Bull. Korean Math. Soc. **46** (2009), No. 4, pp. 743–769
DOI 10.4134/BKMS.2009.46.4.743

NON-INTERACTIVE IDENTITY-BASED DNF SIGNATURE SCHEME AND ITS EXTENSIONS

Kwangsu Lee, Jung Yeon Hwang, and Dong Hoon Lee

Abstract. An ID-based DNF signature scheme is an ID-based signature scheme with an access structure which is expressed as a disjunctive normal form (DNF) with literals of signer identities. ID-based DNF signature schemes are useful to achieve not only signer-privacy but also a multi-user access control. In this paper, we formally define a notion of a (non-interactive) ID-based DNF signature and propose the first *non-interactive* ID-based DNF signature schemes that are secure under the computational Diffie-Hellman and subgroup decision assumptions. Our first scheme uses random oracles, and our second one is designed without random oracles. To construct the second one, we use a novel technique that converts a non-interactive witness indistinguishable proof system of encryption of one bit into a corresponding proof system of encryption of a bit-string. This technique may be of independent interest. The second scheme straightforwardly yields the first ID-based ring signature that achieves anonymity against full key exposure without random oracles. We finally present two extensions of the proposed ID-based DNF signature schemes to support multiple KGCs and different messages.

1. Introduction

The notion of a digital signature is one of the most fundamental and useful inventions of modern cryptography. Since the first public key cryptosystem in [10] was introduced, various signature schemes have been suggested to meet various needs in practical circumstances. In particular, combining an access structure with a signature scheme enables users to achieve important cryptographic goals such as user anonymity and multi-user access control, etc. Traditionally, in large-scale computer systems, the security for the important resources is achieved by access controls that describe which user or component

Received December 15, 2008; Revised July 7, 2009.
2000 *Mathematics Subject Classification.* 94A60.
Key words and phrases. identity-based signature, disjunctive normal form, signer anonymity, access structure.
This work was supported by the IT R&D program of MKE/IITA [2009-S-001-02, Development of Security Technology for Car-Healthcare].
The preliminary version of this paper appeared in the proceedings of the International Conference on Information Security and Cryptology (ICISC) 2008.

of a system is allowed to access what resources. One way to describe the access control is to use an access structure which is defined as a collection of subject sets that can access to the object. In signature systems, a signature may be viewed as a resource and a signer who generates the signature as a subject. That is, the access structure can be used to describe a collection of signer sets who participate in generating a signature. For examples, a public-key signature system implicitly includes an access structure that describes only one signer. A multi-signature system implicitly includes an access structure that describes multiple signers who participate in generating a signature. A ring signature system implicitly includes access structure such that any signer in members of a signer set generates a signature.

By applying ID-based cryptography to a signature scheme, we can construct an ID-based signature scheme in which user identity is used as a user public key [23, 15, 8, 2]. Particularly, an ID-based signature scheme is more suitable for dealing with a complex access structure to represent an authorized set of signers, because it does not require additional information like certificates to verify a signature. In this paper, as a cryptographic primitive for more generalized access structure, we study an *ID-based DNF signature scheme*, that is an ID-based signature scheme associated with an access structure expressed as a disjunctive normal form (DNF) with "OR" and "AND" operators and an identity ID as a literal. An ID-based DNF signature is valid only if the evaluation of the corresponding DNF is true. A literal ID is evaluated to be true when a signature generated by a signer with ID is valid and false when the signature is invalid or no signature is provided. While several ID-based DNF signature schemes have been proposed [14, 9], previously known schemes require interactive co-operation among signers in the access structure. That is, each signer broadcasts his random commitment and generates his own individual signature using others' random commitments. Individual signatures are then sent to a representor of signers who generates a final signature by combining the access structure. Since many parties participate in signing process, this interactive communication requires costly communication complexity with respect to system efficiency. Hence, it is highly desirable for an ID-based DNF signature scheme to be *non-interactive*.

Applications. An ID-based DNF signature scheme is a generalization of an ID-based multi-party signature scheme such as ID-based ring signature, multi-signature, designated-verifier signature, and threshold signature schemes. Thus an ID-based DNF signature scheme can be applied to various applications where an ID-based multi-party signature scheme is applied. Additionally, we can also apply it to other applications to which previous ID-based multi-party signature schemes are not suited because of inefficiency or inadequacy. For example, we may consider the situation that at least two valid signatures are necessary to guarantee the validity of a message without revealing the identities of the signers. A naive approach might be to use a ring signature scheme twice and

generate two ring signatures, one for each signer. In case of using an ID-based DNF signature, two identities can be simply paired by "AND" operator in the access structure. Hence, to verify the validity of a message, we need only one signature, which in turns reduces the verification time of the signature.

Our Results. In this paper, we first give a formal definition of a non-interactive ID-based DNF signature scheme. To capture the non-interactive property, we allow individual signature queries to the adversary. Our unforgeability model captures the attacker of insider corruption and anonymity model captures the attacker of full key exposure. To construct ID-based DNF signature schemes, we extend Groth, Ostrovsky, and Sahai's non-interactive witness indistinguishable (NIWI) proof system [13] of encryption of 0 or 1 bit to encryption of two bit-strings. This extended GOS NIWI proof may be of independent interest. We use this extended one to facilitate all-or-nothing encryption of signer identities in our ID-based DNF signature without random oracles. Next we propose two non-interactive ID-based DNF signature schemes. Our first construction is efficient and the size of a signature is compact. The security of the construction is proven under the computational Diffie-Hellman (CDH) and the subgroup decision (SD) assumptions in the random oracle model. Our second construction is proven secure under the same assumptions without random oracles, while it is relatively inefficient and the size of a signature is not compact, compared to the first one. We note that the second construction directly yields the first ID-based ring signature to achieve signer anonymity against full key exposure without random oracles, because an ID-based ring signature scheme is a special case of an ID-based DNF signature scheme. Finally, we extend our ID-based DNF signature scheme with random oracles to support multiple key generation centers or different messages. Our first extension for multiple KGCs enables signers from different KGCs to generate a signature. Our second extension allows that each signer can independently generates individual signature of his own message.

Related Works. Bresson et al. [7] proposed the first signature scheme with an access structure by extending Rivest et al.'s ring signature scheme [19]. They called their scheme as ad-hoc group signature. Recently, Boyen [6] proposed Mesh signature that allows each signer to generate a signature for different messages by extending the access structure. To overcome the certificate management problem in public key signatures, ID-based signature was proposed [23, 25, 8, 14, 9, 17]. The certificate management is a critical burden in signature schemes with an access structure, because the access structure contains many certificates to be verified. Thus an ID-based signature scheme with an access structure may be one of prominent solutions to resolve this problem. Herranz and Sáez [14] constructed the first ID-based signature with an access structure by extending their ID-based ring signature. Chow et al. [9] proposed another ID-based signature with an access structure by extending their ID-based ring signature that is based on Cha-Cheon ID-based signature scheme

[8]. However previous two schemes require interactive communication between signers and are secure in the random oracle model. As noted above, interaction between signers greatly deteriorates the efficiency of system.

Another line of research that uses access structures is attribute-based encryption (ABE) schemes [20, 12, 4, 18]. In attribute-based encryption schemes, the ciphertext is represented with multiple attributes and the user's private key is associated with an access structure that specifies what kinds of attributes are accepted as valid one. If attributes in the ciphertext satisfies the access structure in the user's private key, then the user can decrypt the ciphertext; otherwise, the user can't decrypt the ciphertext. The attribute-based encryption schemes are easily integrated with the role-based access control (RBAC) system [21], because roles in the RBAC are used for attributes in the ABE. The main difference between attribute-based systems and ID-based DNF systems is that in attribute-based systems, a user has a private key for multiple attributes, while in ID-based DNF systems, a user has a private key for a single attribute. So the non-interactiveness is not needed in attribute-based systems, but the collusion resistance that prevents the construction of new private key from different user's private keys is essential in attribute-based systems.

2. Backgrounds

We review the access structure, the disjunctive normal form, the bilinear groups and the complexity assumptions that our schemes are based on.

2.1. Access structure

Let $\{P_1, P_2, \ldots, P_n\}$ be a set of parties. A collection $\mathbb{A} \subseteq 2^{\{P_1,P_2,\ldots,P_n\}}$ is monotone if $\forall B, C$: if $B \in \mathbb{A}$ and $B \subseteq C$ then $C \in \mathbb{A}$. An access structure (respectively, monotone access structure) is a collection (respectively, monotone collection) $\mathbb{A}$ of non-empty subsets of $\{P_1, P_2, \ldots, P_n\}$, i.e., $A \subseteq 2^{\{P_1,P_2,\ldots,P_n\}} \setminus \{\emptyset\}$ [1]. The sets in $\mathbb{A}$ are called the authorized sets, and the sets not in $\mathbb{A}$ are called the unauthorized sets.

For ID-based systems, the parties are replaced as a set of identities, Thus the access structure $\mathbb{A}$ contains the authorized set of identities.

2.2. Disjunctive normal form

A logical formula ψ is in disjunctive normal form (DNF) if and only if it is a disjunction ($\vee$) of one or more conjunctions ($\wedge$) of one or more literals where literal is an atomic formula (atom) or its negation. We define a DNF formula ψ as a logical formula ψ in disjunctive normal form with restriction that literal is an identity. That is, $\psi = \vee_{i=1}^{a} \wedge_{j=1}^{b_i} \mathsf{ID}_{i,j}$, where $\mathsf{ID}_{i,j}$ is an identity. We say that a set S of identities satisfies a DNF formula ψ if and only if there exists a set $S' \subseteq S$ such that $\psi(S') = 1$.

Note that an access structure $\mathbb{A}$ can be represented as a DNF formula ψ. That is, the conjunction and the disjunction of the DNF formula ψ are used

to represent the subset of parties and the collection of subsets in the access structure $\mathbb{A}$ respectively.

