key generation, private key extraction and signature generation phases, respectively, with recognizable abort functions. Next year, Jiang et al. [122] proposed a non-interactive protocol that defends against adaptive corruption and enables UC security. This scheme constructs a $F_{mul}$ protocol with four sub-protocols based on the Paillier and Beaver triples, thus constructing the threshold IBS scheme.

## 7 Threshold SM2 digital signature

SM2 is an elliptic curve public key cryptographic algorithm released by the State Cryptography Administration of China (SCA) on December 17, 2010, which became a Chinese cryptographic industry standard (GM/T 0003-2012) in 2012 and a Chinese national cryptographic standard (GB/T 32918-2016) in 2016, of which part 2 is a digital signature algorithm. The details of SM2 digital signature are as follows:

**Definition 15** (SM2 Digital Signature Scheme). The system initialization of SM2 digital signature scheme is given security parameters $\lambda$, choose a finite domain $F_p$, generate an elliptic curve equation $y^2 = x^3 + ax + b \ mod \ p$, the algorithms of SM2 signature are as follows:

1. $\mathbb{PP} \leftarrow SM2.Setup(\kappa)$: On input the security parameter, this algorithm output system public parameters $\mathbb{PP} = (F_p, p, a, b, \mathbb{G}, G, q, \mathsf{H}_1)$, where $G$ is a generator of abelian group $\mathbb{G}$ with order $q$, $\mathsf{H}_1$ is a hash function.
2. $(d, Q) \leftarrow SM2.KG(\mathbb{PP})$: Given system public parameters $\mathbb{PP}$, random select a private key $d \leftarrow_R \mathbb{Z}_q^*$ and compute public key $PK = d \cdot G$.
3. $\sigma \leftarrow SM2.Sign(m, d)$: For the message $m$, compute $e = \mathsf{H}_1(m)$, random choose a nonce $k \leftarrow_R \mathbb{Z}_q^*$, compute $R = (r_x, r_y) = k \cdot G$, $r = (r_x + e) \ mod \ q$ and $s = ((1+d)^{-1} \cdot (k - r \cdot d)) \ mod \ q$ to output the signature $\sigma = (r, s)$.
4. $b \leftarrow SM2.Ver(m, \sigma, Q)$: Given system public parameters $\mathbb{PP}$, public key $PK$, message $m$, signature value $\sigma = (r, s)$, the verifier performs the following steps: if $r, s \notin \mathbb{Z}_q^*$, then the verification fails, calculate the message hash value $e = \mathsf{H}_1(m)$, calculate $t = (r + s) \ mod \ q$, if $t = 0$, then the verification fails, calculate $(r_x, r_y) = s \cdot G + t \cdot P$, calculate $R = (e + r_x) \ mod \ q$. Verify whether the equation $R = r$ holds. If it holds, $\sigma$ is a legal signature and output $b = 1$, otherwise, the signature is invalid.

In 2014, Shang et al. [123] proposed the SM2 threshold signature scheme for the first time, based on the secure multi-party computation method realized by the secret sharing idea. The paper uses Shamir's Secret Sharing (SSS) for both the key and the random number and accomplishes the co-signing operation to guarantee that multiple users securely and collaboratively compute the product and the inverse. This paper also points out that the threshold signature set requires the number of signing participants $(n, t)$ to satisfy $n \geqslant 2t + 1$.

Lin et al. [124] designed an SM2 collaborative signature scheme for cloud computing for two-party distributed application scenarios. The basic idea is to split the SM2 private key into two parts, which are stored in the cloud and the terminal, respectively. The signature operation requires the cloud and the terminal to cooperate to complete, thus avoiding the security risks caused by storing all private keys in the terminal.

In 2020, Zhang et al. [125] proposed a two-party SM2 signature algorithm based on homomorphic encryption and gave provable security guarantees. Still, the signing process needs to use Pailier homomorphic encryption operation, and the execution efficiency of the scheme is low. However, the signature process requires Pailier homomorphic encryption, and the execution efficiency of the scheme is low. Moreover, the scheme requires all the participants to operate together online, and the failure of any one party will lead to the failure of the signature, in some distributed scenarios without trusted centres, such as Bitcoin, Ether, the loss of the private key of any one party will lead to the failure of the execution of the signature. In the same year, Su et al. [126] proposed an efficient provable two-party signature protocol that deforms the random numbers $k$ in the original algorithm to $k = k_1 + d_1 k_2$ to reduce the computation and communication, and proposed an application protocol for enhanced security in practical applications

Feng et al. [127] designed a lightweight SM2 two-party signing scheme based on the idea of joint signing in 2020, which allows the client and the server to complete the SM2 digital signature without disclosing part of their respective signing keys, the process of generating signatures must be participated by both parties at the same time, and there is no recovery of the complete signing key in the process of generating signatures, to guarantee the security of the signing key.

In 2022, Han et al. [128] proposed an efficient two-party SM2 signature protocol based on secret sharing and Beaver multiplication triples. The protocol uses two beaver multiplication triples, significantly improves the signature efficiency by reducing the computation and communication overheads, is 30−40 times faster than existing homomorphic encryption schemes, and proves its security under selective message attacks. This scheme relies on a trusted third party and reduces the rounds to five in the case of pre-shared beaver triples, and the signature efficiency is very close to that of the original SM2 algorithm.

