symmetry

MDPI

Article

A Novel Undeniable (t, n)-Threshold Signature with Cheater Identification

Yi-Fan Tseng *,† **and Yan-Bin Lin** †

Department of Computer Science, National Chenchi University, Taipei 11605, Taiwan; 109753111@nccu.edu.tw
* Correspondence: yftseng@cs.nccu.edu.tw
† These authors contributed equally to this work.

Abstract: A digital signature is one of the most widely used cryptographic primitives in asymmetry cryptography. According to the security requirements in different symmetry or asymmetry network models, various digital signatures have been developed in the literature. To protect the right of the signer, Chaum and Antrepen first introduced the concept of an undeniable signature, where interactive protocols are needed for the verification process. Besides, a signer can, also, perform a disavowal protocol to prove that they did not sign the message. On the other hand, threshold cryptography is, usually, used to protect the system from a single point of failure. In a (t, n)-threshold signature scheme, as long as t people in the group of n people participate, the signature can be smoothly signed. By combining these two features, an undeniable threshold signature enjoys the advantages from both sides. After our survey, we found that the existing undeniable threshold signature schemes are either insecure or apply impractical assumptions. Thus, in this manuscript, we aim at designing a novel and provably secure undeniable threshold signature scheme. The proposed scheme is formally proven to be unforgeable and invisible. Besides, our scheme supports cheater identification, which allows one to find the cheater, when a signing protocol fails. Moreover, the proposed scheme can be performed without the help of trusted third parties or secure cryptographic modules, which would be more practical when our scheme is deployed in real-world applications.

Keywords: digital signature; undeniable signature; threshold signature; cryptanalysis

Citation: Tseng, Y.-F.; Lin, Y.-B. A Novel Undeniable (t, n)-Threshold Signature with Cheater Identification. *Symmetry* **2022**, *14*, 1118. https://doi.org/10.3390/sym14061118

Academic Editors: Jose Carlos R. Alcantud and Sergei D. Odintsov

Received: 6 May 2022
Accepted: 27 May 2022
Published: 29 May 2022

1. Introduction

A digital signature is widely used, nowadays. It is a mathematical technique used to validate the authenticity and integrity of a message. Unlike a handwritten signature, it is easily copied and distributed. According to the security requirements in different symmetry or asymmetry network models, various digital signatures have been developed in the literature. For example, the signature scheme used in cryptocurrency and centralized digital currency should be different, due the requirements of the environments, even though they are digital currency as well. Though these properties are convenient, they are unsuitable for some cases. Consider a commitment that is sensitive to some extent, personally or commercially, so one would only want to commit to the party they specified, but not to others. For example, someone may want to break the news of some incriminating information on the Internet. To protect themself, they may only want to sign for a reporter or judge, not for everyone. In such cases, undeniable signatures are well suited. An undeniable signature is a digital signature that allows a signer to be selective to whom they allow to verify their signature. Without the signer's cooperation, the signature could not be verified. One may consider a designated verifier signature [1–5] for such a scenario, which allows an interactive portion of the scheme to be offloaded onto a designated verifier for each signature, reducing the burden on the signer. However, in a designated verifier signature scheme, a signer has to designate a verifier in the signing phase, which would be inconvenient. In the example mentioned above, since the signer could not know which

reporter is interested in the news or which judge would like to verify the news at first, they cannot designate a verifier beforehand.

The first undeniable signature was proposed by Chaum and Antwerpen in 1989 [6]. In an undeniable signature, there are two interaction protocols: confirmation protocol and disavowal protocol. Confirmation protocol confirms that a candidate is a valid signature of the message issued by the signer. However, the signer can always refuse to take part in confirmation protocol and claim that the signature is invalid, at anytime. As the result, disavowal protocol, which confirms that a candidate is not a valid signature of the message issued by the signer, is an important part of an undeniable signature. Such a protocol is not available for a designated verifier signature. Later, in 1990, Chaum proposed a zero-knowledge undeniable signature scheme [7]. Lots of research [8–12] following Chaum's work has been proposed. However, not until 2005 were the unforgeability and invisibility for Chaum's scheme proven by Ogata et al. [13]. More precisely, it was the security of the full-domain hash (FDH) variant of Chaum's scheme that Ogata et al. proved. Due to the special properties of an undeniable signature, a lot of related works [14–18] have been given in the literature.

Though an undeniable signature has been found useful in many scenarios, it can be further improved. The original construction for an undeniable signature is a single-signer paradigm. However, a theft or loss of the signing key can be a catastrophic problem. The most common way for preventing this issue is storing secrets in multiple locations, which prevents the capture of the secrets and the, subsequent, cryptanalysis of the system. Nevertheless, just splitting and combining the secrete is unstable for a corporation because of the single point of failure problem. The key should never be stored at a single location, throughout its entire lifetime. In such cases, threshold cryptosystem [19] and threshold signatures [20–23], in particular, are well suited. In a threshold signature scheme, signing keys are distributed among several servers, which need to act jointly to issue a signature. A threshold signature may fit Bitcoin or blockchain-based applications [24], due to its distributed architecture. In a threshold signature scheme, n parties are allowed to share the signing ability under a single public key. A parameter t is defined, such that an adversary that compromises t or fewer shares is unable to generate a signature and learns no information about the key. Besides, such an approach is able to better fit into a decentralized environment. The first threshold undeniable signature was proposed by Harn and Yang in 1992 [25]. They proposed two threshold undeniable signature schemes: $(1, n)$ scheme and (n, n) scheme. Following Harn and Yang's [25] schemes, Lin et al. [26] presented a general (t, n) threshold undeniable signature scheme. In 1998, Lin and Wu [27] pointed out that their scheme suffers from a cheating attack by a dishonest signer, which can result in a fake group signature. A scheme should identify the evil party that sends out the invalid value in the interaction protocol. As a result, Lin and Wu [27] proposed an undeniable (t, n)-threshold signature scheme, supporting cheater identification. Their scheme does not need to trust any third party, and provides two capabilities to withstand cheating attacks. Each member's secret key corresponds to a public key, which can be verified in the key-generation phase. Besides, their scheme allows one to identify any dishonest participant in the group-signature-generation phase. Nevertheless, Lin and Wu's scheme has the following problem:

1. Although Lin and Wu claimed that no trusted third party is required for key generation, a secure cryptographic module is required in the group-signature-generation phase.
2. There is a vulnerability: a dishonest signer still can forge a fake signature that the secure cryptographic module is unable to detect, which can result in a fake group signature. The corresponding cryptanalysis has been shown in [28], by Lin and Tseng in 2021.
3. There is no security proof in Lin and Wu's paper.

1.1. Contribution

After our survey, we found that there is, probably, no secure threshold undeniable signature with cheater identification. Therefore, in this manuscript, we design an undeniable (t, n)-threshold signature scheme with the following features.

1. Standing from a forge attack from a dishonest signer by identification.
2. Our scheme does not need any trusted third party or secure cryptographic module in any phase.
3. We, formally, demonstrate the security proofs of the unforgeability and invisibility for our scheme.

1.2. Organization

The rest of this manuscript is organized as follows. In Section 2, we present the preliminaries for the proposed scheme. In Section 3, we demonstrate the details of our scheme. In Section 4, we define and prove the security of our scheme. Next, we compare our scheme with others [27,29–33] in Section 5. Finally, in Section 6 we discuss opportunities for future work and conclude this work.