2.3. Bilinear groups of composite order

Let $n = pq$, where p and q are prime numbers. Let $\mathbb{G}$ and $\mathbb{G}_T$ be two multicative cyclic groups of the same composite order n and g a generator of $\mathbb{G}$. The bilinear map $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ has the following properties:

(1) Bilinearity: $\forall u, v \in \mathbb{G}$ and $\forall a, b \in \mathbb{Z}_n$, we have $e(u^a, v^b) = e(u, v)^{ab}$ where the product in the exponent is a defined modulo n.
(2) Non-degeneracy: $e(g, g) \neq 1$ and is a generator of $\mathbb{G}_T$ with order n.

We say that $\mathbb{G}$ is a bilinear group if the group operations in $\mathbb{G}$ and $\mathbb{G}_T$ as well as the bilinear map e are all efficiently computable. Note that $e(\cdot, \cdot)$ is symmetric since $e(g^a, g^b) = e(g, g)^{ab} = e(g^b, g^a)$.

2.4. Complexity assumptions

We define two complexity assumptions: the Computational Diffie-Hellman and the Subgroup Decision assumptions.

Computational Diffie-Hellman (CDH) Assumption. Let $\mathbb{G}$ be a bilinear group of composite order $n = pq$. Let $\mathbb{G}_p$ be a subgroup of order p of $\mathbb{G}$ with a generator $g_p \in \mathbb{G}_p$. The CDH assumption in $\mathbb{G}_p$ with the composite order setting is that there is no probabilistic polynomial-time (PPT) algorithm $\mathcal{A}$ that, given a tuple (g_p, g_p^a, g_p^b) with the description of bilinear group $\mathbb{G}$ and its factorization (p, q) of order n, computes g_p^{ab} with non-negligible advantage. The advantage of $\mathcal{A}$ is defined as follows:

$$\mathsf{Adv}^{\mathsf{CDH}}_{\mathcal{A},\mathbb{G},\mathbb{G}_p} = \Pr\left[\mathcal{A}((n, p, q, \mathbb{G}, \mathbb{G}_T, e), g_p, g_p^a, g_p^b) = g_p^{ab}\right],$$

where the probability is taken over the random choice of the generator $g_p \in \mathbb{G}_p$ and $a, b \in \mathbb{Z}_p$, and the random bits consumed by $\mathcal{A}$.

Subgroup Decision (SD) Assumption. Let $\mathbb{G}$ be a bilinear group of composite order $n = pq$. Let $\mathbb{G}_q$ be a subgroup of order q of $\mathbb{G}$. The Subgroup Decision (SD) assumption is that there is no PPT algorithm $\mathcal{A}$ that, given the description of $\mathbb{G}$ and h selected at random either from $\mathbb{G}$ or from $\mathbb{G}_q$, decides whether $h \in \mathbb{G}_q$ or not with non-negligible advantage. The advantage of $\mathcal{A}$ is defined as follows:

$$\begin{aligned}\mathsf{Adv}^{\mathsf{SD}}_{\mathcal{A},\mathbb{G},\mathbb{G}_q} = \Big| &\Pr\left[h \in_R \mathbb{G} : \mathcal{A}((n, \mathbb{G}, \mathbb{G}_T, e), h) = 1\right] - \\ &\Pr\left[h \in_R \mathbb{G}_q : \mathcal{A}((n, \mathbb{G}, \mathbb{G}_T, e), h) = 1\right] \Big|,\end{aligned}$$

where the probability is taken over the random choice of h and the random bits consumed by $\mathcal{A}$.

3. Definitions

Informally, an ID-based DNF signature scheme is an identity-based signature scheme expressing that the signature was generated by a signer set that satisfies a DNF formula, but it does not leak any information about the signer set. An ID-based DNF signature scheme should satisfy two security properties, namely, unforgeability and anonymity. Unforgeability is satisfied if an adversary cannot construct a valid signature on a DNF formula when he does not know private keys that satisfy the DNF formula. Anonymity is satisfied if an adversary cannot distinguish which signer set generated the signature. For security model, we adopt the strong definitions of ring signatures, namely, unforgeability against *insider corruption* and anonymity against *full key exposure* in [3].

3.1. Definition of scheme

An ID-based DNF signature (IBDNFS) scheme consists of five algorithms (Setup, KeyGen, Sign, Merge, Verify). Formally it is defined as:

- $\mathsf{Setup}(1^\lambda)$. The setup algorithm takes as input a security parameter, outputs a public parameters PP and a master secret key MK.
- $\mathsf{KeyGen}(\mathsf{ID}, \mathsf{MK}, \mathsf{PP})$. The key generation algorithm takes as input an identity ID, the master secret key MK and the public parameters PP, outputs a private key $\mathsf{SK}_{\mathsf{ID}}$.
- $\mathsf{Sign}(M, \psi, \mathsf{SK}_{\mathsf{ID}}, \mathsf{PP})$. The individual signing algorithm takes as input a message M, a DNF formula ψ, a private key $\mathsf{SK}_{\mathsf{ID}}$ and the public parameters PP, then outputs an individual signature θ for M and ψ.
- $\mathsf{Merge}(M, \psi, \mathsf{SS}, \mathsf{PP})$. The merge algorithm takes as input a message M, a DNF formula $\psi = \vee_{i=1}^{a} \wedge_{j=1}^{b_i} \mathsf{ID}_{i,j}$, an individual signature set $\mathsf{SS} = \{(\mathsf{ID}_{i^*,j}, \theta_{i^*,j}) \mid i^* \in \{1, \ldots, a\}\}_{1 \le j \le b_{i^*}}$ and the public parameters PP, then outputs an ID-based DNF signature σ.
- $\mathsf{Verify}(\sigma, M, \psi, \mathsf{PP})$. The verification algorithm takes as input a signature σ, a message M, a DNF formula ψ and the public parameters PP, then outputs "accept" or "reject", depends on the validity of the signature.

For non-interactive ID-based DNF signature schemes, we separated the signature generation algorithm as sign and merge algorithms. Thus each user individually generates its own signature (without interactions), then someone merges the whole individual signatures as an ID-based DNF signature. For interactive ID-based DNF signature schemes, it is possible to combine sign and merge algorithms.

If a DNF formula is represented as $\psi = \vee_{i=1}^{1} \wedge_{j=1}^{b_i} \mathsf{ID}_{i,j}$, then the ID-based DNF signature of ψ equals with the ID-based multi-signature. If $\psi = \vee_{i=1}^{a} \wedge_{j=1}^{1} \mathsf{ID}_{i,j}$, then the ID-based DNF signature of ψ equals with the ID-based ring

signature. In case of the ID-based threshold signature, the t-out-of-n threshold can be restated as a DNF formula ψ.

3.2. Definition of security

Unforgeability against insider corruption is defined via the following game between a challenger $\mathcal{C}$ and an adversary $\mathcal{A}$:

Setup: $\mathcal{C}$ runs the setup algorithm and keeps the master secret key MK to itself, then it gives the public parameters PP to $\mathcal{A}$.

Queries: Adaptively, $\mathcal{A}$ can request any queries described below.

- Private key query: $\mathcal{A}$ requests a private key on an identity ID.
- Individual signature query: $\mathcal{A}$ requests an individual signature for a message M, a DNF formula ψ and an identity ID.
- Signature query: $\mathcal{A}$ requests a signature for a message M and a DNF formula ψ.

$\mathcal{C}$ accepts or responds to each request before accepting the next one. $\mathcal{A}$ makes q_E private key queries, q_S signature queries (including individual signature queries).

Output: Finally, $\mathcal{A}$ outputs a pair (σ^*, M^*, ψ^*) and wins the game if (1) the corrupted identities set $C = \{\mathsf{ID}_i\}_{1\le i\le q_E}$ by private key queries does not satisfy the DNF formula ψ^*; (2) let S be the set of identities that was requested an individual signatures queries for (M^*, ψ^*), then $S \cup C$ does not satisfy the DNF formula ψ^*; (3) $\mathcal{A}$ did not request a signature for a pair (M^*, ψ^*); (4) $\mathsf{Verify}(\sigma^*, M^*, \psi^*, \mathsf{PP}) =$ "accept".

Let Succ be the event that $\mathcal{A}$ wins the above game. The advantage of $\mathcal{A}$ is defined as $\mathsf{Adv}_{\mathcal{A}}^{\mathsf{IBDNFS\text{-}UF}} = \Pr[\mathsf{Succ}]$ where the probability is taken over the coin tosses made by $\mathcal{A}$ and $\mathcal{C}$.

Definition 1. An adversary $\mathcal{A}$ is said to (t, ϵ, q_E, q_S)-break an ID-based DNF signature scheme if $\mathcal{A}$ runs in time at most t, $\mathcal{A}$ makes at most q_E private key queries and at most q_S signing oracle queries, and $\mathsf{Adv}_{\mathcal{A}}^{\mathsf{IBDNFS\text{-}UF}}$ is at least ϵ. An ID-based DNF signature scheme is (t, ϵ, q_E, q_S)-unforgeable if there exists no adversary that (t, ϵ, q_E, q_S)-breaks it.

Anonymity against full key exposure is defined via the following game between a challenger $\mathcal{C}$ and an adversary $\mathcal{A}$.

Setup: $\mathcal{C}$ runs the setup algorithm and keeps the master secret key MK to itself, then it gives the public parameters PP to $\mathcal{A}$.

Queries: Adaptively, $\mathcal{A}$ can request any queries described below.

- Private key query: $\mathcal{A}$ requests a private key on an identity ID.
- Individual signature query: $\mathcal{A}$ requests an individual signature for a message M, a DNF formula ψ and an identity ID.

- Signature query: $\mathcal{A}$ requests a signature for a message M and a DNF formula ψ.

$\mathcal{C}$ accepts or responds to each request before accepting the next one. $\mathcal{A}$ makes q_E private key queries, q_S signature queries (including individual signature queries).