In 2024, Liang et al. [129] proposed a non-interactive SM2 threshold signature scheme with recognizable abort properties, aiming to address the shortcomings of existing schemes in terms of efficiency and security. This scheme uses partial homomorphic encryption to encrypt and share private keys and random numbers. It optimizes Paillier encryption so that the signature generation process only requires the input of a message in the last round, significantly reducing the number of interaction rounds and improving the efficiency of the online signature phase. This scheme also employs

SSS and Verifiable Secret Sharing (VSS) techniques to ensure secure key distribution in the key generation phase and realizes distributed computation of threshold signatures through multiparty additive sharing. In addition, the scheme designs a key update policy that allows dynamic adjustment of the threshold and the number of participants, making it more flexible for application scenarios in blockchain and cryptocurrency wallets. Compared with Canetti's method [87], although the communication overhead in the pre-signature phase grows linearly with the number of participants, this scheme is significantly optimized in terms of communication and computation overheads and still offers better performance and scalability in general.

In the same year, Li et al. [130] proposed a fast two-square scheme using MtA functionality named 2SM2. This paper is constructed based on the re-sharing and MtA protocols, with multiplicative sharing of private keys and additive sharing of random numbers. They design and implement a non-interactive online signing phase and the Paillier-based MtA protocol is required only once in the scheme, improving efficiency. Finally, they proved the security of the scheme under the ROM of a static adversary.

In 2024, Liu et al. [131] organized a two-party SM2 signature summary based on the previous schemes, introduced it based on multiplicative and additive splitting of the private key, included a variety of methods for morphing random numbers and keys, and gave a security and performance analysis. Finally, we give a comparison of multiple two-party SM2 signatures in Table 7.

## 8 Relevant applications

Threshold signatures play important roles in many fields and applications, such as blockchain [77], cloud computing [132], and Internet of Things (IoT) [133]. Threshold ECDSA, one of the earliest schemes is applied in blockchain [76,77,134]. There have been many studies in the literature on the application of threshold ECDSA in blockchain, and its application scenario is mainly to protect the security of blockchain transactions. Different from ECDSA signatures, Schnorr and EdDSA are threshold friendly and their threshold versions require less extra overhead, so now many blockchain systems have started to use EdDSA as a threshold signature algorithm gradually [135,136]. Bilinear pair-based threshold algorithms are favoured in blockchain as they are more efficient in signature aggregation when the number of operators is large [107,110,134]. In addition, the thresholding of IBS can also deal with the problem of MSK (mater secret key) leakage in IoT.

Schnorr signatures can be used for consistency protocols in distributed systems [42], such as consensus mechanisms in distributed ledger technology. The system can coordinate and agree among multiple nodes through threshold signatures and signature aggregation. EdDSA is used in certain security protocols, such as distributed authentication and authorization systems [75], which allow multiple authorized parties to generate valid signatures. It can also be used for collaborative signatures in distributed applications, such as smart contract execution in blockchain systems [73,137], which allows multiple participants to co-sign transactions or contracts.

## 9 Potential directions

In this paper, we have not presented the thresholding of RSA signatures and Post-Quantum (PQ)-based signatures, mainly because there are still many aspects in the development of schemes on these two types of threshold signatures that need to be addressed in the investigation of thresholding.

RSA algorithm is a public key algorithm constructed based on the difficulty of the large number decomposition problem and is one of the most commonly used public key encryption and digital signature algorithms in the world. Earlier researchers have proposed more threshold RSA signature schemes that can achieve active security in different schemes [138], adaptive security [139], etc.

Frederiksen et al. [140] proposed a relatively two-party RSA key generation scheme, however, the scheme requires multiple uses of the Paillier homomorphic encryption algorithm, which imposes high communication and computational overheads, and it cannot be scaled up to a multi-party framework.

**Table 7** Summary of two-party SM2 schemes

| Schemes | Round | Communication | Computation (Offline) | | Computation (Online) | |
|---|---|---|---|---|---|---|
| | | | $P_1$ | $P_2$ | $P_1$ | $P_2$ |
| SM2 | 0 | 0 | 1Mul. | 0 | 0 | 0 |
| Shang [123] | 18 | $6\|p\|+7\|n\|$ | 3Mul. | 3Mul. | 0 | 0 |
| Lin [124] | 1 | $1\|H\|+2\|p\|+3\|n\|$ | 1Mul. | 1Mul. | 0 | 1 |
| Zhang [125] | 1 | $4\|p\|+2\|C\|$ | 1Mul.+1Hom. | 1Mul. | 1Mul.+1Hom. | 3Mul.+2Hom. |
| Han [128] | 5+2 | $4\|p\|+9\|n\|$ | 1Mul. | 1Mul. | 0 | 0 |
| Feng [127] | 2+2 | $1\|H\|+4\|p\|+2\|n\|$ | 1Mul. | 2Mul. | 0 | 0 |
| Su [126] | 2+2 | $4\|p\|+2\|n\|$ | 2Mul. | 2Mul. | 0 | 0 |
| Li [130] | 5+1 | $5\|p\|+2\|n\|$ | 3Mul. | 2Mul. | 0 | 0 |

⋆ Mul is scalar multiplication, and Hom denotes a Parillier encryption or decryption operation.
⋆ $|p|$ and $|n|$ denote the message length of the bits. The $|H|$ is the output of the Hash function, and the $|C|$ is ciphertext of Paillier.

In 2023, Tessaro et al. [141] proposed a scheme based on linear Hash functions that achieves security while relying only on the discrete logarithm assumption and the RSA assumption, and partially realizes non-interactivity. By generalizing the existing FROST [38] and MuSig2 [56] schemes, the authors construct a two-round protocol where the first round of messages is independent of the signature content, greatly improving the practicality. The scheme simplifies the computation by replacing the exponential mapping with a linear mapping via a linear hash function and proves security under the Algebraic One-More Preimage Resistance (AOMPR). In addition, the authors address the difficulties in the computation of Lagrange coefficients and modulo inverse operations in RSA threshold signatures and propose a concrete implementation based on discrete logarithms and RSA under the stochastic predicate machine model.