2. Preliminaries

We, first, define some notations in Table 1 for our manuscript. We use pk, sk, and tk to represent public key, secret key, and trapdoor key, respectively. By m, we mean a message used in a signature scheme or a commitment scheme. For the output (C, D) of a commitment scheme **Com**, C and D are the commitment string and the decommitment string, respectively. In our undeniable threshold signature scheme, n and t are used for the number of users and the threshold value for signing a message, respectively. For an integer q, Z_q is the set $\{0, 1, \ldots, q-1\}$. Besides, by Prob$[E]$, we mean the probability that event E happens.

Table 1. Notations.

Notation	Meaning
pk	public key
sk	secret key
tk	trapdoor key
m	message
(C, D)	commitment string/decommitmenet string
n	number of users
t	threshold value
Z_q	$\{0, 1, \ldots, q-1\}$
Prob$[E]$	the probability that event E happens

2.1. Non-Malleable Equivocable Commitments

A trapdoor commitment scheme, or equivocable commitments [34], is a commitment scheme with a special property, called "equivocable". As a commitment scheme, a trapdoor commitment scheme should be hiding and binding. The former requires a commitment string that reveals no information about the committed message, while the latter guarantees the committed message cannot be altered after commitment. Moreover, the "equivocable" property allows a sender to open a commitment string in any possible way using the trapdoor. A trapdoor commitment scheme consists of the following algorithms: **KG**, **Com**, **Ver**, **Equiv**.

- **KG**(1^λ). On inputting the security parameter 1^λ, the key generation algorithm **KG** outputs a public/trapdoor key pair (pk, tk).
- **Com**(pk, m). On inputting the public key pk and a message m, the commitment algorithm **Com** outputs $[C(m), D(m)] =$ **Com** $(\text{pk}, m; R)$, where R is the random

coin used in the algorithm. Here, $C(m)$ is the commitment string, and $D(m)$ is the decommitment string, which should be kept secret before opening.

- **Ver**(pk, C, D). On inputting the public key pk and the commitment/decommitment string C, D, the verification algorithm **Ver**, also known as the open algorithm, either outputs a message m or an invalid symbol $\perp$.
- **Equiv**$(\mathrm{pk}, m, R, m', \mathrm{tk})$ is the algorithm that realizes the equivocable property. It takes as inputs the public key pk, strings m, R for $[C, D] = \mathrm{Com}(\mathrm{pk}, m, R)$, a message $m' \neq m$, and the trapdoor key tk, and outputs D' such that $\mathrm{Ver}(\mathrm{pk}, C, D') = m'$.

A trapdoor commitment scheme should satisfy the following properties:

- **Correctness**: If $[C, D] = \mathrm{Com}(\mathrm{pk}, m, R)$ then $\mathrm{Ver}(\mathrm{pk}, C, D) = m$.
- **Hiding**: For every message pair m, m' the distributions $C(m)$ and $C(m')$ are statistically close.
- **Binding**: There is no probabilistic polynomial-time algorithm $\mathcal{A}$ that is able to output C, D, D', such that $\mathrm{Ver}(\mathrm{pk}, C, D) = m, \mathrm{Ver}(\mathrm{pk}, C, D') = m'$ and $m \neq m'$.

A commitment scheme is said to be non-malleable [35], if there is no adversary intercepting $C(m)$, so it is able to compute $C(m')$ for a related message m'. We refer the readers to [36–40], for more details about non-malleable commitments.

2.2. Threshold Signatures

A threshold signature is a multi-signer digital signature scheme. Let $\mathcal{S}$ = (Key-Gen, Sig, Ver) a signature scheme. A (t, n)-threshold signature scheme $\mathcal{T}$ for $\mathcal{S}$ realizes the functionality to distribute the signing key among a group of n players, $P_1, \ldots, P_n$. To sign on a message, there must be at least $t + 1$ players to jointly perform a multiparty protocol in order to generate a valid signature. A threshold signature scheme $\mathcal{T}$ consists of two protocols:

- **Thresh-Key-Gen**(1^λ), the distributed key-generation protocol. Taking as input the security parameter 1^λ, in this protocol, n players jointly compute the public key pk and private keys sk_i for player P_i, for $i = 1, \ldots, n$.
- **Thresh-Sig**, the distributed signing protocol, which takes as a public input a message m to be signed, and private inputs $\{\mathrm{sk}_i\}_{i \in I}$ for $I \subseteq \{1, \ldots, n\}$ and $|I| \geq t + 1$. The output of the protocol is a signature σ.

Note that, to verify the signature σ on message m with public key pk, one can run the Ver algorithm of $\mathcal{S}$.

2.3. The FDH Variant of Chaum's Undeniable Signature Scheme

The full-domain hash (FDH) variant of Chaum's scheme is, briefly, described as follows. Let G be a commutative group of prime order q, and with a generator g.

- Key Generation. On inputting the security parameter 1^λ, the algorithm outputs the public key $\mathrm{pk} = (g, y, H)$ and the secret key $\mathrm{sk} = x$, where x is randomly chosen from Z_q, $y = g^x$, $H : \{0,1\}^* \to G$ is a cryptographic hash function.
- Signing. Taking as inputs the public key $\mathrm{pk} = (g, y, H)$, the secret key $\mathrm{sk} = x$, and a message $m \in \{0,1\}^*$, the algorithm outputs the signature as $\sigma = H(m)^x$.

There are two protocols, the confirmation protocol and the disavowal protocol. The former allows one to prove that she/he is indeed the signer, while the later is used to prove that one is *not* the signer. Here, we omit the description for the confirmation protocol and disavowal protocol, since they are unnecessary for reading our manuscript. As one would observe in Section 3, the signature of our scheme is of the same form of that of Chaum's scheme. Therefore, in the security proofs shown in Section 4, we will prove the security of our scheme based on the security of Chaum's scheme. Besides, as we mentioned in Section 1, Chaum's scheme has been proven unforgeable and invisible by Ogata et al. [13]. For more details, the readers are referred to [13].

2.4. Cheater Identification

In this manuscript, we follow the definition of secure multi-party computation with an identifiable abort defined in [41], which allows the computation to fail (abort), while guaranteeing that all the honest parties agree on the identity P_i of a cheater.

If F is the functionality computed by the original MPC protocol, then a protocol for F with identifiable aborts, computes a modified functionality F' that either computes F or outputs the identity P_i of a cheater, in case of an abort.

2.5. Undeniable (t, n)-Threshold Signature with Cheater Identification

The Undeniable (t, n)-Threshold Signature with Cheater Identification is described as follows:

- **Key Generation**. A verifiable (t, n)-threshold secret sharing [42] for a secret x consists of n shares $x_1, \ldots, x_n$ distributed to n parties. Any party can check the share they get during the phase. If the check does not hold, the protocol will abort and the cheater will be identified.
- **Signature Generation**. An undeniable signature that t parties cooperate, generates a valid signature that does not reveal any information of the party's share. If any party sends out an invalid value that causes the signature to be invalid, the protocol will abort and the cheater will be identified.
- **Confirmation Protocol**. Given a message-signature pair, the group of t parties cooperates, to prove the validity of the signature that does not reveal any information about the party's share. If any party sends out an invalid value that causes the proof to fail, the protocol will abort, and the cheater will be identified.
- **Disavowal Protocol**. Given a message-signature pair, the group of t parties cooperates, to prove the invalidity of the signature that does not reveal any information of the party's share. If any party sends out an invalid value that causes the proof to fail, the protocol will abort and the cheater will be identified.