Challenge: $\mathcal{A}$ submits a challenge tuple (M, ψ, i_0, i_1) where $\psi = \vee_{i=1}^{a} \wedge_{j=1}^{b_i} \mathsf{ID}_{i,j}$ and $1 \leq i_0 \neq i_1 \leq a$. $\mathcal{C}$ chooses a random coin $c \in \{0,1\}$ and computes $\sigma = \mathsf{Merge}(M, \psi, \mathsf{SS}_c, \mathsf{PP})$ where $\mathsf{SS}_c = \{(\mathsf{ID}_{i_c,j}, \theta_{i_c,j})\}_{1 \leq j \leq b_{i_c}}$ such that $\theta_{i_c,j}$ is an individual signature for (M, ψ) by the private key of $\mathsf{ID}_{i_c,j}$. Then $\mathcal{C}$ gives σ to $\mathcal{A}$.

Output: Finally, $\mathcal{A}$ outputs a guess c' of c and wins the game if $c' = c$.

Let Succ be the event that $\mathcal{A}$ wins the above game. The advantage of $\mathcal{A}$ is defined as $\mathsf{Adv}_{\mathcal{A}}^{\mathsf{IBDNFS\text{-}AN}} = \left| \Pr[\mathsf{Succ}] - \frac{1}{2} \right|$, where the probability is taken over the coin tosses made by $\mathcal{A}$ and $\mathcal{C}$.

Definition 2. An adversary $\mathcal{A}$ is said to (t, ϵ, q_E, q_S)-break an ID-based DNF signature scheme if $\mathcal{A}$ runs in time at most t, $\mathcal{A}$ makes at most q_E private key queries and at most q_S signing oracle queries, and $\mathsf{Adv}_{\mathcal{A}}^{\mathsf{IBDNFS\text{-}AN}}$ is at least ϵ. An ID-based DNF signature scheme is (t, ϵ, q_E, q_S)-anonymous if there exists no adversary that (t, ϵ, q_E, q_S)-breaks it.

4. Extended GOS proof

Boneh, Goh, and Nissim [5] proposed an encryption scheme that has homomorphic property that allows computations on ciphertexts involving arbitrary additions and one multiplication. Groth, Ostrovsky, and Sahai [13] constructed efficient non-interactive witness-indistinguishable proof system based on BGN encryption system. In this section, we construct an extended GOS proof system for encryption of two l-bit strings $(0, \ldots, 0)$ and $(1, \ldots, 1)$. Later, we use it for our construction of a DNF signature without random oracles. The extended GOS proof is described as follows.

$\mathsf{Setup}(1^\lambda)$: The setup algorithm takes as input a security parameter λ, then it generates a bilinear group $\mathbb{G}$ of composite order $n = pq$, where p and q are random primes of bit size $\Theta(\lambda)$, and it selects random generators $g \in \mathbb{G}$ and $h \in \mathbb{G}_q$. Then the common reference string is set by $\mathsf{CRS} = (n, \mathbb{G}, \mathbb{G}_T, e, g, h)$.

$\mathsf{Statement}$: Let $A = (0, \ldots, 0)_l \in \mathbb{Z}_p^l$ and $B = (1, \ldots, 1)_l \in \mathbb{Z}_p^l$, where $l < p$. The statement is a ciphertext $C = (C_1, \ldots, C_l)$, and the claim is that there exists a witness $W = (M = (m_1, \ldots, m_l), Z = (z_1, \ldots, z_l)) \in \{A, B\} \times \mathbb{Z}_n^l$ such that $m_i \in \{0,1\}$ and $C_i = g^{m_i} h^{z_i}$.

$\mathsf{Prove}(C, W, \mathsf{CRS})$: To generate a proof on the ciphertext $C = (C_1, \ldots, C_l)$ with the witness $W = (M = (m_1, \ldots, m_l), Z = (z_1, \ldots, z_l))$, it first checks

$M \stackrel{?}{\in} \{A, B\}$ and $C_i \stackrel{?}{=} g^{m_i} h^{z_i}$ for all $i \in \{1, \ldots, l\}$. Next it defines f as $f = 0$ if $M = A$ and $f = 1$ if $M = B$. Then it outputs a proof of the claim as $P = \big(\pi_1 = g^{m_1} h^{z_1}, \ldots, \pi_l = g^{m_l} h^{z_l},\ \pi = (g^{l(2f-1)} \cdot h^{\Sigma_{i=1}^l z_i})^{\Sigma_{i=1}^l z_i}\big)$.

$\mathsf{Verify}(C, P, \mathsf{CRS})$: To verify the proof $P = (\pi_1, \ldots, \pi_l, \pi)$ for the ciphertext $C = (C_1, \ldots, C_l)$, it checks $e(C_i, C_i/g) \stackrel{?}{=} e(h, \pi_i)$ for all $i \in \{1, \ldots, l\}$, and checks $e(\prod_{j=1}^l C_j, \prod_{j=1}^l (C_j/g)) \stackrel{?}{=} e(h, \pi)$. If all tests are successful, then it outputs "accept"; otherwise it outputs "reject".

Remark 4.1. For $A = 0$ and $B = 1$ with bit-length 1, our extended GOS proof is exactly the original GOS proof.

Theorem 4.2. *The above extended GOS proof satisfies perfect completeness, perfect soundness, and computational witness indistinguishability under the subgroup decision assumption.*

Proof. Perfect Completeness. We know that $C_i = g^{m_i} h^{z_i}$ where $m_i \in \{0, 1\}$. This gives us that $e(C_i, C_i/g) = e(g^{m_i} h^{z_i}, g^{m_i - 1} h^{z_i}) = e(g, g)^{m_i(m_i-1)} \cdot e(h^{z_i}, g^{2m_i - 1} h^{z_i}) = e(h, \pi_i)$ for all $i \in \{1, \ldots, l\}$. Let $z^{sum} = \sum_{i=1}^l z_i$. We have $\prod_{i=1}^l C_i = g^{\Sigma_{i=1}^l m_i} h^{z^{sum}}$, where $\sum_{i=1}^l m_i \in \{0, l\}$, and f is defined as $f = (\sum_{i=1}^l m_i)/l$. Thus we have $e(\prod_{i=1}^l C_i, \prod_{i=1}^l (C_i/g)) = e(g, g)^{\Sigma_{i=1}^l m_i \cdot \Sigma_{i=1}^l (m_i - 1)} \cdot e(h^{z^{sum}}, g^{\Sigma_{i=1}^l (2m_i - 1)} h^{z^{sum}}) = e(h, (g^{l(2f-1)} h^{z^{sum}})^{z^{sum}}) = e(h, \pi)$.

Perfect Soundness. Since the proof π_i satisfies the verification equation, we have $e(C_i, C_i/g)^q = e(h, \pi_i)^q = e(h^q, \pi_i) = 1$ for all $i \in \{1, \ldots, l\}$. This means that C_i or C_i/g has order 1 or q. Since C_i can be written as $g^{m_i} h^{z_i}$ for some $m_i \in \mathbb{Z}_p, z_i \in \mathbb{Z}_n$, we see that $g^{m_i} h^{z_i}$ or $g^{m_i - 1} h^{z_i}$ has order 1 or q. Since h has order q, this means that $m_i = 0 \mod p$ or $m_i - 1 = 0 \mod p$.

Since the proof π is a valid one, we have $e(\prod_{i=1}^l C_i, \prod_{i=1}^l (C_i/g))^q = e(h, \pi)^q = e(h^q, \pi) = 1$. This means that $\prod_{i=1}^l C_i$ or $\prod_{i=1}^l (C_i/g)$ has order 1 or q. Since C_i is well formed as $g^{m_i} h^{z_i}$, where $m_i \in \{0, 1\}$, it implies that $g^{\Sigma_{i=1}^l m_i} h^{z^{sum}}$ or $g^{\Sigma_{i=1}^l (m_i - 1)} h^{z^{sum}}$ has order 1 or q. Since h has order q, this means that $\sum_{i=1}^l m_i = 0 \mod p$ or $\sum_{i=1}^l (m_i - 1) = 0 \mod p$. If at least one m_i is 1, where $m_i \in \{0, 1\}$, then $\sum_{i=1}^l m_i \neq 0 \mod p$. Thus the equation $\sum_{i=1}^l m_i = 0 \mod p$ gives us that all m_i are 0. Next if at least one $m_i = 0$, then $\sum_{i=1}^l m_i \neq l \mod p$. Thus the equation $\sum_{i=1}^l m_i = l \mod p$ gives us that all m_i are 1. Therefore we have that $(m_1, \ldots, m_l)$ is $(0, \ldots, 0)_l$ or $(1, \ldots, 1)_l$ in $\mathbb{Z}_p^l$.

Computational Witness Indistinguishability. If h is a generator of $\mathbb{G}$ instead of $\mathbb{G}_q$, there exist $z_{i0}, z_{i1} \in \mathbb{Z}_n$ such that $C_i = h^{z_{i0}} = g h^{z_{i1}}$ for all $i \in \{1, \ldots, l\}$. This implies that $\prod_{i=1}^l C_i = h^{\Sigma_{i=1}^l z_{i0}} = g^l h^{\Sigma_{i=1}^l z_{i1}}$. Let $(\pi|_{f=c})$ the value which π is assigned if f is set to $c \in \{0, 1\}$. We have $(\pi_i|_{f=0}) = \big(g^{-1} h^{z_{i0}}\big)^{z_{i0}} = \big(h^{z_{i1}}\big)^{z_{i0}} = \big(h^{z_{i0}}\big)^{z_{i1}} = \big(g h^{z_{i1}}\big)^{z_{i1}} = (\pi_i|_{f=1})$ and $(\pi|_{f=0}) = \big(g^{-l} h^{\Sigma_{i=1}^l z_{i0}}\big)^{\Sigma_{i=1}^l z_{i0}} = \big(h^{\Sigma_{i=1}^l z_{i1}}\big)^{\Sigma_{i=1}^l z_{i0}} = \big(h^{\Sigma_{i=1}^l z_{i0}}\big)^{\Sigma_{i=1}^l z_{i1}} = \big(g^l h^{\Sigma_{i=1}^l z_{i1}}\big)^{\Sigma_{i=1}^l z_{i1}}$

$= (\pi|_{f=1})$. Thus the proof gives no information about the witness. If h is a generator of $\mathbb{G}_q$, then it equals with our extended GOS proof. Therefore the difference probability between two cases of h gives the advantage of the subgroup decision assumption. □

5. Construction with random oracles

In this section, we construct a non-interactive ID-based DNF signature scheme and prove the security of our scheme in the random oracle model. Design intuition for our construction is consistently combining Shacham-Waters ring signature scheme [22] with Gentry-Ramzan multi-signature scheme [11]. To combine these two schemes, we work in a bilinear group of composite order. Our construction is described as follows.