Overall, the functional design of threshold RSA signatures has been well studied, but researchers in the distributed key generation and security part need further research, such as finding the inverse in the case of modulo non-disclosure

Existing public-key cryptography relies on mathematically intractable problems that are currently considered intractable on conventional computers and require significant computational time. There are already well-developed systems in conventional threshold signatures, and there are many schemes that satisfy different levels of security assumptions and implement many features. However, with the advent of quantum computers, with the help of Shor's algorithm [142], there is a potential to crack the large integer decomposition problems in a shorter period of time. Discrete logarithmic problems and elliptic curve problems are also at risk under the attack of quantum computers.

To cope with quantum computer attacks, researchers have started working on PQ-based signatures. In 2017, NIST began to collect proposals for PQ-based signature schemes, and finally, three schemes [143−145] were selected for digital signatures.

In addition to the schemes standardized by NIST, nowadays anti-quantum signature algorithms can be classified into lattice-based, hash-based, multivariate quadratic-based, and homology-based. Several researchers have successively proposed different methods to instantiate the threshold signature based lattice [146,147]. Next post-quantum threshold signatures will become the focus of future research, and how to ensure the security of transmission in the interaction is also one of the difficulties of research.

Instead, research on EdDSA can focus on how to design high-performance threshold signature schemes that comply with standards, such as high efficiency and low communication. Research on ECDSA threshold signature can focus on designing a weight plus fault tolerance scheme to satisfy scenarios such as board voting, which can be accomplished when the weights are large enough. In addition to that, consider how to remove the bulletin board while implementing IA and proactive functionalities.

## Acknowledgments

This paper was supported by the Major Program (JD) of Hubei Province (No. 2023BAA027), the National Natural Science Foundation of China (Grant Nos. 62202339, 62172307, U21-A20466).

## Competing interests

The authors declare that they have no competing interests or financial conflicts to disclose.

## Open Access



## References

[1] Desmedt Y, Frankel Y. Shared generation of authenticators and signatures. In: Annual International Cryptology Conference. Berlin: Springer, 1991, 457−469

[2] Desmedt Y. Threshold cryptosystems. In: Proceedings of Workshop on the Theory and Application of Cryptographic Techniques. 1993, 1−14

[3] Brandão L T A N, Peralta R. Nist first call for multi-party threshold schemes. Gaithersburg: NIST, 2025

[4] Ergezer S, Kinkelin H, Rezabek F. A survey on threshold signature schemes. Network, See net.in.tum.de/fileadmin/TUM/NET/NET-2020-11-1/NET-2020-11-1_10.pdf website,2020, 49−53

[5] Aumasson J P, Hamelink A, Shlomovits O. A survey of ECDSA threshold signing. IACR Cryptology ePrint Archive, 2020: 1390. See eprint.iacr.org/2020/1390 website, 2020

[6] Sedghighadikolaei K, Yavuz A A. A comprehensive survey of threshold signatures: NIST standards, post-quantum cryptography, exotic techniques, and real-world applications. 2024, arXiv preprint arXiv: 2311.05514

[7] Jing J W, Zhang S C, Wang P J. Threshold cryptography technology and standardization process. Journal of Cryptologic Research, 2024, 11(1): 227−254

[8] ISO. ISO/IEC 14888−3 It security techniques - digital signatures with appendix - Part 3: discrete logarithm based mechanisms. Vernier: International Organization for Standardization, 2018

[9] Yao A C. Protocols for secure computations. In: Proceedings of the 23rd Annual Symposium on Foundations of Computer Science. 1982, 160−164

[10] Tong Y, Feng Q, Luo M, He D. Multi-party privacy-preserving decision tree training with a privileged party. Science China Information Sciences, 2024, 67(8): 182303

[11] Feng D, Yang K. Concretely efficient secure multi-party computation protocols: survey and more. Security and Safety, 2022, 1: 2021001

[12] Saleem H, Ziashahabi A, Naveed M, Avestimehr S. Hawk: accurate and fast privacy-preserving machine learning using secure lookup table computation. 2024, arXiv preprint arXiv: 2403.17296

[13] Hao M, Liu W, Peng L, Li H, Zhang C, Chen H, Zhang T. Unbalanced Circuit-PSI from oblivious Key-Value retrieval. In: Proceedings of the 33rd USENIX Security Symposium. 2024, 6435−6451

[14] Shamir A. How to share a secret. Communications of the ACM, 1979, 22(11): 612−613

[15] Blakley G R. Safeguarding cryptographic keys. In: Proceedings of 1979 International Workshop on Managing Requirements Knowledge. 1979, 313−318

[16] Chor B, Goldwasser S, Micali S, Awerbuch B. Verifiable secret sharing and achieving simultaneity in the presence of faults. In: Proceedings of the 26th Annual Symposium on Foundations of Computer Science. 1985, 383−395

[17] Feldman P. A practical scheme for non-interactive verifiable secret sharing. In: Proceedings of the 28th Annual Symposium on Foundations of Computer Science. 1987, 427−438

[18] Pedersen T P. Non-interactive and information-theoretic secure verifiable secret sharing. In: Feigenbaum J, ed. Advances in Cryptology — CRYPTO ’91. Berlin: Springer, 1992, 129−140

[19] Stadler M. Publicly verifiable secret sharing. In: Proceedings of International Conference on the Theory and Applications of Cryptographic Techniques. 1996, 190−199

[20] Cachin C, Kursawe K, Lysyanskaya A, Strobl R. Asynchronous verifiable secret sharing and proactive cryptosystems. In: Proceedings of the 9th ACM Conference on Computer and Communications Security. 2002, 88−97

[21] Ostrovsky R, Yung M. How to withstand mobile virus attacks (extended abstract). In: Proceedings of the 10th Annual ACM Symposium on Principles of Distributed Computing. 1991, 51−59

[22] Rivest R L, Adleman L, Deaouzos M L. On data banks and privacy homomorphism. In: DeMillo R A, ed. Foundations of Secure Computation. New York: Academic Press, 1978, 169−180