3. The Proposal Scheme

The most difference of our scheme and Lin et al.'s scheme is that it is not necessary for the usage of secure cryptographic module, which can be viewed as an online trusted third party. We show the details of our scheme as follows:

Initialization Phase: Let G be a group of signers $\{U_1, U_2, \ldots, U_n\}$, where each signer U_i is identified by a unique string ID_i. The following public parameters are defined by the group before other phases. Let p, q be two large primes, such that $p = 2q + 1$. For the multiplicative group of the field $GF(p)$, let g be a generator of order q.

Key Generation Phase: The key generation process is described as follows

Step 1: Each U_i selects a random value $x_i \in Z_q$, computes $y_i = g^{x_i} \in GF(p)$ and $[C_i, D_i] = Com(y_i)$, then broadcasts C_i.

Step 2: Each U_i broadcasts D_i. Let y_i be the value decommitted by U_i.

Step 3: Each U_i randomly generates a $t-1$ degree polynomial.

$$f_i(x) = a_{i0} + a_{i1}x + a_{i2}x^2 + \ldots + a_{i,t-1}x^{t-1} \tag{1}$$

where each $a_{ij} \in Z_q$ and $a_{i0} = x_i$. Then, each U_i computes

$$v_{ij} = g^{a_{ij}} \bmod p \tag{2}$$

for $j = 0, \ldots, t-1$, and publishes $V_i = [v_{i0}, v_{i1}, \ldots, v_{i,t-i}]$. Note that $v_{i0} = y_i$.

Step 4: Each U_i in G computes a shadow key.

$$u_{ij} = f_i(ID_j) \bmod q \tag{3}$$

for each $U_j \in G \backslash \{U_i\}$. Next, u_{ij} is sent to U_j via a secure channel. Note that U_j is able to check the validity of u_{ij} by checking

$$g^{u_{ij}} \stackrel{?}{=} \prod_{k=0}^{t-1}((v_{ik})^{(ID_j)^k}) (\mathrm{mod}\, p). \tag{4}$$

U_i will be identified as a cheater, if Equation (4) does not hold.

Step 5: After receiving $n-1$ shadow keys from the others, the secret key u_i of each U_i in G can be computed by U_i as:

$$u_i = \sum_{U_k \in G} u_{ki} \bmod q. \tag{5}$$

Note that U_i's share public key n_i can be computed as:

$$n_i = g^{u_i} = \prod_{\ell=0}^{t-1}\prod_{k=0}^{t-1}((v_{\ell k})^{(ID_i)^k}) \bmod p$$

Once all signers's secret keys are verified, the public key for the signer group G is computed as $y = \prod_{U_i \in G} v_{i0} \bmod p$, and the secret key will be the form $x = \sum_{U_i \in G} a_{i0} \bmod p$.

Signature Generation Phase: Any group $W \subseteq G$ of t signers are able to jointly sign on a message M, via the following protocol:

Step 1: Each $U_i \in W$ computes their partial signature $T_i = M^{u_i}$, computes $[C_i, D_i] = Com(T_i)$, and broadcasts C_i.

Step 2: Each U_i broadcasts D_i. Let S_i be the value decommitted by U_i. Note that each U_i can prove to others that they know u_i s.t. $S_i = M^{u_i}, n_i = g^{u_i}$ using Chaum's [7] zero-knowledge protocol, by replacing Z with S_i, x with u_i, and y with n_i, respectively.
If all S_i are accepted, the players compute:

$$Z = \prod_{U_i \in W} (S_i^{\prod_{U_j \in W \backslash U_i} -ID_j \cdot (ID_i - ID_j)^{-1}}) \bmod p,$$

which will be the form of M^x.

Confirmation Protocol: Assume that a group $W' \subseteq G$ of any t signers, who are willing to help a verifier to verify, the signature is Z. Then, the following zero-knowledge protocol will be run jointly, by the members in W' and the verifier.

Step 1: The verifier sends $\left\{D = \left(h(M)^\alpha \cdot g^\beta\right\}\right.$ to W', where α, β are randomly chosen from Z_q, and h is a collision-resistant hash function.

Step 2: W' performs Step 1 to Step 2 of the **Signature Generation Phase**, whereas the message M is replaced with D. Let δ be the result after Step 2 of the **Signature Generation Phase**. Note that, $\delta = D^x \bmod p$. Next, W' randomly chooses r and sends $R = \delta^r \pmod p$ to the verifier.

Step 3: The verifier then sends (α, β) to W'.

Step 4: W' then verify $D \stackrel{?}{=} \left(h(M)^\alpha \cdot g^\beta\right) \pmod p$. If the equation holds, then W' reveals r to the verifier; otherwise the protocol is terminated.

Step 5: Finally, the verifier checks

$$R \stackrel{?}{=} \left(Z^a \cdot Y^\beta\right)^r.$$

If the equation holds, then the verifier accepts the signature Z; otherwise, it is rejected.

Disavowal Protocal: To convince a verifier that a specific value $Z \neq M^x$, for a given message M and public key $y = g^x$, a group of signer W' of any t members in G perform the following protocol with the verifier.

Step 1: The verifier chooses, uniformly at random, an integer s from $\{0, 1, \ldots, k\}$, where k should be mutually agreed. Besides, the verifier randomly chooses $a \in Z_q$. Then, the verifier sends $\{D = M^s g^a, E = Z^s y^a\}$ to the group W'.

Step 2: W' performs Step 1 to Step 2 of the **Signature Generation Phase**, whereas the message M is replaced with D. Let δ be the result after Step 2 of the **Signature Generation Phase**, $\delta = (M^s g^a)^x = M^{sx} y^a$.

Step 3: The group W' chooses $\zeta \in \{0, 1, \ldots, k\}$ and tests if

$$\frac{E}{\delta} = (Z \cdot M^{-x})^{\zeta}. \tag{6}$$

The equality of Equation (6) means that ζ is equal to the value s chosen by the verifier. Since there are at most $k + 1$ choices of s, W' is able to find the correct ζ, with at most $k + 1$ trials. Note that if Z, indeed, is equal to M^x, then W' can only guess the correct $\zeta \in \{0, 1, \ldots, k\}$ with probability $1/(k + 1)$, since "$Z = M^x$" implies $Z \cdot M^{-x} = 1$. Next, W' uses a commitment scheme to commit the correct ζ, which is the one that makes Equation (6) hold, and sends the commitment C of ζ to the verifier.

Step 4: The verifier then sends a to W'.

Step 5: If $D = M^{\zeta} g^a$, then W' sends ζ and the decommitment string D to the verifier.

Step 6: The verifier accepts if C opens ζ via D and $\zeta = s$.

4. Security Proof

There are two security notions defined by Chaumin his paper [7]: unforgeability and invisibility [13]. Unforgeability guarantees that an adversary cannot forge a valid signature pair without the private key. Invisibility guarantees that one cannot determine whether a given signature/message pair is valid. There is another security notion, called anonymity, defined by Galbraith and Mao [43], which (roughly) states that it is difficult to determine the real signer, if there are two or more signers. It has been shown that, if the message space are the same for all the signer groups, then invisibility and anonymity are equivalent. In this section, we will prove the unforgeability and invisibility for our undeniable threshold signature scheme. We start the proof with a technical overview below.