5.1. Description

$\mathsf{Setup}(1^\lambda)$: The setup algorithm first generates a bilinear group $\mathbb{G}$ of composite order $n = pq$, where p and q are random primes of bit size $\Theta(\lambda)$. Next, it chooses random $g, w \in \mathbb{G}, h \in \mathbb{G}_q$, and $s \in \mathbb{Z}_n$. Finally it chooses cryptographic hash functions $H_1, H_2 : \{0,1\}^* \to \mathbb{G}$. Then the public parameters PP and the master secret key MK are set by

$$\mathsf{PP} = \big(n, \mathbb{G}, \mathbb{G}_T, e, g, g_1 = g^s, h, h_1 = h^s, w, H_1, H_2\big),\ \mathsf{MK} = s.$$

$\mathsf{KeyGen}(\mathsf{ID}, \mathsf{MK}, \mathsf{PP})$: The key generation algorithm takes as input an identity ID, the master secret key MK, and the public parameters PP, then outputs a private key $\mathsf{SK}_{\mathsf{ID}} = H_1(\mathsf{ID})^s$.

$\mathsf{Sign}(M, \psi, \mathsf{SK}_{\mathsf{ID}}, \mathsf{PP})$: The sign algorithm takes as input a message M, a DNF formula ψ, a private key $\mathsf{SK}_{\mathsf{ID}} = H_1(\mathsf{ID})^s$. Next, it computes $H_m = H_2(M, \psi)$, chooses a random $r \in \mathbb{Z}_n$, and then outputs an individual signature $\theta = (V, R) = \big(H_1(\mathsf{ID})^s \cdot H_m^r,\ g^r\big) \in \mathbb{G}^2$.

$\mathsf{Merge}(M, \psi, \mathsf{SS}, \mathsf{PP})$: The merge algorithm takes as input a message M, a DNF formula $\psi = \vee_{i=1}^{a} \wedge_{j=1}^{b_i} \mathsf{ID}_{i,j}$, a set $\mathsf{SS} = \{(\mathsf{ID}_{i^*,j}, \theta_{i^*,j})\}_{1 \le j \le b_{i^*}}$, where i^* is an index such that $1 \le i^* \le a$ and $\theta_{i^*,j}$ is an individual signature $(V_{i^*,j}, R_{i^*,j})$ that was generated by $\mathsf{ID}_{i^*,j}$. Let $\{f_i\}_{1 \le i \le a}$ be such that $f_i = 1$ if $i = i^*$ and $f_i = 0$ if $i \neq i^*$. To generate a signature, it proceeds as follows:

(1) First, it constructs a (aggregate) multi-signature of the message M as $\tilde{V} = \prod_{j=1}^{b_{i^*}} V_{i^*,j}$ and $\tilde{R} = \prod_{j=1}^{b_{i^*}} R_{i^*,j}$ using the set SS.
(2) For each $i \in \{1, \ldots, a\}$, it chooses a random $z_i \in \mathbb{Z}_n$ and computes $(C_i = (Y_i/w)^{f_i} h^{z_i}, \pi_i = ((Y_i/w)^{2f_i - 1} h^{z_i})^{z_i})$ where $Y_i = \prod_{j=1}^{b_i} H_1(\mathsf{ID}_{i,j})$.
(3) To convert $(\tilde{V}, \tilde{R})$ as a blinded one that is verifiable and anonymous, it sets $z = \sum_{i=1}^{a} z_i$ and constructs $\sigma_1 = \tilde{V} \cdot h_1^z$ and $\sigma_2 = \tilde{R}$.
(4) It outputs a DNF signature $\sigma = \big(\sigma_1,\ \sigma_2,\ \{(C_i, \pi_i)\}_{1 \le i \le a}\big) \in \mathbb{G}^{2a+2}$.

$\mathsf{Verify}(\sigma, M, \psi, \mathsf{PP})$: The verify algorithm takes as input a signature σ, a message M and a DNF formula $\psi = \vee_{i=1}^{a} \wedge_{j=1}^{b_i} \mathsf{ID}_{i,j}$, then proceeds as follows:

(1) For all $i \in \{1, \ldots, a\}$, it checks if $e(C_i, C_i/(Y_i/w)) \stackrel{?}{=} e(h, \pi_i)$, where $Y_i = \prod_{j=1}^{b_i} H_1(\mathsf{ID}_{i,j})$.
(2) Next, it checks if $e(g, \sigma_1) \stackrel{?}{=} e(g_1, w \prod_{i=1}^{a} C_i) \cdot e(\sigma_2, H_m)$, where $H_m = H_2(M, \psi)$.
(3) If all tests are successful, then it outputs "accept"; otherwise it outputs "reject".

It is easy to show that the above scheme satisfies the correctness as follows.

$$\begin{aligned} e(g, \sigma_1) &= e(g, \prod_{j=1}^{b_{i^*}} (H_1(\mathsf{ID}_{i^*,j})^s H_m^{r_j}) \cdot h_1^z) = e(g^s, Y_{i^*} \cdot h^z) \cdot e(\prod_{j=1}^{b_{i^*}} g^{r_j}, H_m) \\ &= e(g_1, w \prod_{i=1}^{a} C_i) \cdot e(\sigma_2, H_m), \end{aligned}$$

where $Y_{i^*} = \prod_{j=1}^{b_{i^*}} H_1(\mathsf{ID}_{i^*,j})$ and $z = \sum_{i=1}^{a} z_i$.

5.2. Security

Theorem 5.1. *The above ID-based DNF signature scheme satisfies unforgeability under the CDH assumption on $\mathbb{G}_p$ in the random oracle model.*

Proof. In this proof, we construct an algorithm $\mathcal{B}$ that solves the CDH problem in $\mathbb{G}_p$ running an adversary $\mathcal{A}$ attacking the presented scheme. Note that each proof (C_i, π_i) in a forged signature generated by $\mathcal{A}$ must pass the verification equation, $e(C_i, C_i/(Y_i/w)) = e(h, \pi_i)$. As described in [13], this implies that C_i has the form $(Y_i/w)^{f_i} h^{z_i}$ for some $f_i \in \{0, 1\}$ and $z_i \in \mathbb{Z}_n$. According to the value of $\sum_{i=1}^{a} f_i$, we consider two types of adversaries as follows.

(1) Type-1 adversary $\mathcal{A}_1$ is one of which forgery is not such that exactly one of the exponents $\{f_i\}$ equals 1, that is, $\sum_{i=1}^{a} f_i \neq 1$.
(2) Type-2 adversary $\mathcal{A}_2$ is one of which forgery is such that exactly one of the exponents $\{f_i\}$ equals 1, that is, $\sum_{i=1}^{a} f_i = 1$.

For each type of adversary $\mathcal{A}_1$ and $\mathcal{A}_2$, we will construct algorithms $\mathcal{B}_1$ and $\mathcal{B}_2$ to solve the CDH problem in $\mathbb{G}_p$, respectively. The proof easily follows from the following two lemmas. □

Before giving the detailed proofs of the following lemmas, we give the intuitive description of them. In case of the type-2 adversary $\mathcal{A}_2$, the algorithm $\mathcal{B}_2$ can extract an ID-based signature from the output of $\mathcal{A}_2$, because $\mathcal{B}_2$ knows the factorization p, q of n and the output of $\mathcal{A}_2$ satisfies $\sum_{i=1}^{a} f_i = 1$.

In case of the type-1 adversary $\mathcal{A}_1$, the algorithm $\mathcal{B}_1$ can not use the method of $\mathcal{B}_2$ because the output of $\mathcal{A}_1$ satisfies $\sum_{i=1}^{a} f_i \neq 1$. Instead it uses Shacham-Waters ring signature technique [22]. That is, it embeds g_p^{α} and g_p^{β} of the

CDH assumption to the g_1 and w of the public parameters, respectively. If the output of $\mathcal{A}_1$ satisfies $\sum_{i=1}^{a} f_i \neq 1$, then it can solve the CDH assumption using the output of $\mathcal{A}_1$.

Lemma 5.2. *If there exists a type-1 adversary $\mathcal{A}_1$, then there exists an algorithm $\mathcal{B}_1$ that solves the CDH problem.*

Proof. Suppose there exists a type-1 adversary $\mathcal{A}_1$ that breaks unforgeability of the above scheme. The algorithm $\mathcal{B}_1$ that solves the CDH problem using $\mathcal{A}_1$ is given: The description of the bilinear group $\mathbb{G}$, the factorization p, q of order n, and a random CDH challenge $(g_p, g_p^{\alpha}, g_p^{\beta}) \in \mathbb{G}_p^3$, where g_p is a generator of $\mathbb{G}_p$. Its goal is to compute $g_p^{\alpha\beta}$. The algorithm $\mathcal{B}_1$ interacts with $\mathcal{A}_1$ as follows:

Setup: $\mathcal{B}_1$ selects a generator $h \in \mathbb{G}_q$ and chooses random values $r_1 \in \mathbb{Z}_q^*$, $r_2, r_3 \in \mathbb{Z}_q$. Next it sets the public parameters $\mathsf{PP} = (n, \mathbb{G}, \mathbb{G}_T, e, g = g_p h^{r_1}, g_1 = g_p^{\alpha} h^{r_2}, h, h_1 = h^{r_2/r_1}, w = g_p^{\beta} h^{r_3}, H_1, H_2)$ and gives PP to $\mathcal{A}_1$. The PP are correctly distributed because $e(g_1, h) = e(g_p^{\alpha} h^{r_2}, h) = e(h^{r_1}, h^{r_2/r_1}) = e(g_p h^{r_1}, h^{r_2/r_1}) = e(g, h_1)$.