[23] Gentry C. Fully homomorphic encryption using ideal lattices. In: Proceedings of the 41st Annual ACM Symposium on Theory of Computing. 2009, 169−178

[24] Fan J, Vercauteren F. Somewhat practical fully homomorphic encryption. IACR Cryptology ePrint Archive, 2012: 144. See eprint.iacr.org/2012/144 website, 2012

[25] Gentry C, Sahai A, Waters B. Homomorphic encryption from learning with errors: conceptually-simpler, asymptotically-faster, attribute-based. In: Proceedings of the 33rd Annual Cryptology Conference. 2013, 75−92

[26] Cheon J H, Kim A, Kim M, Song Y. Homomorphic encryption for arithmetic of approximate numbers. In: Proceedings of the 23rd International Conference on the Theory and Applications of Cryptology and Information Security. 2017, 409−437

[27] Rabin M O. How to exchange secrets with oblivious transfer. IACR Cryptology ePrint Archive, 2005: 187. Cambridge: Harvard University Technical Report 81,2005

[28] Even S, Goldreich O, Lempel A. A randomized protocol for signing contracts. Communications of the ACM, 1985, 28(6): 637−647

[29] Brassard G, Crepeau C, Robert J M. All-or-nothing disclosure of secrets. In: Odlyzko A M, ed. Advances in Cryptology - CRYPTO ’86. Berlin: Springer, 1987, 234−238

[30] Beaver D. Precomputing oblivious transfer. In: Proceedings of the 15th Annual International Cryptology Conference. 1995, 97−109

[31] Beaver D. Correlated pseudorandomness and the complexity of private computations. In: Proceedings of the 28th Annual ACM Symposium on Theory of Computing. 1996, 479−488

[32] Gennaro R, Jarecki S, Krawczyk H, Rabin T. Robust threshold DSS signatures. In: Proceedings of International Conference on the Theory and Application of Cryptographic Techniques Saragossa. 1996, 354−371

[33] Schnorr C P. Efficient identification and signatures for smart cards. In: Brassard G, ed. Advances in Cryptology - CRYPTO ’89. New York: Springer, 1990, 239–252

[34] Stinson D R, Strobl R. Provably secure distributed Schnorr signatures and a (*t, n*) threshold scheme for implicit certificates. In: Proceedings of the 6th Australasian Conference on Information Security and Privacy. 2001, 417−434

[35] Gennaro R, Jarecki S, Krawczyk H, Rabin T. Secure distributed key generation for discrete-log based cryptosystems. In: Proceedings of the 17th International Conference on Theory and Application of Cryptographic Techniques. 1999, 295−310

[36] Gennaro R, Jarecki S, Krawczyk H, Rabin T. Secure applications of Pedersen’s distributed key generation protocol. In: Proceedings of Cryptographers’ Track at the RSA Conference 2003. 2003, 373−390

[37] Pedersen T P. A threshold cryptosystem without a trusted party. In: Proceedings of Workshop on the Theory and Application of Cryptographic Techniques. 1991, 522−526

[38] Komlo C, Goldberg I. FROST: flexible round-optimized Schnorr threshold signatures. In: Proceedings of the 27th International Conference on Selected Areas in Cryptography. 2021, 34−65

[39] Joshi S, Pandey D, Srinathan K. ATSSIA: asynchronous truly-threshold Schnorr signing for inconsistent availability. In: Proceedings of 24th International Conference on Information Security and Cryptology. 2021, 71−91

[40] Ruffing T, Ronge V, Jin E, Schneider-Bensch J, Schroder D. ROAST: robust asynchronous Schnorr threshold signatures. In: Proceedings of 2022 ACM SIGSAC Conference on Computer and Communications Security. 2022, 2551−2564

[41] González A, Ratoanina H, Salen R, Sharifian S, Soukharev V. Identifiable cheating entity flexible round-optimized Schnorr threshold (ICE FROST) signature protocol. IACR Cryptology ePrint Archive, 2021: 1658. See eprint.iacr.org/2021/1658 website, 2021

[42] Benhamouda F, Halevi S, Krawczyk H, Ma Y, Rabin T. SPRINT: high-throughput robust distributed Schnorr signatures. In: Proceedings of the 43rd Annual International Conference on the Theory and Applications of Cryptographic Techniques. 2024, 62−91

[43] Patra A, Choudhary A, Rangan C P. Efficient statistical asynchronous verifiable secret sharing with optimal resilience. In: Proceedings of the 4th International Conference on Information-