Our scheme is based on a full-domain hash (FDH) variant of Chaum's scheme, which has been proven to be unforgeable [44]. The form of the signature generated at the end of the **Signature Generation Phase** is of the same form as that of Chaum's scheme. The concept of our proof is that, if there exists an adversary $\mathcal{A}$ being able to forge a valid signature of scheme with a probability $\epsilon \geq \lambda^{-c}$, then we can build a forger $\mathcal{F}$ to forge a valid signature for Chaum's undeniable signature scheme, also with probability $\epsilon \geq \lambda^{-c}$. We assume that $\mathcal{F}$ can simulate the signing oracle and the confirmation/disavowal oracle. These assumption is the same for an invisibility distinguisher $\mathcal{D}$.

Without loss of generality, we may assume that the adversary controls players $U_2, \ldots, U_t$ and that U_1 is the honest player who always speaks first at each round. We, then, build a simulator acting on behalf of U_1, to simulate the protocol without knowledge of U_1's private key, to interact with the $U_2, \ldots U_n$ that the adversary controlled.

By $\mathcal{A}(\tau_{\mathcal{A}})_{U_1(\tau_1)}$, where $\tau_{\mathcal{A}}$ is the random tape of $\mathcal{A}$, and τ_1 is the random tape of U_1, we denote the output of $\mathcal{A}$. Assume that

$$\text{Prob}_{\tau_1, \tau_{\mathcal{A}}} \left[\mathcal{A}(\tau_{\mathcal{A}})_{U_1(\tau_1)} \text{ is a successful forgery } \right] \geq \epsilon.$$

for some value ϵ. As stated in [44], if we rewind $\mathcal{A}$, we still have:

$$\text{Prob}_{\tau_1}\left[\mathcal{A}(\tau_{\mathcal{A}})_{P_1(\tau_1)} \text{ is a successful forgery }\right] \geq \frac{\epsilon}{2}.$$

This fact will help us to make the simulation sound using trapdoor commitments. If the simulation is indistinguishable from the real protocol from the view of $\mathcal{A}$, the adversary will act as if it is in the real protocol, and output the corresponding result. Next, we describe the details of the simulation as follows:

4.1. Unforgeability

Consider the following game.

1. First, $\mathcal{A}$ participates in the key generation protocol to jointly compute a public key y for the undeniable threshold signature scheme.
2. Next, $\mathcal{A}$ is allowed to request the group of players to sign on messages $M_1, \ldots, M_\ell$, and the signer group jointly performs the signing protocol to generate the corresponding signatures $Z_1, \ldots, Z_\ell$.
3. Besides, $\mathcal{A}$ is also allowed to send message-signature pairs $(M_1, Z_1), \ldots, (M_\ell, Z_\ell)$, and the signer group performs the confirmation/disavowal protocol with the pairs.
4. At the end, the adversary outputs a message-signature pair (M^*, Z^*) for it, under the public key y, with probability of at least ϵ.

Definition 1 (Unforgeability)**.** *An undeniable threshold signature scheme achieves the existential unforgeability under the adaptive chosen message attacks, if ϵ is negligible.*

We state the proof for the unforgeability as follows. In our proof, we aim at build a forger $\mathcal{F}$ for Chaum's undeniable signature scheme, who will simulate the view for an adversary $\mathcal{A}$ to our scheme, according to the unforgeability game. Note that, as an attacker to Chaum's signature scheme, $\mathcal{F}$ is given a public key y and allowed to access the signature/confirmation/disavowal oracles of Chaum's signature scheme. An overview is illustrated in Figure 1.

4.1.1. Simulation for the Key Generation Phase

The simulation is described below. On input, a group public key $y = g^x$ for Chaum's undeniable signature scheme, the forger $\mathcal{F}$ plays the role of U_1, as follows:

1. $\mathcal{F}$ selects a random value $x_1 \in Z_q$, and computes $[C_1, D_1] = Com(g^{x_1})$ and broadcasts C_1. Meanwhile, $\mathcal{A}$ broadcasts commitments C_i for $i = 2, \ldots, t$.
2. Each player U_i broadcasts D_i. Let y_i be the value decommitted from C_i using D_i.
3. $\mathcal{F}$ then rewinds $\mathcal{A}$ to Step 2. Now, $\mathcal{F}$ can compute a new $\hat{D}_1$, such that, C_1 is decommitted to $\hat{y}_1 = y \cdot \prod_{i=2}^{n} y_i^{-1}$ using $\hat{D}_1$. This can be done by adopting the **Equiv** algorithm of the trapdoor commitment scheme.
4. Next, $\mathcal{F}$ chooses $u_{12}, \ldots, u_{1t} \in Z_q$ and sets a polynomial $f_1(x) = a_0 + a_1 x + \ldots + a_{t-1}x^{t-1}$, satisfying $f_1(ID_i) = u_{1i}$ and $\hat{y}_1 \cdot \prod_{k=1}^{t-1} g^{a_k (ID)^k} = g^{u_{1i}}$ for $i = 2, \ldots, t$.
5. $\mathcal{F}$ simulates the remaining parts of the protocol, following the procedure of the **Key Generation Phase** shown in Section 3.
6. At the end of the simulation of this phase, a public key $\hat{y}$ is outputted.

In the simulation, the view of the adversary $\mathcal{A}$ is the same as that in the real scheme, thanks to the equivocable property of the trapdoor commitments. Besides, one can see that $v_{i0} = y_i$ for $i = 2, \ldots, t$, and $v_{10} = \hat{y}_1 = y \cdot \prod_{i=2}^{n} y_i^{-1}$. Therefore, the public key outputted at the end of the simulation is:

$$\hat{y} = \prod_{i=1}^{t} v_{i0} = (y \cdot \prod_{i=2}^{n} y_i^{-1}) \cdot y_2 \ldots y_t = y,$$

which is actually the given public key of Chaum's signature scheme.

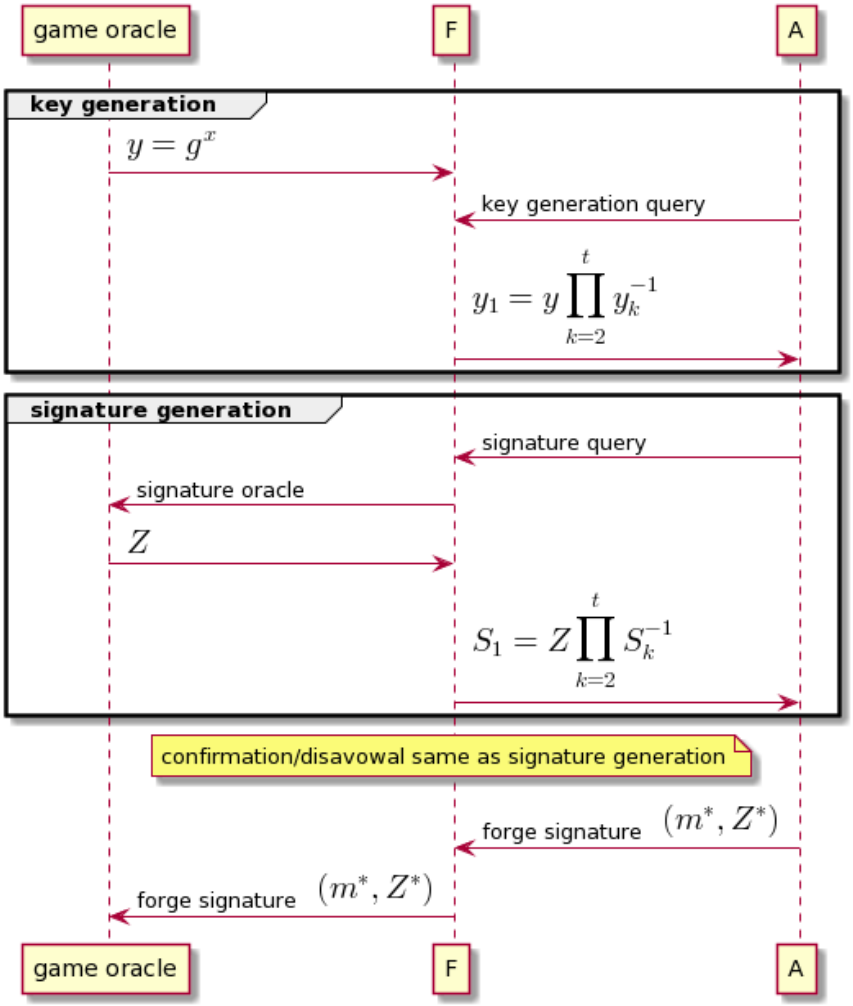

Figure 1. Simulation of proving unforgeability.