Queries: Adaptively $\mathcal{A}_1$ can make an H_1-hash query, H_2-hash query, private key query, or signature query at any time. For hash queries, $\mathcal{B}_1$ maintains H_1-list and H_2-list relating to its previous hash query responses for consistency.

For a H_1-hash query on ID_i, $\mathcal{B}_1$ generates a random $c_i \in \mathbb{Z}_n$ and responds with $H_1(\mathsf{ID}_i) = g^{c_i}$. For a H_2-hash query on (M_i, ψ_i), $\mathcal{B}_1$ generates a random $d_i \in \mathbb{Z}_n$ and responds with $H_2(M_i, \psi_i) = g^{d_i}$. For a private key query on ID_i, $\mathcal{B}_1$ can generate the private key $g_1^{c_i}$ because it knows the discrete logarithm $c_i = \log_g H_1(\mathsf{ID}_i)$ of H_1-hash values. Using the private key, it is easy to respond for individual signature queries and signature queries.

Output: Finally, $\mathcal{A}_1$ outputs a forged DNF signature (σ^*, M^*, ψ^*), where $\sigma^* = (\sigma_1, \sigma_2, \{(C_i, \pi_i)\}_{1 \leq i \leq a})$ and $\psi^* = \vee_{i=1}^{a} \wedge_{j=1}^{b_i} \mathsf{ID}_{i,j}$.

If (1) the corrupted identities set $C = \{\mathsf{ID}_i\}_{1 \leq i \leq q_E}$ by private key queries satisfies the DNF formula ψ^*; or (2) let S be the set of identity that was requested an individual signatures queries for (M^*, ψ^*), then $S \cup C$ satisfies the DNF formula ψ^*; or (3) $\mathcal{A}$ did request a signature for a pair (M^*, ψ^*); or (4) $\mathsf{Verify}(\sigma^*, M^*, \psi^*, \mathsf{PP}) \neq$ "accept", then $\mathcal{B}_1$ stops the simulation because $\mathcal{A}_1$ was not successful.

$\mathcal{B}_1$ solves the given CDH problem as follows: Let δ_p be such that $\delta_p = 0 \mod q$ and $\delta_p = 1 \mod p$. By the property of bilinear groups of composite order, we have $u^{\delta_p} \in \mathbb{G}_p$ for all $u \in \mathbb{G}$, because $u^{\delta_p} = 1$ if and only if $u \in \mathbb{G}_q$. We obtain $C_i^{\delta_p} = (Y_i^{\delta_p} / w^{\delta_p})^{f_i} = ((\prod_{j=1}^{b_i} g^{c_{i,j}})^{\delta_p} / w^{\delta_p})^{f_i} = (g_p^{\sum_{j=1}^{b_i} c_{i,j}} / g_p^{\beta})^{f_i}$ for all $i \in \{1, \ldots, a\}$, and so $C^{\delta_p} = \prod_{i=1}^{a} {C_i}^{\delta_p} = g_p^{c} / (g_p^{\beta})^{f}$, where $c = \sum_{i=1}^{a} (\sum_{j=1}^{b_i} c_{i,j}) f_i$ and $f = \sum_{i=1}^{a} f_i$. From the verification equation, we obtain $e(g_p, {\sigma_1}^{\delta_p}) = e(g_p^{\alpha}, g_p^{\beta} \cdot g_p^{c} / (g_p^{\beta})^{f}) \cdot e({\sigma_2}^{\delta_p}, g_p^{d})$, where $H_2(M^*, \psi^*)^{\delta_p} =$

g_p^d. By restating this equation, we have $e(g_p^\alpha, g_p^\beta)^{1-f} = e(g_p, {\sigma_1}^{\delta_p} \cdot ({\sigma_2}^{\delta_p})^{-d} \cdot (g_p^\alpha)^{-c})$. $\mathcal{B}_1$ can recover $(\mathsf{ID}_{i,j}, c_{i,j})$ and (M^*, ψ^*, d) from the H_1-list and the H_2-list. In addition, $\mathcal{B}_1$ recovers $\{f_i\}_{1 \le i \le a}$ values using the property $C_i^{\delta_p} = 1$ if and only if $f_i = 0$. By assumption that $f = \sum_{i=1}^{a} f_i \neq 1$, we know that $(1-f)^{-1} \pmod p$ exists. Therefore, it solves the CDH problem as follows:

$$g_p^{\alpha\beta} = \left({\sigma_1}^{\delta_p} \cdot ({\sigma_2}^{\delta_p})^{-d} \cdot (g_p^\alpha)^{-c}\right)^{1/(1-f)}.$$

Since $\mathcal{B}_1$ succeeds whenever $\mathcal{A}_1$ does, we obtain the inequality $\mathsf{Adv}_{\mathcal{B}_1}^{\mathsf{CDH}} \ge \mathsf{Adv}_{\mathcal{A}_1}^{\mathsf{IBDNF\text{-}UF}}$. □

Lemma 5.3. *If there exists a type-2 adversary $\mathcal{A}_2$, then there exists an algorithm $\mathcal{B}_2$ that solves the CDH problem.*

Proof. Suppose there exists a type-2 adversary $\mathcal{A}_2$ that breaks unforgeability of the above scheme. The algorithm $\mathcal{B}_2$ that solves the CDH problem using $\mathcal{A}_2$ is given: The description of the bilinear group $\mathbb{G}$, the factorization p, q of order n, and the tuple $(g_p, g_p^\alpha, g_p^\beta) \in \mathbb{G}_p^3$, where g_p is a generator of $\mathbb{G}_p$. Its goal is to compute $g_p^{\alpha\beta}$. $\mathcal{B}_2$ interacts with $\mathcal{A}_2$ as follows:

Setup: $\mathcal{B}_2$ selects a generator $h \in \mathbb{G}_q$, chooses random values $r_1 \in \mathbb{Z}_q^*$, $r_2, r_3, r_4 \in \mathbb{Z}_q$, $r_5 \in \mathbb{Z}_p$. Next it sets $\mathsf{PP} = (n, \mathbb{G}, \mathbb{G}_T, e, g = g_p h^{r_1}, g_1 = g_p^\alpha h^{r_2}, h, h_1 = h^{r_2/r_1}, w = g_p^{r_5} h^{r_3}, H_1, H_2)$ and $g_2 = g_p^\beta h^{r_4}$. Then it gives PP to $\mathcal{A}$. The PP are correctly distributed by $e(g_1, h) = e(g_p^\alpha h^{r_2}, h) = e(h^{r_1}, h^{r_2/r_1}) = e(g_p h^{r_1}, h^{r_2/r_1}) = e(g, h_1)$.

Queries: Adaptively $\mathcal{A}_2$ can make an H_1-hash query, H_2-hash query, private key query, or signature query at any time. For hash queries, $\mathcal{B}_2$ gives identical responses to identical queries by maintaining lists relating to its previous hash query responses for consistency.

For a H_1-hash query on ID_i, $\mathcal{B}_2$ responds as follows: If ID_i was in a previous H_1-hash query, it recovers $(\mathsf{ID}_i, H_1\text{-}coin_i, c_i)$ from its H_1-list; else, it generates a random $H_1\text{-}coin_i \in \{0,1\}$ so that $\Pr[H_1\text{-}coin_i = 1] = \rho_1$ for ρ_1 to be determined later. It generates random $c_i \in \mathbb{Z}_n^*$ and logs $(\mathsf{ID}_i, H_1\text{-}coin_i, c_i)$ in its H_1-list. If $H_1\text{-}coin_i = 0$ then it responds with $H_1(\mathsf{ID}_i) = g^{c_i}$; else, it responds with $H_1(\mathsf{ID}_i) = g_2^{c_i}$.

For a H_2-hash query on (M_i, ψ_i), $\mathcal{B}_2$ responds as follows: If (M_i, ψ_i) was in a previous H_2-hash query, it recovers $(M_i, \psi_i, H_2\text{-}coin_i, d_i, d_i')$ from its H_2-list; else, it generates a random $H_2\text{-}coin_i \in \{0,1\}$ so that $\Pr[H_2\text{-}coin_i = 1] = \rho_2$ for ρ_2 to be determined later. If $H_2\text{-}coin_i = 0$, it generates a random $d_i \in \mathbb{Z}_n$ and sets $d_i' = 0$; else it generates random $d_i, d_i' \in \mathbb{Z}_n^*$. It logs $(M_i, \psi_i, H_2\text{-}coin_i, d_i, d_i')$ in its H_2-list. It responds with $H_2(M_i, \psi_i) = g^{d_i} g_1^{d_i'}$.

For a private key query on ID_i, $\mathcal{B}_2$ first recovers $(\mathsf{ID}_i, H_1\text{-}coin_i, c_i)$ from H_1-list. If $H_1\text{-}coin_i = 0$, then it responds with $\mathsf{SK}_{\mathsf{ID}_i} = g_1^{c_i} = H_1(\mathsf{ID}_i)^s$; else, it aborts.