Theoretic Cryptography. 2010, 74−92
[44] Franklin M, Yung M. Communication complexity of secure computation (extended abstract). In: Proceedings of the 24th Annual ACM Symposium on Theory of Computing. 1992, 699−710
[45] Hirt M, Nielsen J B. Robust multiparty computation with linear communication complexity. In: Proceedings of the 26th Annual International Cryptology Conference. 2006, 463−482
[46] Shoup V. The many faces of schnorr. IACR Communications in Cryptology, 2023
[47] Groth J, Shoup V. Fast batched asynchronous distributed key generation. In: Proceedings of the 43rd Annual International Conference on the Theory and Applications of Cryptographic Techniques. 2024, 370−400
[48] Shoup V, Smart N P. Lightweight asynchronous verifiable secret sharing with optimal resilience. Journal of Cryptology, 2024, 37(3): 27
[49] Bellare M, Boldyreva A, Staddon J. Randomness re-use in multi-recipient encryption schemeas. In: Proceedings of the 6th International Workshop on Theory and Practice in Public Key Cryptography. 2002, 85−99
[50] Drijvers M, Edalatnejad K, Ford B, Kiltz E, Loss J, Neven G, Stepanovs I. On the security of two-round multi-signatures. In: Proceedings of 2019 IEEE Symposium on Security and Privacy. 2019, 1084−1101
[51] Bellare M, Crites E, Komlo C, Maller M, Tessaro S, Zhu C. Better than advertised security for non-interactive threshold signatures. In: Proceedings of the 42nd Annual International Cryptology Conference. 2022, 517−550
[52] Bellare M, Tessaro S, Zhu C. Stronger security for non-interactive threshold signatures: BLS and FROST. IACR Cryptology ePrint Archive, 2022:833. See eprint.iacr.org/2022/833 website, 2022
[53] Bauer B, Fuchsbauer G, Plouviez A. The one-more discrete logarithm assumption in the generic group model. In: Proceedings of the 27th International Conference on the Theory and Application of Cryptology and Information Security. 2021, 587−617
[54] Lindell Y. Simple three-round multiparty Schnorr signing with full simulatability. IACR Communications in Cryptology, 2022
[55] Crites E, Komlo C, Maller M. Fully adaptive Schnorr threshold signatures. In: Proceedings of the 43rd Annual International Cryptology Conference. 2023, 678−709
[56] Nick J, Ruffing T, Seurin Y. MuSig2: simple two-round Schnorr multi-signatures. In: Proceedings of the 41st Annual International Cryptology Conference. 2021, 189−221
[57] Bacho R, Loss J, Stern G, Wagner B. HARTS: high-threshold, adaptively secure, and robust threshold Schnorr signatures. In: Proceedings of International Conference on the Theory and Application of Cryptology and Information Security. 2024, 104−140
[58] Abraham I, Jovanovic P, Maller M, Meiklejohn S, Stern G. Bingo: adaptivity and asynchrony in verifiable secret sharing and distributed key generation. In: Proceedings of the 43rd Annual International Cryptology Conference. 2023, 39−70
[59] Bacho R, Loss J, Tessaro S, Wagner B, Zhu C. Twinkle: threshold signatures from DDH with full adaptive security. In: Proceedings of the 43rd Annual International Conference on the Theory and Applications of Cryptographic Techniques. 2024, 429−459
[60] Garillot F, Kondi Y, Mohassel P, Nikolaenko V. Threshold Schnorr with stateless deterministic signing from standard assumptions. In: Proceedings of the 41st Annual International Cryptology Conference. 2021, 127−156
[61] Jawurek M, Kerschbaum F, Orlandi C. Zero-knowledge using garbled circuits: how to prove non-algebraic statements efficiently. In: Proceedings of 2013 ACM SIGSAC Conference on Computer & Communications Security. 2013, 955−966
[62] Bonte C, Smart N P, Tanguy T. Thresholdizing hashEdDSA: MPC to the rescue. International Journal of Information Security, 2021, 20(6): 879−894
[63] Kondi Y, Orlandi C, Roy L. Two-round stateless deterministic two-party Schnorr signatures from pseudorandom correlation functions. In: Proceedings of the 43rd Annual International Cryptology Conference. 2023, 646−677
[64] Roy L. SoftspokenOT: communication-computation tradeoffs in OT extension. IACR Cryptology ePrint Archive, 2022: 192. See eprint.iacr.org/2022/192 website, 2022
[65] Aumann Y, Lindell Y. Security against covert adversaries: efficient protocols for realistic adversaries. Journal of Cryptology, 2010, 23(2): 281−343
[66] Orlandi C, Scholl P, Yakoubov S. The rise of paillier: homomorphic secret sharing and public-key silent OT. In: Proceedings of the 40th Annual International Conference on the Theory and Applications of Cryptographic Techniques. 2021, 678−708
[67] Boneh D, Komlo C. Threshold signatures with private accountability. In: Proceedings of the 42nd Annual International Cryptology Conference. 2022, 551−581
[68] Bünz B, Bootle J, Boneh D, Poelstra A, Wuille P, Maxwell G. Bulletproofs: short proofs for confidential transactions and more. In: Proceedings of 2018 IEEE Symposium on Security and Privacy. 2018, 315−334
[69] Bootle J, Cerulli A, Chaidos P, Groth J, Petit C. Efficient zero-knowledge arguments for arithmetic circuits in the discrete log setting. In: Proceedings of the 35th Annual International Conference on the Theory and Applications of Cryptographic Techniques. 2016, 327−357
[70] Josefsson S, Liusvaara I. Edwards-curve digital signature algorithm (edDSA). See rfc-editor.org/rfc/rfc8032 website, 2017, 1−60
[71] Bernstein D J, Duif N, Lange T, Schwabe P, Yang B Y. High-speed high-security signatures. Journal of Cryptographic Engineering, 2012, 2(2): 77−89
[72] Bernstein D J, Josefsson S, Lange T, Schwabe P, Yang B Y. EdDSA for more curves. IACR Cryptology ePrint Archive, 2015, 2015: 677
[73] Feng Q, He D, Luo M, Li Z, Choo K K R. Practical secure two-party EdDSA signature generation with key protection and applications in cryptocurrency. In: Proceedings of the 2020 IEEE 19th International Conference on Trust, Security and Privacy in Computing and Communications. 2020, 137−147
[74] Feng Q, Yang K, Zhang K, Wang X, Yu Y, Xie X, He D. Stateless deterministic multi-party EdDSA signatures with low communication. In: Proceedings of IACR International Conference on Public-Key Cryptography, 2025
[75] Xie Y, Fan Q, Zhang C, Wu T, Zhou Y, He D, Zhu L. Accountable

and secure threshold EdDSA signature and its applications. IEEE Transactions on Information Forensics and Security, 2024, 19: 7033−7046

[76] Goldfeder S, Gennaro R, Kalodner H, Bonneau J, Kroll J A, Felten E W, Narayanan A. Securing bitcoin wallets via a new DSA/ECDSA threshold signature scheme. See stevengoldfeder.com/papers/threshold_sigs.pdf, 2015