4.1.2. Simulation for the Signature Generation Phase

$\mathcal{F}$ simulates this phase as follows, with $\mathcal{A}$ for generating a signature of a message M.

1. $\mathcal{F}$ selects a random value $x_1 \in Z_q$, computes $[C_1, D_1] = Com(M^{x_1})$ and broadcasts C_1. Meanwhile, $\mathcal{A}$ broadcasts commitments C_i for $i = 2, \ldots, t$.
2. Next, each player U_i broadcasts D_i. Let S_i be the result decommitted from C_i using D_i.
3. $\mathcal{F}$ queries the signature oracle with M, to obtain a signature Z, and, then, rewinds $\mathcal{A}$ to Step 2. Now, $\mathcal{F}$ is able to compute a new $\hat{D}_1$, such that, $\hat{D}_1$ decommits C_1 to

$$\hat{S}_1 = \left[Z \cdot \left(\prod_{i=2}^{t} S_i^{\prod_{j \in \{1,\ldots,t\}/\{j\}} \frac{-ID_j}{(ID_i - ID_j)}} \right)^{-1} \right]^{\prod_{j=2}^{t} \frac{ID_1 - ID_j}{-ID_j}}.$$

4. At the end of the simulation, a signature $\hat{Z}$ is outputted.

Thanks again to the equivocable commitment scheme, the view in the simulation if the same as that in a real protocol to the adversary. Besides, one can observe that

$$\hat{Z} = \hat{S}_1^{\prod_{j=2}^{t} \frac{-ID_j}{ID_1 - ID_j}} \cdot \prod_{i=2}^{t} S_i^{\prod_{j \in \{1,\ldots,t\}/\{j\}} \frac{-ID_j}{(ID_i - ID_j)}} = Z.$$

Besides, $Z = M^x$, where $x = \log_g y$, since it is a valid signature of Chaum's signature scheme. Therefore, the signature $\hat{Z}$ generated in this phase is, also, a valid signature of our scheme, since the form of a signature in our scheme is the same as that in Chaum's scheme.

4.1.3. Simulating the Confirmation/Disavowal Protocol

Since $\mathcal{F}$ is allowed to query to confirmation/disavowal oracles, and $\mathcal{F}$ can successfully simulate the signature generation phase, $\mathcal{F}$ is able to simulate confirmation/disavowal protocol for our scheme, by just following the procedures shown in Section 3.

Lemma 1. *Our scheme is unforgeable, if the full-domain hash (FDH) variant of Chaum's scheme is unforgeable.*

Proof. As stated in Sections 4.1.1–4.1.3, the view simulated by the forger $\mathcal{F}$ is indistinguishable from a real protocol, for the adversary $\mathcal{A}$. Therefore, $\mathcal{A}$ will output a valid forgery with the same probability, say ϵ, as that against a real protocol. However, due to the usage of the rewinding technique, the probability for $\mathcal{F}$ succeeding in the key-generation phase and the signature-generation phase is at least $\frac{\epsilon}{2}$. Thus, the probability for $\mathcal{F}$ obtaining a forgery from $\mathcal{A}$ is at least $\frac{\epsilon^2}{4}$. Note that the signature of our scheme is of the same form of Chaum's scheme. If $\mathcal{A}$ outputs a successful forgery, then $\mathcal{F}$ can output what $\mathcal{A}$ outputs, to break the unforgeability of Chaum's scheme. Under the assumption that Chaum's scheme is unforgeable, the probability of success of $\mathcal{F}$ must be negligible, which implies that $\mathcal{A}$ can forge the scheme with the probability ϵ, which must, also, be negligible. □

4.2. Invisibility

Consider the following game:

1. First, $\mathcal{A}$ participates in the key generation protocol to jointly compute a public key y, for the undeniable threshold signature scheme.
2. Next, $\mathcal{A}$ is allowed to request the group of players to sign on messages $M_1, \ldots, M_\ell$, and the signer group jointly performs the signing protocol, to generate the corresponding signatures $Z_1, \ldots, Z_\ell$.
3. Besides, $\mathcal{A}$ is, also, allowed to send message-signature pairs $(M_1, Z_1), \ldots, (M_\ell, Z_\ell)$, and the signer group performs the confirmation/disavowal protocol with the pairs.
4. At some point, $\mathcal{A}$ outputs a message M^* that has never been queried before and is given challenge signature Z^*. The generation of Z^* follows the rules below. First, a coin toss b hidden from $\mathcal{A}$'s view is determined. Second, Z^* is a valid signature on M^*, if $b = 1$; Z^* is uniformly chosen from the signature space at random if $b = 0$.
5. $\mathcal{A}$ keeps making queries as before, except when:
   - making a signing query with M^*;
   - making a confirmation/disavowal query with (M^*, Z^*).
6. At the end, the adversary outputs a guess b', such that $b' = b$, with probability at least ϵ.

Definition 2 (Invisibility)**.** *An undeniable threshold signature scheme achieves invisibility under an adaptive chosen message attack, if $\epsilon - \frac{1}{2}$ is negligible.*

Note that the term $\epsilon - \frac{1}{2}$ is usually defined as the *advantage* for an adversary $\mathcal{A}$, in winning the invisibility game.

We state the proof for the invisibility as follows. In our proof, we aim at build a distinguisher $\mathcal{D}$ for Chaum's undeniable signature scheme, which will simulate the view for an adversary $\mathcal{A}$ to our scheme, according to the invisibility game. Note that, as an attacker to Chaum's signature scheme, $\mathcal{F}$ is given a public key y and allowed to access the signature/confirmation/disavowal oracles of Chaum's signature scheme. An overview of the proof is illustrated in Figure 2.