For an individual signature query on $(M_i, \psi_i, \mathsf{ID}_i)$, $\mathcal{B}_2$ responds as follows: It first recovers $(\mathsf{ID}_i, H_1\text{-}coin_i, c_i)$ from H_1-list and $(M_i, \psi_i, H_2\text{-}coin_i, d_i, d_i')$ from H_2-list. If $H_1\text{-}coin_i = 0$, it generates a random $r \in \mathbb{Z}_n$ and outputs a signature $\theta_i = (V_i, R_i)$ such that $V_i = g_1^{c_i} \cdot H_m^r = H_1(\mathsf{ID}_i)^s \cdot H_m^r$ and $R_i = g^r$, where $H_m = H_2(M_i, \psi_i)$; else if $H_1\text{-}coin_i = 1$ and $H_2\text{-}coin_i = 0$, it aborts; else if $H_1\text{-}coin_i = 1$ and $H_2\text{-}coin_i = 1$, it generates a random $r \in \mathbb{Z}_n$ and outputs a signature as $\theta_i = (V_i, R_i)$ such that $V_i = (g_2^{c_i})^{-\frac{d_i}{d_i'}} \cdot (g^{d_i} g_1^{d_i'})^r = (g_2^{c_i})^s \cdot (g^{d_i} g_1^{d_i'})^{(r - \frac{c_i}{d_i'} \cdot \beta^*)} = H_1(\mathsf{ID}_i)^s \cdot H_m^{r'}$ and $R_i = g^r \cdot g_2^{-\frac{c_i}{d_i'}} = g^{(r - \frac{c_i}{d_i'} \cdot \beta^*)} = g^{r'}$ by letting $g_2 = g^{\beta^*}$.

For a signature query on (M_k, ψ_k), where $\psi_k = \vee_{i=1}^{a} \wedge_{j=1}^{b_i} \mathsf{ID}_{i,j}$, $\mathcal{B}_2$ responds as follows: First it recovers $(M_k, \psi_k, H_2\text{-}coin_k, d_k, d_k')$ from H_2-list and selects an index i^* such that $1 \le i^* \le a$, then it generates individual signatures for all $j \in \{1, \ldots, b_{i^*}\}$ using the above simulation method if there is no j such that $H_1\text{-}coin_{i^*,j} = 1$ and $H_2\text{-}coin_{i^*,j} = 0$; If there is no i^*, then it abort. Finally it constructs a signature σ_k using the merge algorithm.

Output: Finally, $\mathcal{A}_2$ outputs a forged DNF signature (σ^*, M^*, ψ^*), where $\sigma^* = (\sigma_1, \sigma_2, \{(C_i, \pi_i)\}_{1 \le i \le a})$ and $\psi^* = \vee_{i=1}^{a} \wedge_{j=1}^{b_i} \mathsf{ID}_{i,j}$.

If (1) the corrupted identities set $C = \{\mathsf{ID}_i\}_{1 \le i \le q_E}$ by private key queries satisfies the DNF formula ψ^*; or (2) let S be the set of identity that was requested an individual signatures queries for (M^*, ψ^*), then $S \cup C$ satisfies the DNF formula ψ^*; or (3) $\mathcal{A}$ did request a signature for a pair (M^*, ψ^*); or (4) $\mathsf{Verify}(\sigma^*, M^*, \psi^*, \mathsf{PP}) \neq$ "accept", then $\mathcal{B}_2$ stops the simulation because $\mathcal{A}_2$ was not successful.

$\mathcal{B}_2$ solves the given CDH problem as follows: Let δ_p be such that $\delta_p = 0 \mod q$ and $\delta_p = 1 \mod p$. $\mathcal{B}_2$ recovers $\{f_i\}_{1 \le i \le a}$ values using the property $C_i^{\delta_p} = 1$ if and only if $f_i = 0$. Since there is exactly one index i^* such that $f_{i^*} = 1$, it can recover the index i^* and $\{\mathsf{ID}_{i^*,j}\}_{1 \le j \le b_{i^*}}$ that were used to generate the forgery. Next it recovers $(\mathsf{ID}_{i^*,j}, H_1\text{-}coin_{i^*,j}, c_{i^*,j})$ from H_1-list for all $j \in \{1, \ldots, b_{i^*}\}$ and recovers $(M^*, \psi^*, H_2\text{-}coin, d, d')$ from H_2-list. Let B_0 be a set of index j such that $H_1\text{-}coin_{i^*,j} = 0$ and B_1 be a set of index j such that $H_1\text{-}coin_{i^*,j} = 1$. If $H_2\text{-}coin^* = 1$, it aborts; otherwise, we obtain an equation as follows:

$$
\begin{aligned}
\sigma_1 &= \prod_{j=1}^{b_{i^*}} \left(H_1(\mathsf{ID}_{i^*,j})^s \cdot H_m^{r_j} \right) \cdot h_1^z \\
&= \prod_{j \in B_0} H_1(\mathsf{ID}_{i^*,j})^s \cdot \prod_{j \in B_1} H_1(\mathsf{ID}_{i^*,j})^s \cdot \prod_{j=1}^{b_{i^*}} (g^{d^*})^{r_j} \cdot h_1^z \\
&= \prod_{j \in B_0} (g^{c_{i^*,j}})^s \cdot \prod_{j \in B_1} (g_2^{c_{i^*,j}})^s \cdot {\sigma_2}^d \cdot h_1^z
\end{aligned}
$$

$$= \quad g_1^{\sum_{j\in B_0} c_{i^*,j}} \cdot g_2^{s\cdot\sum_{j\in B_1} c_{i^*,j}} \cdot {\sigma_2}^d \cdot h_1^z.$$

Let $c_{B_0} = \sum_{j\in B_0} c_{i^*,j}$ and $c_{B_1} = \sum_{j\in B_1} c_{i^*,j}$. If $c_{B_1} \mod n = 0$, it aborts because it can't solve the CDH problem; otherwise, we obtain $g_2^s = \big(\sigma_1 \cdot \sigma_2^{-d} \cdot g_1^{-c_{B_0}} \cdot h_1^{-z}\big)^{1/c_{B_1}}$ by rearranging the above equation. Additionally, we have $e(g_1, g_2) = e(g_p^\alpha h^{r_2}, g_p^\beta h^{r_4}) = e(g_p h^{r_1}, g_p^{\alpha\beta} h^{r_2 r_4/r_1}) = e(g, g_2^s)$. Therefore, it solves the CDH problem as follows:

$$g_p^{\alpha\beta} = (g_2^s)^{\delta_p} = \big({\sigma_1}^{\delta_p} \cdot ({\sigma_2}^{\delta_p})^{-d} \cdot (g_p^\alpha)^{-c_{B_0}}\big)^{1/c_{B_1}}.$$

Analysis. For the analysis, let abort be the event that $\mathcal{B}_2$ aborts during the simulation, let forge be the event that $\mathcal{A}_2$ produces a valid forgery according to the definition of unforgeability game. We have

$$\begin{aligned} \mathsf{Adv}^{\mathsf{CDH}}_{\mathcal{B}_2} &\geq \Pr[\mathsf{forge} \wedge \neg\mathsf{abort}] = \Pr[\mathsf{forge}|\neg\mathsf{abort}] \cdot \Pr[\neg\mathsf{abort}] \\ &= \mathsf{Adv}^{\mathsf{IBDNF\text{-}UF}}_{\mathcal{A}_2} \cdot \Pr[\neg\mathsf{abort}]. \end{aligned}$$

The third equality follows from the fact that if the abort does not occur then the above simulation equals with the unforgeability game.

Let abort_E be the event that $\mathcal{B}_2$ aborts at the private key query step, abort_S be the event that $\mathcal{B}_2$ aborts at the signature query (including the individual signature query) step, abort_M be the event that $\mathcal{B}_2$ aborts when $H_2\text{-}coin^* = 1$ after $\mathcal{A}_2$ outputs a forgery, and abort_C be the event that $\mathcal{B}_2$ aborts when $c_{B_1} \mod p = 0$. Then we have

$$\begin{aligned} \Pr[\neg\mathsf{abort}] &= \Pr[\neg\mathsf{abort}_E \wedge \neg\mathsf{abort}_S \wedge \neg\mathsf{abort}_M \wedge \neg\mathsf{abort}_C] \\ &= \Pr[\neg\mathsf{abort}_E] \cdot \Pr[\neg\mathsf{abort}_S|\neg\mathsf{abort}_E] \cdot \\ &\quad \Pr[\neg\mathsf{abort}_M \wedge \neg\mathsf{abort}_C|\neg\mathsf{abort}_E \wedge \neg\mathsf{abort}_S] \\ &= \Pr[\neg\mathsf{abort}_E] \cdot \Pr[\neg\mathsf{abort}_S|\neg\mathsf{abort}_E] \cdot \Pr[\neg\mathsf{abort}_M] \cdot \\ &\quad \Pr[\neg\mathsf{abort}_C|\neg\mathsf{abort}_E \wedge \neg\mathsf{abort}_S] \\ &\geq (1-\rho_1)^{q_E} \cdot \rho_2^{q_S} \cdot (1-\rho_2) \cdot (1-1/p)\rho_1. \end{aligned}$$

The third equality follows from the facts that the events abort_M and abort_C are independent and abort_M is independent of other events. The forth inequality follows from probability calculations.

To complete the analysis, let us define $f(\rho_1, \rho_2) = A \cdot (1-\rho_1)^{q_E}\rho_1 \cdot \rho_2^{q_S}(1-\rho_2)$, where $A = (1-1/p)$. It is not hard to obtain that f is maximized at $\rho_1^{opt} = 1/(q_E+1)$ and $\rho_2^{opt} = q_S/(q_S+1)$. This gives us that $f(\rho_1^{opt}, \rho_2^{opt}) \approx A/(e^2 q_E q_S)$. By setting $\rho_1 = \rho_1^{opt}$ and $\rho_2 = \rho_2^{opt}$ in the simulation, we obtain

$$\mathsf{Adv}^{\mathsf{CDH}}_{\mathcal{B}_2} \quad \geq \quad \mathsf{Adv}^{\mathsf{IBDNF\text{-}UF}}_{\mathcal{A}_2} \cdot \frac{(1-1/p)}{e^2\ q_E\ q_S}.$$

This completes our proof. □

Theorem 5.4. *The above ID-based DNF signature scheme satisfies anonymity under the SD assumption in a bilinear group $\mathbb{G}$ of composite order n.*

Proof. Suppose there exists an adversary $\mathcal{A}$ that breaks anonymity of the above scheme. First we define two games G_0 and G_1 as the anonymity game with the following differences. In the game G_0, h is chosen uniformly from $\mathbb{G}_q$; but in the game G_1, h is chosen uniformly from $\mathbb{G}$. Let $\mathsf{Adv}_{\mathcal{A}}^{G_b}$ be the advantage $\mathcal{A}$ has over $1/2$ in the game G_b for $b \in \{0,1\}$. The proof can be obtained from Lemma 5.5 and Lemma 5.6. □

Lemma 5.5. *For any polynomially bounded adversary,* $Adv_{\mathcal{A}}^{IBDNFS\text{-}AN} - Adv_{\mathcal{A}}^{G_1} \leq 2Adv_{\mathcal{B},\mathbb{G},\mathbb{G}_q}^{SD}$.