[77] Gennaro R, Goldfeder S, Narayanan A. Threshold-optimal DSA/ECDSA signatures and an application to bitcoin wallet security. In: Proceedings of the 14th International Conference on Applied Cryptography and Network Security. 2016, 156−174

[78] Boneh D, Gennaro R, Goldfeder S. Using level-1 homomorphic encryption to improve threshold DSA signatures for bitcoin wallet security. In: Proceedings of the 5th International Conference on Cryptology and Information Security in Latin America. 2019, 352−377

[79] Catalano D, Fiore D. Using linearly-homomorphic encryption to evaluate degree-2 functions on encrypted data. In: Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security. 2015, 1518−1529

[80] Castagnos G, Laguillaumie F. Linearly homomorphic encryption from DDH. In: Proceedings of Cryptographer's Track at the RSA Conference 2015. 2015, 487−505

[81] Lindell Y. Fast secure two-party ECDSA signing. In: Proceedings of the 37th Annual International Cryptology Conference. 2017, 613−644

[82] Gennaro R, Goldfeder S. Fast multiparty threshold ECDSA with fast trustless setup. In: Proceedings of 2018 ACM SIGSAC Conference on Computer and Communications Security. 2018, 1179−1194

[83] Gilboa N. Two party RSA key generation. In: Proceedings of the 19th Annual International Cryptology Conference. 1999, 116−129

[84] Gennaro R, Goldfeder S. One round threshold ECDSA with identifiable abort. IACR Cryptology ePrint Archive, 2020: 540. See eprint.iacr.org/2020/540 website, 2020

[85] Lindell Y, Nof A. Fast secure multiparty ECDSA with practical distributed key generation and applications to cryptocurrency custody. In: Proceedings of 2018 ACM SIGSAC Conference on Computer and Communications Security. 2018, 1837−1854

[86] Castagnos G, Catalano D, Laguillaumie F, Savasta F, Tucker I. Bandwidth-efficient threshold EC-DSA. In: Proceedings of the 23rd IACR International Conference on Public-Key Cryptography. 2020, 266−296

[87] Canetti R, Gennaro R, Goldfeder S, Makriyannis N, Peled U. UC non-interactive, proactive, threshold ECDSA with identifiable aborts. In: Proceedings of 2020 ACM SIGSAC Conference on Computer and Communications Security. 2020, 1769−1787

[88] Doerner J, Kondi Y, Lee E, Shelat A. Secure two-party threshold ECDSA from ECDSA assumptions. In: Proceedings of 2018 IEEE Symposium on Security and Privacy. 2018, 980−997

[89] Doerner J, Kondi Y, Lee E, Shelat A. Threshold ECDSA from ECDSA assumptions: the multiparty case. In: Proceedings of 2019 IEEE Symposium on Security and Privacy. 2019, 1051−1066

[90] Doerner J, Kondi Y, Lee E, Shelat A. Threshold ECDSA in three rounds. Cryptology In: Proceedings of 2024 IEEE Symposium on Security and Privacy. 2024, 3053−3071

[91] Kondi Y, Magri B, Orlandi C, Shlomovits O. Refresh when you wake up: proactive threshold wallets with offline devices. In: Proceedings of 2021 IEEE Symposium on Security and Privacy. 2021, 608−625

[92] Abram D, Nof A, Orlandi C, Scholl P, Shlomovits O. Low-bandwidth threshold ECDSA via pseudorandom correlation generators. In: Proceedings of 2022 IEEE Symposium on Security and Privacy. 2022, 2554−2572

[93] Boyle E, Couteau G, Gilboa N, Ishai Y, Kohl L, Scholl P. Efficient pseudorandom correlation generators: silent OT extension and more. In: Proceedings of the 39th Annual International Cryptology Conference. 2019, 489−518

[94] Xue H, Au M H, Xie X, Yuen T H, Cui H. Efficient online-friendly two-party ECDSA signature. In: Proceedings of 2021 ACM SIGSAC Conference on Computer and Communications Security. 2021, 558−573

[95] Xue H, Au M H, Liu M, Chan K Y, Cui H, Xie X, Yuen T H, Zhang C. Efficient multiplicative-to-additive function from Joye-Libert cryptosystem and its application to threshold ECDSA. In: Proceedings of 2023 ACM SIGSAC Conference on Computer and Communications Security. 2023, 2974−2988

[96] Joye M, Libert B. Efficient cryptosystems from $2^k$-th power residue symbols. In: Proceedings of the 32nd Annual International Conference on the Theory and Applications of Cryptographic Techniques. 2013, 76−92

[97] Wong H W H, Ma J P K, Yin H H F, Chow S S M. Real threshold ECDSA. In: Proceedings of the 30th Annual Network and Distributed System Security Symposium. 2023

[98] Boneh D, Franklin M. Identity-based encryption from the Weil pairing. In: Proceedings of the 21st Annual International Cryptology Conference. 2001, 213−229

[99] Galbraith S D, Harrison K, Soldera D. Implementing the Tate pairing. In: Proceedings of the 5th International Algorithmic Number Theory Symposium. 2002, 324−337

[100] Scott M. Computing the Tate pairing. In: Proceedings of Cryptographers' Track at the RSA Conference 2005. 2005, 293−304

[101] Arène C, Lange T, Naehrig M, Ritzenthaler C. Faster computation of the Tate pairing. Journal of Number Theory, 2011, 131(5): 842−857

[102] Boldyreva A. Threshold signatures, multisignatures and blind signatures based on the gap-diffie-Hellman-group signature scheme. In: Proceedings of the 6th International Workshop on Theory and Practice in Public Key Cryptography. 2002, 31−46

[103] Joux A. A one round protocol for tripartite Diffie–Hellman. In: Proceedings of the 4th International Symposium. 2000, 385−393