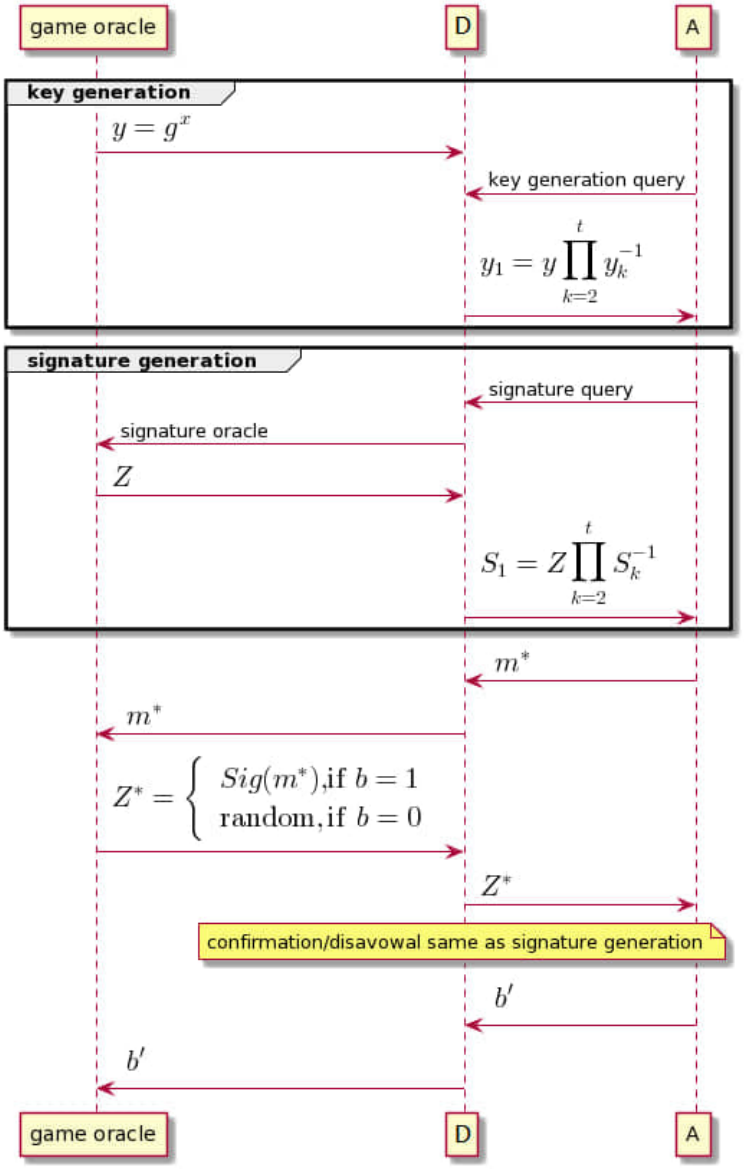

Figure 2. Simulation of proving invisibility security.

Simulation

The simulation is as follows:

1. To simulate the key generation protocol, the distinguisher $\mathcal{D}$ plays the role of P_1 and does the same to simulate as the forger $\mathcal{F}$ does in the unforgeability proof, shown in Section 4.1.1.
2. To simulate the signature generation protocol, the distinguisher $\mathcal{D}$ plays the role of P_1 and does the same to simulate as the forger $\mathcal{F}$ does in the unforgeability proof, shown in Section 4.1.2.
3. To simulate the confirmation/disavowal protocol, the distinguisher $\mathcal{D}$ plays the role of P_1 and does the same to simulate as the forger $\mathcal{F}$ does in the unforgeability proof, shown in Section 4.1.3.
4. At some point, the adversary outputs a message M^* to the distinguisher $\mathcal{D}$. Then, $\mathcal{D}$ forwards M^* to the oracle of Chaum's invisibility game, to obtain a signature Z^*. Finally Z^* is sent to $\mathcal{A}$.

Lemma 2. *Our scheme satisfies invisibility if full-domain hash (FDH) variant of Chaum's scheme is invisibility.*

Proof. Since the view simulated by the distinguisher $\mathcal{D}$ is indistinguishable from a real protocol for the adversary $\mathcal{A}$, $\mathcal{A}$ is able to distinguish Z^* with the same probability, say ϵ, as that in a real protocol. However, due to the usage of the rewinding technique, the probability for $\mathcal{F}$ succeeding in the key generation phase and the signature generation phase is at least $\frac{\epsilon}{2}$. Thus, the probability for $\mathcal{F}$ obtaining a forgery from $\mathcal{A}$ is at least $\frac{\epsilon^2}{4}$. As we mentioned in the proof of Lemma 1, since the signature of our scheme is of the same form of Chaum's scheme, we make $\mathcal{D}$ output what $\mathcal{A}$ outputs. Obviously, if $\mathcal{A}$ is able to tell whether Z^* is a valid signature or a random element from the signature space, then $\mathcal{D}$ is able to break the invisibility of Chaum's scheme. Thus, if Chaum's scheme satisfies the invisibility, the advantage for $\mathcal{D}$ succeeding in the invisibility game must be negligible, which implies that $\mathcal{A}$ can distinguish the signature with the advantage that must, also, be negligible. □

5. Comparison and Analysis

In this section, we compare our scheme with undeniable threshold signature schemes [27,29–32], an undeniable signature scheme [14], and a threshold signature scheme, supporting designated verifier [33], for their security properties in Table 2. We, also, make a comparison on computation cost, where the corresponding result is shown in Table 3.

5.1. Security Properties

In this section, we will compare the scheme with the following properties:

- **Share distribution center.** The scheme does not need a trusted third party or secure cryptographic module.
- **Security proof.** Security proof for unforgeability and invisibility are provided.
- **Cheater identification.** The scheme can detect cheaters in the signing and key distributed phase.
- **Avoid single point of failure.** The scheme would not suffer from a single point of failure problem, which means just one malfunction or fault of a participator would not cause the whole phase to stop working.

Table 2 lists the results of the comparison between our schemes and their security properties. Most schemes need a trust third party. Liu et al.'s scheme [33] needs a signature combination. Although Wang and Qing's scheme [30], also, satisfies the share-distribution-center property, it suffers from a single point of failure problem. Only [32,33] give the security proof for the unforgeability, however, they do not prove the invisibility. For the lattice-based undeniable signature scheme proposed by Rawal et al. [14], the unforgeability and invisibility are proven. However, Rawal et al.'s construction supports only a single-signer version, not a threshold version, and, thus, cannot avoid a single point of failure. Besides, their scheme does not support cheater identification, since only the single-signer scenario is considered. One can see that, in Table 2, ours is the only one satisfying all the required properties.

Table 2. Comparison of security properties with other schemes.

Schemes	Share Distribution	Security Proof	Cheater Identification	Avoid SPOF [1]
[27]	no	no	no [2]	yes
[29]	no	no	yes	yes
[30]	yes	no	yes	no
[31]	no	no	no	yes
[32]	no	unforgeability	yes	yes
[33]	key generation phase	unforgeability	yes	yes
[14]	yes	yes	no	no
Ours	yes	yes	yes	yes

[1] Single Point of Failure, SPOF. [2] Crack shown in [28].

5.2. Computational Complexity

In this section, we will evaluate the computational complexity of [14,27,29–33] and ours, in the following phases.

- **Individual signature.** The time for computing exponent generated by each individual participants.
- **Group signature.** The time for computing the exponent of the group-signing generation.
- **Confirmation protocol.** The time for computing the exponent of the confirmation protocol.
- **Disavowal protoco.l**: The time for computing the exponent of the confirmation protocol.

Table 3 lists the results of the comparison between our schemes and others. We assume that each participant only needs to run the proof protocol one time to show they are not a cheater. Let t be the least number of members that are needed for the protocol, T_m represents the time for computing multiply in the elliptic curve, T_e represents the modular exponentiation operation, and ℓ denotes the worst time for the group to guess the correct random number, selected by the verifier in the disavowal protocol. Besides, the computation costs for the pre-image sampleable function *SamplePre* and matrix multiplication are denoted by T_{PSF} and $T_{\mathbf{M}}$, respectively. One can observe that our scheme does not perform well. It might be a trade-off between security and efficiency, since our scheme provides more security properties than other works, e.g., invisibility. In the future, we will try our best to improve the performance.

Table 3. Comparison of computational complexity with other schemes.