Proof. Consider an algorithm $\mathcal{B}$ that plays the subgroup decision game. Given the subgroup decision challenge $(n, \mathbb{G}, \mathbb{G}_T, e, h)$, $\mathcal{B}$ plays the anonymity game with $\mathcal{A}$ as follows.

Setup: $\mathcal{B}$ follows the setup algorithm using the given subgroup decision challenge $(n, \mathbb{G}, \mathbb{G}_T, e, h)$. Next it gives PP to $\mathcal{A}$.

Queries: $\mathcal{B}$ can correctly respond to $\mathcal{A}$'s various queries, since it knows the master secret key.

Challenge: $\mathcal{A}$ requests a challenge with the values (M, ψ, i_0, i_1), where $\psi = \vee_{i=1}^{a} \wedge_{j=1}^{b_i} \mathsf{ID}_{i,j}$ and $1 \leq i_0 \neq i_1 \leq a$. Then $\mathcal{B}$ chooses a random bit $c \in \{0,1\}$ and computes the challenge signature σ using the sign and merge algorithms for the index i_c, and gives σ to $\mathcal{A}$.

Output: Finally, $\mathcal{A}$ outputs its guess c' for c. $\mathcal{B}$ outputs $b = 1$ if $c = c'$, $b = 0$ otherwise.

Clearly, we have $\mathsf{Adv}_{\mathcal{A}}^{G_0} = \mathsf{Adv}_{\mathcal{A}}^{\mathsf{IBDNFS\text{-}AN}}$ because h is a generator of $\mathbb{G}_q$. As we know that $\Pr[h \in \mathbb{G}] = \Pr[h \in \mathbb{G}_q] = 1/2$, we obtain

$$\begin{aligned} \mathsf{Adv}_{\mathcal{A}}^{G_0} - \mathsf{Adv}_{\mathcal{A}}^{G_1} &= \Pr[b = 1 | h \in \mathbb{G}_q] - \Pr[b = 1 | h \in \mathbb{G}] \\ &= 2\Pr[b = 1, h \in \mathbb{G}_q] - 2\Pr[b = 1, h \in \mathbb{G}] \leq 2\mathsf{Adv}_{\mathcal{B},\mathbb{G},\mathbb{G}_q}^{SD}. \end{aligned}$$

This completes our proof. □

Lemma 5.6. *For any adversary* $\mathcal{A}$*, we have that* $Adv_{\mathcal{A}}^{G_1} = 0$.

Proof. We argue that when h is chosen from $\mathbb{G}$ instead of $\mathbb{G}_q$, then the challenge signature is statistically independent of signer identity set, that is, $\mathsf{Adv}_{\mathcal{A}}^{G_1} = 0$. Consider the challenge signature $\sigma = (\sigma_1, \sigma_2, \{(C_i, \pi_i)\}_{1 \leq i \leq a})$ and determine what an adversary can deduce from it.

First, observe that σ_2 is unrelated to the choice of signer. Next, consider $C_i = (Y_i/w)^{f_i} h^{z_i}$ and $\pi_i = ((Y_i/w)^{2f_i - 1} h^{z_i})^{z_i}$ for each i. When h is a generator of $\mathbb{G}$, there exist $\tau_{i0}, \tau_{i1} \in \mathbb{Z}_n$ such that $C_i = (Y_i/w)h^{\tau_{i1}} = h^{\tau_{i0}}$. Denoting by $(\pi_i|_{f_i=b})$ the value which π_i is assigned if f_i is set to $b \in \{0,1\}$, we have $(\pi_i|_{f_i=1}) = ((Y_i/w)^1 h^{\tau_{i1}})^{\tau_{i1}} = (h^{\tau_{i0}})^{\tau_{i1}} = (h^{\tau_{i1}})^{\tau_{i0}} = ((Y_i/w)^{-1} h^{\tau_{i0}})^{\tau_{i0}} = (\pi_i|_{f_i=0})$. So the pair (C_i, π_i) is consistent with either $f_i = 0$ or $f_i = 1$ for each i, and $\mathcal{A}$ gains no information from this part of the signature. Last, we consider σ_1. If σ_2 and $\{(C_i, \pi_i)\}$ are fixed, σ_1 is the unique value satisfying

the verification equation. Specifically, letting $g_1 = g^s$, $\sigma_2 = g^r$, and $wC = g^c$ (all of which a computationally unbounded adversary can obtain), we have $\sigma_1 = g^{cs} \cdot H_m^r$. Thus this value gives no information about the signer identity set. This establishes $\mathsf{Adv}_{\mathcal{A}}^{G_1} = 0$. □

6. Construction without random oracles

In this section, we construct an ID-based DNF signature without random oracles.

Design Principle. The main idea of our construction to remove random oracles is combining Shacham-Waters ring signature scheme [22] with Waters two-level signature scheme [24]. However, a simple combination of the two schemes does not lead to a provably secure scheme, because Waters two-level signature scheme reveals the number of actual signers through the size of signature. To overcome the problem, we first construct an ID-based DNF signature where the number of identities in conjunctions is the same and then we remove the restriction.

For the construction where the number of identities in conjunctions is the same, each signer first generates Waters two-level signature by re-randomizing the private key to break linkability of the signature. Next, a representor of signers combines these signatures to generate a DNF signature associated with a DNF formula $\psi = \vee_{i=1}^{a} \wedge_{j=1}^{b} \mathsf{ID}_{i,j}$. This DNF formula ψ can be represented as a $b \times a$ matrix where each column has identities in conjunction of ψ. We use Shacham-Waters ring signature techniques for each row by constructing BGN encryptions and GOS proofs for each entry in the matrix. To guarantee that the actual signers come from the same column in the matrix, we apply our extended GOS proof technique to each column. Additionally, we construct BGN encryptions and GOS proofs for each bit value of actual signers. Since the product of BGN encryptions of each bit value is the same with the product of BGN encryption of rows in the matrix, these are redundant values. However we need these values for our security proof. The construction is described as follows.

6.1. Description

$\mathsf{Setup}(1^\lambda)$: The setup algorithm first generates a bilinear group $\mathbb{G}$ of composite order $n = pq$, where p and q are random primes of bit size $\Theta(\lambda)$. Next, it chooses random $g, g_2, u', u_1, \ldots, u_l, v', v_1, \ldots, v_m, w \in \mathbb{G}, h \in \mathbb{G}_q$, $\alpha \in \mathbb{Z}_n$, and a collision-resistant hash function $H : \{0,1\}^* \to \{0,1\}^m$. Then the public parameters PP and the master secret key MK are set by

$$\mathsf{PP} = \big(n, \mathbb{G}, \mathbb{G}_T, e, g, g_1 = g^\alpha, g_2, h, u', u_1, \ldots, u_l, v', v_1, \ldots, v_m, w, H\big),$$
$$\mathsf{MK} = g_2^\alpha.$$

$\mathsf{KeyGen}(\mathsf{ID}, \mathsf{MK}, \mathsf{PP})$: The key generation algorithm takes as input an identity $\mathsf{ID} = (\kappa_1, \ldots, \kappa_l) \in \{0,1\}^l$, the master secret key MK, and the public

parameters PP, then it chooses a random exponent $s_1 \in \mathbb{Z}_n$ and outputs

$$\mathsf{SK}_{\mathsf{ID}} = (K_1, K_2, K_3) = \big(g_2^{\alpha} \cdot (u' \prod_{i=1}^{l} u_i^{\kappa_i})^{s_1},\ g^{s_1},\ h^{s_1}\big) \in \mathbb{G}^3.$$

$\mathsf{Sign}(M, \psi, \mathsf{SK}_{\mathsf{ID}}, \mathsf{PP})$: The sign algorithm takes as input a message M, a DNF formula ψ, and a private key $\mathsf{SK}_{\mathsf{ID}}$. Next, it computes $(\mu_1, \ldots, \mu_m) = H(M, \psi)$ and chooses random exponents $s_2, r \in \mathbb{Z}_n$, then it constructs $V = K_1 \cdot (u' \prod_{i=1}^{l} u_i^{\kappa_i})^{s_2} \cdot (v' \prod_{j=1}^{m} v_j^{\mu_j})^{r}$, $S = K_2 \cdot g^{s_2}$, $T = K_3 \cdot h^{s_2}$, and $R = g^r$. By letting $s = s_1 + s_2$, it outputs an individual signature as

$$\theta = \big(V, S, T, R\big) = \big(g_2^{\alpha} \cdot (u' \prod_{i=1}^{l} u_i^{\kappa_i})^{s} \cdot (v' \prod_{j=1}^{m} v_j^{\mu_j})^{r},\ g^{s},\ h^{s},\ g^{r}\big) \in \mathbb{G}^4.$$

$\mathsf{Merge}(M, \psi, \mathsf{SS}, \mathsf{PP})$: The merge algorithm takes as input a message M, a DNF formula $\psi = \vee_{i=1}^{a} \wedge_{j=1}^{b} \mathsf{ID}_{i,j}$, a set $\mathsf{SS} = \{(\mathsf{ID}_{i^*,j}, \theta_{i^*,j})\}_{1 \le j \le b}$, where i^* is an index such that $1 \le i^* \le a$ and $\theta_{i^*,j}$ is an individual signature $(V_{i^*,j}, S_{i^*,j}, T_{i^*,j}, R_{i^*,j})$ that was generated by $\mathsf{ID}_{i^*,j}$. Let $\{f_i\}_{1 \le i \le a}$ be such that $f_i = 1$ if $i = i^*$ and $f_i = 0$ if $i \neq i^*$. To generate a signature, it proceeds as follows:

(1) First, it constructs a (unaggregate) multi-signature of the message M as $\{(\tilde{V}_j = V_{i^*,j},\ \tilde{S}_j = S_{i^*,j},\ \tilde{T}_j = T_{i^*,j},\ \tilde{R}_j = R_{i^*,j})\}_{1 \le j \le b}$ using the set SS.