[104] Bacho R, Loss J. On the adaptive security of the threshold BLS signature scheme. In: Proceedings of 2022 ACM SIGSAC Conference on Computer and Communications Security. 2022, 193−207

[105] Tomescu A, Chen R, Zheng Y, Abraham I, Pinkas B, Gueta G G, Devadas S. Towards scalable threshold cryptosystems. In: Proceedings of 2020 IEEE Symposium on Security and Privacy. 2020, 877−893

[106] Kate A, Zaverucha G M, Goldberg I. Constant-size commitments to polynomials and their applications. In: Proceedings of the 16th International Conference on the Theory and Application of Cryptology and Information Security. 2010, 177−194

[107] Garg S, Jain A, Mukherjee P, Sinha R, Wang M, Zhang Y. hinTS: threshold signatures with silent setup. In: Proceedings of 2024 IEEE Symposium on Security and Privacy. 2024, 3034−3052

[108] Gabizon A, Williamson Z J, Ciobotaru O. PLONK: permutations over Lagrange-bases for oecumenical noninteractive arguments of knowledge. IACR Cryptology ePrint Archive, 2019: 953. See eprint.iacr.org/2019/953 website, 2019

[109] Ràfols C, Zapico A. An algebraic framework for universal and updatable SNARKs. In: Proceedings of the 41st Annual International Cryptology Conference. 2021, 774−804

[110] Das S, Camacho P, Xiang Z, Nieto J, Bünz B, Ren L. Threshold signatures from inner product argument: succinct, weighted, and multi-threshold. In: Proceedings of 2023 ACM SIGSAC Conference on Computer and Communications Security. 2023, 356−370

[111] IEEE. 1363.3-2013 IEEE standard for identity-based cryptographic techniques using pairings. Piscataway: IEEE, 2013, 1−151

[112] He D, Zhang Y. SM9 digital signature generation method and system: CN107579819B. 2019–11-19

[113] He D, Feng Q, Wang J, Zhou X. Method for multi-party combined generation of SM9 digital signature in asymmetric environment: CN109194478B. 2021–12-07

[114] He D, Feng Q, Wang J, Lin C, Zhang Y. Method for generating SM9 digital signature through multi-party association under symmetrical environment: CN109660361B. 2020–11-24

[115] Mu Y, Xu H, Li P, Ma T. Secure two-party SM9 signing. Science China Information Sciences, 2020, 63(8): 189101

[116] Zhang R, Zou H, Zhang C, Xiao Y, Tao Y. Distributed key generation for SM9-based systems. In: Proceedings of the 16th International Conference on Information Security and Cryptology. 2021, 113−129

[117] Feng Q, He D, Liu Z, Wang D, Choo K K R. Distributed signing protocol for IEEE P1363-compliant identity-based signature scheme. IET Information Security, 2020, 14(4): 443−451

[118] He D, Zhang Y, Wang D, Choo K K R. Secure and efficient two-party signing protocol for the identity-based signature scheme in the IEEE P1363 standard for public key cryptography. IEEE Transactions on Dependable and Secure Computing, 2020, 17(5): 1124−1132

[119] Feng Q, He D, Wang H, Wang D, Huang X. Multi-party key generation protocol for the identity-based signature scheme in the IEEE P1363 standard for public key cryptography. IET Information Security, 2020, 14(6): 724−732

[120] Barreto P S L M, Libert B, McCullagh N, Quisquater J J. Efficient and provably-secure identity-based signatures and signcryption from bilinear maps. In: Proceedings of the 11th International Conference on the Theory and Application of Cryptology and Information Security. 2005, 515−532

[121] Jiang Y, Zhu Y, Wang J, Li X. Fully distributed identity-based threshold signatures with identifiable aborts. Frontiers of Computer Science, 2023, 17(5): 175813

[122] Jiang Y, Zhu Y, Wang J, Zhang Y. Efficient online and non-interactive threshold signatures with identifiable aborts for identity-based signatures in the IEEE P1363 standard. IACR Cryptology ePrint Archive, 2024: 1333. See eprint.iacr.org/2024/1333 website, 2024

[123] Shang M, Ma Y, Lin J Q, Jing J W. A threshold scheme for SM2 elliptic curve cryptographic algorithm. Journal of Cryptologic Research, 2014, 1(2): 155−166

[124] Lin J, Ma Y, Jing J, Wang Q, Lei L, Cai Q, Wang L. Signing and decrypting method and system applied to cloud computing and based on SM2 algorithm: CN104243456A. 2014–12–24

[125] Zhang Y, He D, Zhang M, Choo K K R. A provable-secure and practical two-party distributed signing protocol for SM2 signature algorithm. Frontiers of Computer Science, 2020, 14(3): 143803

[126] Su Y X, Tian H B. A two-party SM2 signing protocol and its application. Chinese Journal of Computers, 2020, 43(4): 701−710

[127] Feng Q, He D, Luo M, Li L. Efficient two-party SM2 signing protocol for mobile internet. Journal of Computer Research and Development, 2020, 57(10): 2136−2146

[128] Han G, Bai X, Geng S, Qin B. Efficient two-party SM2 signing protocol based on secret sharing. Journal of Systems Architecture, 2022, 132: 102738

[129] Liang H, Chen J. Non-interactive SM2 threshold signature scheme with identifiable abort. Frontiers of Computer Science, 2024, 18(1): 181802

[130] Li S, Yang W, Zhang F, Huang X, Chen R. Practical two-party SM2 signing using multiplicative-to-additive functionality. Computer Standards & Interfaces, 2025, 92: 103928

[131] Liu Z Y, Lin J Q. Framework of two-party threshold schemes for SM2 digital signatures. Journal of Software, 2024, doi: 10.13328/j.cnki.jos.006978