Schemes	Individual Signature	Group Signature	Confirmation Protocol	Disavowal Protocol
[27]	$(1+2t)T_e$	$2tT_e$	$(9+8t)T_e$	no
[29]	$7tT_e$	tT_e	$(4+8t)T_e$	$(4+4t\ell)T_e$
[30]	$9tT_e$	tT_e	$(4+9t)T_e$	$2(4+9t)T_e$
[31]	tT_e	–	$(6+4t)T_e$	$(4+4t+\ell)T_e$
[32]	$10tT_e$	–	$(6+23t)T_e$	$(4+21t+\ell)T_e$
[33]	$4tT_m$	–	T_e	no
[14]	$T_{PSF}+T_{\mathbf{M}}$	–	$3T_{\mathbf{M}}$	$3T_{\mathbf{M}}$
Ours	$9tT_e$	tT_e	$(7+10t)T_e$	$(6+10t+2\ell)T_e$

6. Conclusions

Undeniable threshold signature can protect the right of signers and prevent the single point of failure at the same time. It would, also, be suitable for decentralized environments, due to its non-centralized architecture. In this manuscript, we present a new undeniable (t,n)-threshold-signature scheme, supporting cheater identification. In our scheme, there is no need for a trusted third party or a secure cryptographic module. The unforgeability and invisibility of our scheme have, also, been formally proven. Compared to other related works, though the efficiency is not the best, our scheme is the only one that achieves decentralization, cheater identification, unforgeability, and invisibility, simultaneously.

In our scheme, to identify the cheater, the step to verify cannot be avoided. The property of the share-distribution center and to avoid a single point of failure, also, increase the computation overhead of verification. As a result, the performance of our scheme, shown in Section 5.2, is not well. Our scheme adopts the zero-knowledge proof skill in Chaum's scheme [7], which is the first undeniable signature scheme using the zero-knowledge-proof technique. Using another more efficient zero-knowledge protocol in the confirmation/disavowal protocol may improve the performance of our scheme. Another

direction for improvement could be to design a quantum-resistant construction, e.g., lattice-based construction.

Author Contributions: Conceptualization, Y.-B.L. and Y.-F.T.; methodology, Y.-B.L. and Y.-F.T.; validation, Y.-F.T.; investigation, Y.-B.L.; writing–original draft preparation and editing, Y.-B.L.; writing–review, Y.-F.T.; visualization, Y.-B.L.; supervision, Y.-F.T. All authors have read and agreed to the published version of the manuscript.

Funding: This work was partially supported by the Ministry of Science and Technology of Taiwan, under grants MOST 110-2221-E-004 -003 -, MOST 110-2218-E-004-001-MBK, MOST 109-2221-E-004 -011 -MY3, and MOST 110-2622-8-004-001 -.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: I am really really grateful to my advisor Yi-Fan Tseng, for advising me and introducing the project to me, which has helped me complete my project. I am also dearly obliged to Raylin Tso and Zi-Yuan Liu, for advising me in writing this manuscript.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Jakobsson, M.; Sako, K.; Impagliazzo, R. Designated verifier proofs and their applications. In Proceedings of the International Conference on the Theory and Applications of Cryptographic Techniques, Saragossa, Spain, 12–16 May 1996; Springer: Berlin/Heidelberg, Germany, 1996; pp. 143–154.
2. Li, Y.; Susilo, W.; Mu, Y.; Pei, D. Designated verifier signature: Definition, framework and new constructions. In Proceedings of the International Conference on Ubiquitous Intelligence and Computing, Hong Kong, China, 11–13 July 2007; Springer: Berlin/Heidelberg, Germany, 2007; pp. 1191–1200.
3. Lipmaa, H.; Wang, G.; Bao, F. Designated verifier signature schemes: Attacks, new security notions and a new construction. In Proceedings of the International Colloquium on Automata, Languages, and Programming, Lisbon, Portugal, 11–15 July 2005; Springer: Berlin/Heidelberg, Germany, 2005; pp. 459–471.
4. Steinfeld, R.; Bull, L.; Wang, H.; Pieprzyk, J. Universal designated-verifier signatures. In Proceedings of the International Conference on the Theory and Application of Cryptology and Information Security, Taipei, Taiwan, 30 November–4 December 2003; Springer: Berlin/Heidelberg, Germany, 2003; pp. 523–542.
5. Saeednia, S.; Kremer, S.; Markowitch, O. An efficient strong designated verifier signature scheme. In Proceedings of the International Conference on Information Security and Cryptology, Taipei, Taiwan, 30 November–4 December 2003; Springer: Berlin/Heidelberg, Germany, 2003; pp. 40–54.
6. Chaum, D.; Van Antwerpen, H. Undeniable signatures. In Proceedings of the Conference on the Theory and Application of Cryptology, Santa Barbara, CA, USA, 20–24 August 1989; Springer: New York, NY, USA, 1989; pp. 212–216.
7. Chaum, D. Zero-knowledge undeniable signatures. In Proceedings of the Workshop on the Theory and Application of of Cryptographic Techniques, Aarhus, Denmark, 21–24 May 1990; Springer: Berlin/Heidelberg, Germany, 1990; pp. 458–464.
8. Kurosawa, K.; Heng, S.H. 3-move undeniable signature scheme. In Proceedings of the Annual International Conference on the Theory and Applications of Cryptographic Techniques, Aarhus, Denmark, 22–26 May 2005; Springer: Berlin/Heidelberg, Germany, 2005; pp. 181–197.
9. Damgård, I.; Pedersen, T. New convertible undeniable signature schemes. In Proceedings of the International Conference on the Theory and Applications of Cryptographic Techniques, Saragossa, Spain, 12–16 May 1996; Springer: Berlin/Heidelberg, Germany, 1996; pp. 372–386.
10. Michels, M.; Stadler, M. Efficient convertible undeniable signature schemes. In Proceedings of the 4th Annual Workshop on Selected Areas in Cryptography (SAC'97), Ottawa, ON, Canada, 11–12 August 1997; pp. 231–244.
11. Duan, S. Certificateless undeniable signature scheme. *Inf. Sci.* **2008**, *178*, 742–755. [CrossRef]
12. Kurosawa, K.; Furukawa, J. Universally composable undeniable signature. In Proceedings of the International Colloquium on Automata, Languages, and Programming, Reykjavik, Iceland, 7–11 July 2008; Springer: Berlin/Heidelberg, Germany, 2008; pp. 524–535.
13. Ogata, W.; Kurosawa, K.; Heng, S.H. The security of the FDH variant of Chaum's undeniable signature scheme. In Proceedings of the International Workshop on Public Key Cryptography, Les Diablerets, Switzerland, 23–26 January 2005; Springer: Berlin/Heidelberg, Germany, 2005; pp. 328–345.
14. Rawal, S.; Padhye, S.; He, D. Lattice-based undeniable signature scheme. *Ann. Telecommun.* **2022**, *77*, 119–126. [CrossRef]
15. Yun, S. The Blockchain based Undeniable Multi-Signature Scheme for Protection of Multiple Authorship on Wisdom Contents. *J. Korea Internet Things Soc.* **2021**, *7*, 7–12.

16. Loh, J.C.; Heng, S.H.; Tan, S.Y.; Kurosawa, K. On the invisibility and anonymity of undeniable signature schemes. *J. Wirel. Mob. Netw. Ubiquitous Comput. Dependable Appl. (JoWUA)* **2020**, *11*, 18–34.
17. Yuen, T.H.; Heng, S.H. Security-mediated certificateless undeniable signature scheme. In *Third International Congress on Information and Communication Technology*; Springer: Singapore, 2019; pp. 25–32.
18. Aleksandrova, E.B.; Shkorkina, E. Using Undeniable Signature on Elliptic Curves to Verify Servers in Outsourced Computations. *Autom. Control Comput. Sci.* **2018**, *52*, 1160–1163. [CrossRef]
19. Desmedt, Y.G. Threshold cryptography. *Eur. Trans. Telecommun.* **1994**, *5*, 449–458. [CrossRef]
20. Camenisch, J.; Drijvers, M.; Lehmann, A.; Neven, G.; Towa, P. Short threshold dynamic group signatures. In Proceedings of the International Conference on Security and Cryptography for Networks, Amalfi, Italy, 14–16 September 2020; Springer: Cham, Switzerland, 2020; pp. 401–423.
21. Battagliola, M.; Longo, R.; Meneghetti, A.; Sala, M. Threshold ECDSA with an offline recovery party. *Mediterr. J. Math.* **2022**, *19*, 1–29. [CrossRef]
22. Komlo, C.; Goldberg, I. FROST: Flexible round-optimized Schnorr threshold signatures. In Proceedings of the International Conference on Selected Areas in Cryptography, Virtual Event, 19–23 October 2020; Springer: Cham, Switzerland, 2020; pp. 34–65.
23. Li, Y.; Wang, C.; Zhang, Y.; Yang, X.; Huang, H. Secure obfuscation for encrypted threshold signatures. *J. Commun.* **2020**, *41*, 61–69. [CrossRef]
24. Ruffing, T.; Ronge, V.; Jin, E.; Schneider-Bensch, J.; Schröder, D. ROAST: Robust Asynchronous Schnorr Threshold Signatures. Cryptology ePrint Archive, Report 2022/550, 2022. Available online: https://ia.cr/2022/550 (accessed on 10 May 2022).
25. Harn, L.; Yang, S. Group-oriented undeniable signature schemes without the assistance of a mutually trusted party. In Proceedings of the International Workshop on the Theory and Application of Cryptographic Techniques, Balatonfüred, Hungary, 24–28 May 1992; Springer: Berlin/Heidelberg, Germany, 1992; pp. 133–142.
26. Lin, C.H.; Wang, C.T.; Chang, C.C. A group-oriented (t, n) undeniable signature scheme without trusted center. In Proceedings of the Australasian Conference on Information Security and Privacy, Wollongong, NSW, Australia, 24–26 June 1996; Springer: Berlin/Heidelberg, Germany, 1996; pp. 266–274.
27. Lin, T.Y.; Wu, T.C. Undeniable (t, n)-threshold signature scheme with cheater identification. *J. Chin. Inst. Eng.* **1998**, *21*, 775–780. [CrossRef]
28. Lin, Y.B.; Tsengg, Y.F. Cryptanalysis on Lin and Wu's Undeniable (t, n)-Threshold Signature Scheme with Cheater Identification. In Proceedings of the 2021 International Symposium on Intelligent Signal Processing and Communication Systems (ISPACS), Hualien City, Taiwan, 16–19 November 2021; pp. 1–2.
29. Wang, G.; Qing, S.; Wang, M.; Zhou, Z. Threshold undeniable RSA signature scheme. In Proceedings of the International Conference on Information and Communications Security, Chongqing, China, 19–21 November 2021; Springer: Berlin/Heidelberg, Germany, 2001; pp. 221–232.
30. Wang, G.l.; Qing, S.H. A threshold undeniable signature scheme without a trusted party. *J. Softw.* **2002**, *13*, 1758–1764.
31. Lee, N.Y.; Hwang, T. Group-oriented undeniable signature schemes with a trusted center. *Comput. Commun.* **1999**, *22*, 730–734. [CrossRef]
32. Hwang, S.J.; Liao, H.C. A Group-Oriented Undeniable Signature Scheme for Unlikely Signers and Verifiers. *J. Appl. Sci. Eng.* **2006**, *9*, 45–54.
33. Liu, Y.; Liu, T. A novel threshold signature scheme based on elliptic curve with designated verifier. In Proceedings of the International Conference on Artificial Intelligence and Security, New York, NY, USA, 26–28 July 2019; Springer: Cham, Switzerland, 2019; pp. 332–342.
34. Gennaro, R.; Goldfeder, S. Fast Multiparty Threshold ECDSA with Fast Trustless Setup. In Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security (CCS '18), Toronto, ON, Canada, 15–19 October 2018; Association for Computing Machinery: New York, NY, USA, 2018; pp. 1179–1194. [CrossRef]
35. Dolev, D.; Dwork, C.; Naor, M. Nonmalleable cryptography. *SIAM Rev.* **2003**, *45*, 727–784. [CrossRef]
36. Di Crescenzo, G.; Ishai, Y.; Ostrovsky, R. Non-interactive and non-malleable commitment. In Proceedings of the Thirtieth Annual ACM Symposium on Theory of Computing, Dallas, TX, USA, 24–26 May 1998; pp. 141–150.
37. Crescenzo, G.D.; Katz, J.; Ostrovsky, R.; Smith, A. Efficient and non-interactive non-malleable commitment. In Proceedings of the International Conference on the Theory and Applications of Cryptographic Techniques, Innsbruck, Austria, 6–10 May 2001; Springer: Berlin/Heidelberg, Germany, 2001; pp. 40–59.
38. Damgard, I.; Groth, J. Non-interactive and reusable non-malleable commitment schemes. In Proceedings of the Thirty-Fifth Annual ACM Symposium on Theory of Computing, San Diego, CA, USA, 9–11 June 2003; pp. 426–437.
39. Gennaro, R. Multi-trapdoor commitments and their applications to proofs of knowledge secure under concurrent man-in-the-middle attacks. In Proceedings of the Annual International Cryptology Conference, Santa Barbara, CA, USA, 15–19 August 2004; Springer: Berlin/Heidelberg, Germany, 2004; pp. 220–236.
40. MacKenzie, P.; Yang, K. On simulation-sound trapdoor commitments. In Proceedings of the International Conference on the Theory and Applications of Cryptographic Techniques, Interlaken, Switzerland, 2–6 May 2004; Springer: Berlin/Heidelberg, Germany, 2004; pp. 382–400.
41. Ishai, Y.; Ostrovsky, R.; Zikas, V. Secure multi-party computation with identifiable abort. In Proceedings of the Annual Cryptology Conference, Santa Barbara, CA, USA, 17–21 August 2014; Springer: Berlin/Heidelberg, Germany, 2014; pp. 369–386.

42. Feldman, P. A practical scheme for non-interactive verifiable secret sharing. In Proceedings of the 28th Annual Symposium on Foundations of Computer Science (sfcs 1987), Los Angeles, CA, USA, 12–14 October 1987; pp. 427–438.
43. Galbraith, S.D.; Mao, W. Invisibility and anonymity of undeniable and confirmer signatures. In Proceedings of the Cryptographers' Track at the RSA Conference, San Francisco, CA, USA, 13–17 April 2003; Springer: Berlin/Heidelberg, Germany, 2003; pp. 80–97.
44. Pointcheval, D.; Stern, J. Security proofs for signature schemes. In Proceedings of the International Conference on the Theory and Applications of Cryptographic Techniques, Saragossa, Spain, 12–16 May 1996; Springer: Berlin/Heidelberg, Germany, 1996; pp. 387–398.