[132] Chen L, Guo C, Gong B, Waqas M, Deng L, Qin H. A secure cross-domain authentication scheme based on threshold signature for MEC. Journal of Cloud Computing, 2024, 13(1): 70

[133] Yang A, Weng J, Yang K, Huang C, Shen X. Delegating authentication to edge: a decentralized authentication architecture for vehicular networks. IEEE Transactions on Intelligent Transportation Systems, 2022, 23(2): 1284−1298

[134] Dziembowski S, Jarecki S, Kedzior P, Krawczyk H, Ngo C N, Xu J. Password-protected threshold signatures. In: Proceedings of the 30th International Conference on the Theory and Application of Cryptology and Information Security. 2024, 174−206

[135] Maxwell G, Poelstra A, Seurin Y, Wuille P. Simple Schnorr multi-signatures with applications to bitcoin. Designs, Codes and Cryptography, 2019, 87(9): 2139−2164

[136] Shi Y, Liang J, Li M, Ma T, Ye G, Li J, Zhao Q. Threshold EdDSA signature for blockchain-based decentralized finance applications. In: Proceedings of the 25th International Symposium on Research in Attacks, Intrusions and Defenses. 2022, 129−142

[137] Feng Q, Yang K, Ma M, He D. Efficient multi-party EdDSA signature with identifiable aborts and its applications to blockchain. IEEE Transactions on Information Forensics and Security, 2023, 18: 1937−1950

[138] Rabin T. A simplified approach to threshold and proactive RSA. In: Proceedings of the 18th Annual International Cryptology Conference. 1998, 89−104

[139] Almansa J F, Damgård I, Nielsen J B. Simplified threshold RSA with adaptive and proactive security. In: Proceedings of the 25th International Conference on the Theory and Applications of Cryptographic Techniques. 2006, 593−611

[140] Frederiksen T K, Lindell Y, Osheter V, Pinkas B. Fast distributed RSA key generation for semi-honest and malicious adversaries. In: Proceedings of the 38th Annual International Cryptology Conference.

2018, 331−361

[141] Tessaro S, Zhu C. Threshold and multi-signature schemes from linear hash functions. In: Proceedings of the 42nd Annual International Conference on the Theory and Applications of Cryptographic Techniques. 2023, 628−658

[142] Shor P W. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. SIAM Review, 1999, 41(2): 303−332

[143] Fouque P A, Hoffstein J, Kirchner P, Lyubashevsky V, Pornin T, Prest T, Ricosset T, Seiler G, Whyte W, Zhang Z. Falcon: fast-Fourier lattice-based compact signatures over NTRU. Submission to the NIST's Post-Quantum Cryptography Standardization Process, 2018, 36(5): 1−75

[144] Ducas L, Lepoint T, Lyubashevsky V, Schwabe P, Seiler G, Stehle D. CRYSTALS–dilithium: digital signatures from module lattices. IACR Cryptology ePrint Archive, 2017: 633. See eprint.iacr.org/2017/633 website,2017

[145] Aumasson J P, Bernstein D J, Beullens W, Dobraunig C, Eichlseder M, Fluhrer S, Gazdag S L, Hülsing A, Kampanakis P, Kölbl S, Lange T, Lauridsen M M, Mendel F, Niederhagen R, Rechberger C, Rijneveld J, Schwabe P, Westerbaan B. SPHINCS+. Submission to the NIST's Post-Quantum Cryptography Standardization Process, See sphincs.org/data/sphincs+-round3-specification.pdf website, 2020

[146] Espitau T, Katsumata S, Takemure K. Two-round threshold signature from algebraic one-more learning with errors. In: Proceedings of the 44th Annual International Cryptology Conference. 2024, 387−424

[147] Espitau T, Niot G, Prest T. *Flood and submerse*: distributed key generation and robust threshold signature from lattices. In: Proceedings of the 44th Annual International Cryptology Conference. 2024, 425−458

Yu PENG received his MSc degree in Artificial Intelligence and Adaptive Systems from Sussex Artificial Intelligence Institute, Zhejiang Gongshang University, Hangzhou, China in 2024. He is currently pursuing the PhD degree with the Key Laboratory of Aerospace Information Security and Trusted Computing, Ministry of Education, School of Cyber Science and Engineering, Wuhan University, Wuhan 430072, China. His research interests includes secure multiparty computation and cryptographic protocols.

Qi FENG received her PhD degree in cyberspace security from School of Cyber Science and Engineering, Wuhan University, China in 2021. She is currently an associate researcher of the Key Laboratory of Aerospace Information Security and Trusted Computing, Ministry of Education, School of Cyber Science and Engineering, Wuhan University, China. Her research interests include cryptographic protocols.

De-Biao HE received his PhD degree in aplied mathematics from School of Mathematics and Statistics, Wuhan University, China in 2009. He is currently a professor of the School of Cyber Science and Engineering, Wuhan University, China. His main research interests include cryptography and information security, in particular, cryptographic protocols. He has published over 150 research papers in refereed international journals and conferences, such as IEEE Transactions on Dependable and Secure Computing, IEEE Transactions on Information Security and Forensic, and Usenix Security Symposium. He is the recipient of the 2018 IEEE Systems Journal Best Paper Award and the 2019 IET Information Security Best Paper Award. His work has been cited more than 7000 times at Google Scholar. He is in the Editorial Board of several international journals, such as Journal of Information Security and Applications, Frontiers of Computer Science, and Human-centric Computing & Information Sciences.

Min LUO received his PhD degree in computer science from Wuhan University, China in 2003. He is currently a professor at the School of Cyber Science and Engineering, Wuhan University, China. He has published papers in international conferences/journals, such as S&P, ACM TRETS, IEEE SYST J, and IEEE TVT. His research interests mainly include applied cryptography and blockchain technology.