(2) For all $i \in \{1, \ldots, a\}$, it chooses random $z_{i,1}, \ldots, z_{i,b} \in \mathbb{Z}_n$ and computes $\{(C_{i,j} = (Y_{i,j}/w)^{f_i} h^{z_{i,j}},\ \pi_{i,j}^{C} = ((Y_{i,j}/w)^{2f_i - 1} h^{z_{i,j}})^{z_{i,j}}\}_{1 \le j \le b}$, where $Y_{i,j} = u' \prod_{k=1}^{l} u_k^{\kappa_{i,j,k}}$ and $\mathsf{ID}_{i,j} = (\kappa_{i,j,1}, \ldots, \kappa_{i,j,l}) \in \{0,1\}^l$. Next, it computes $\{\pi_i^{col} = ((\prod_{j=1}^{b} (Y_{i,j}/w))^{2f_i - 1} h^{z_i^{col}})^{z_i^{col}}\}_{1 \le i \le a}$, where $z_i^{col} = \sum_{j=1}^{b} z_{i,j}$.

(3) For all $j \in \{1, \ldots, b\}$, it chooses random $t_{j,1}, \ldots, t_{j,l-1} \in \mathbb{Z}_n$ and sets $t_{j,l} = \sum_{i=1}^{a} z_{i,j} - \sum_{k=1}^{l-1} t_{j,k}$, then it constructs $\{(D_{j,k} = u_k^{\kappa_{i^*,j,k}} \cdot h^{t_{j,k}},\ \pi_{j,k}^{D} = (u_k^{2\kappa_{i^*,j,k} - 1} \cdot h^{t_{j,k}})^{t_{j,k}})\}_{1 \le k \le l}$ for the identity $\mathsf{ID}_{i^*,j} = (\kappa_{i^*,j,1}, \ldots, \kappa_{i^*,j,l}) \in \{0,1\}^l$.

(4) To convert $\{(\tilde{V}_j, \tilde{S}_j, \tilde{T}_j, \tilde{R}_j)\}_{1 \le j \le b}$ as a blinded one that is verifiable and anonymous, it sets $\{z_j^{row} = \sum_{i=1}^{a} z_{i,j}\}_{1 \le j \le b}$ and constructs $\{(\sigma_{1,j} = \tilde{V}_j \cdot \tilde{T}_j^{\,z_j^{row}},\ \sigma_{2,j} = \tilde{S}_j,\ \sigma_{3,j} = \tilde{R}_j)\}_{1 \le j \le b}$.

(5) The final signature is output as

$$\sigma = \big(\{(\sigma_{1,j}, \sigma_{2,j}, \sigma_{3,j}\}_{1 \le j \le b},\ \{(\{(C_{i,j}, \pi_{i,j}^{C})\}_{1 \le j \le b},\ \pi_i^{col})\}_{1 \le i \le a},\ \{(D_{j,k}, \pi_{j,k}^{D})\}_{1 \le j \le b, 1 \le k \le l}\ \big) \in \mathbb{G}^{2ab + a + 3b + 2lb}.$$

$\mathsf{Verify}(\sigma, M, \psi, \mathsf{PP})$: The verify algorithm takes as input a signature σ, a message M, and a DNF formula $\psi = \vee_{i=1}^{a} \wedge_{j=1}^{b} \mathsf{ID}_{i,j}$, then it proceeds as follows:

(1) For all $i \in \{1, \ldots, a\}$, it computes $Y_{i,j} = u' \prod_{k=1}^{l} u_k^{\kappa_{i,j,k}}$ where $\mathsf{ID}_{i,j} = (\kappa_{i,j,1}, \ldots, \kappa_{i,j,l})$, then it checks if $e(C_{i,j}, C_{i,j}/(Y_{i,j}/w)) \stackrel{?}{=} e(h, \pi_{i,j}^{C})$ for all $j \in \{1, \ldots, b\}$, and checks if $e(\prod_{j=1}^{b} C_{i,j}, \prod_{j=1}^{b}(C_{i,j}/(Y_{i,j}/w))) \stackrel{?}{=} e(h, \pi_i^{col})$.
(2) For all $j \in \{1, \ldots, b\}$, it checks if $e(D_{j,k}, D_{j,k}/u_k) \stackrel{?}{=} e(h, \pi_{j,k}^{D})$ for all $k \in \{1, \ldots, l\}$, and $u' \prod_{k=1}^{l} D_{j,k} \stackrel{?}{=} w \prod_{i=1}^{l} C_{i,j}$.
(3) Next, it computes $(\mu_1, \ldots, \mu_m) = H(M, \psi)$ and checks if $e(g, \sigma_{1,j}) \stackrel{?}{=} e(g_1, g_2) \cdot e(\sigma_{2,j}, w \prod_{i=1}^{a} C_{i,j}) \cdot e(\sigma_{3,j}, v' \prod_{i=1}^{m} v_j^{\mu_i})$ for all $j \in \{1, \ldots, b\}$.
(4) If all tests are successful, then it outputs "accept"; otherwise it outputs "reject".

It is easy to show that the above scheme satisfies the correctness as follows.

$$\begin{aligned} e(g, \sigma_{1,j}) &= e(g, g_2^{\alpha} \cdot (u' \prod_{k=1}^{l} u_k^{\kappa_{i^*,j,k}})^{s_j} \cdot (v' \prod_{k=1}^{m} v_k^{\mu_k})^{r_j} \cdot h^{s_j \cdot z_j^{row}}) \\ &= e(g_1, g_2) \cdot e(\sigma_{2,j}, w \prod_{i=1}^{a} C_{i,j}) \cdot e(\sigma_{3,j}, v' \prod_{k=1}^{m} v_k^{\mu_k}), \end{aligned}$$

where $w \prod_{i=1}^{a} C_{i,j} = (u' \prod_{k=1}^{l} u_k^{\kappa_{i^*,j,k}}) \cdot h^{z_j^{row}}$.

6.2. Security

Theorem 6.1. *The above ID-based DNF signature scheme satisfies unforgeability under the CDH assumption on $\mathbb{G}_p$ and the collision-resistant hash function H.*

Proof. In this proof, we suppose that the adversary does not cause hash collision. That is, it does not issue two message pair (M, ψ) and (M', ψ') such that $(M, \psi) \neq (M', \psi')$ but $H(M, \psi) = H(M', \psi')$. Note that if the adversary causes hash collision, it can be converted to an adversary for collision-resistant hash functions. Thus we can divide the adversary as two types according to their forgery as follows.

(1) Type-1 adversary $\mathcal{A}_1$ is one of which forgery is not such that exactly one of the exponents $\{f_i\}$ equals 1, that is, $\sum_{i=1}^{a} f_i \neq 1$.
(2) Type-2 adversary $\mathcal{A}_2$ is one of which forgery is such that exactly one of the exponents $\{f_i\}$ equals 1, that is, $\sum_{i=1}^{a} f_i = 1$.

For each type of adversary $\mathcal{A}_1$ and $\mathcal{A}_2$, we will construct algorithms $\mathcal{B}_1$ and $\mathcal{B}_2$ respectively. The proof easily follows from following two lemmas and facts that the CDH attacker can be constructed from the discrete logarithm attacker and Waters two-level signature is secure under the CDH assumption. □

Before giving the detailed proofs of the following lemmas, we give the intuitive description of them. In case of the type-2 adversary $\mathcal{A}_2$, the algorithm $\mathcal{B}_2$

can extract an ID-based signature from the output of $\mathcal{A}_2$, because $\mathcal{B}_2$ knows the factorization p, q of n and the output of $\mathcal{A}_2$ satisfies $\sum_{i=1}^{a} f_i = 1$.

In case of the type-1 adversary $\mathcal{A}_1$, the algorithm $\mathcal{B}_1$ can not use Shacham-Waters ring signature technique [22] because it can not embed the CDH instances to the public parameters. That is, it should embeds g_p^{α} and g_p^{β} of the CDH assumption to the g^{s_1} of the private key and w of the public parameters respectively to use Shacham-Waters ring signature technique. But g^{s_1} of the private key is not a fixed one, so $\mathcal{B}_1$ can not embed g_p^{α} to g^{s_1}. Instead, $\mathcal{B}_1$ solves the discrete logarithm problem of w using the $\{(D_{j,k}, \pi_{j,k}^{D})\}$ values.

Lemma 6.2. *If there exists a type-1 adversary $\mathcal{A}_1$, then there exists an algorithm $\mathcal{B}_1$ that solves the discrete logarithm problem on $\mathbb{G}_p$.*

Proof. Suppose there exists a type-1 adversary $\mathcal{A}_1$ that breaks unforgeability of the above scheme. The algorithm $\mathcal{B}_1$ that solves the discrete logarithm problem using $\mathcal{A}_1$ is given: The description of the bilinear group $\mathbb{G}$, the factorization p, q of order n, and the tuple (g_p, g_p^{α}), where g_p is a generator of $\mathbb{G}_p$. Its goal is to compute α. Then $\mathcal{B}_1$ that interacts with $\mathcal{A}_1$ is described as follows.

Setup: $\mathcal{B}_1$ selects random generators $(g, g_2, v', v_1, \ldots, v_m) \in \mathbb{G}^{m+3}$, $h \in \mathbb{G}_q$, and random exponents $(\gamma, x', x_1, \ldots, x_l, y) \in \mathbb{Z}_n^{l+3}$ with restriction that $y \bmod p \neq 0$. Next it selects a collision-resistant hash function H and sets $\mathsf{PP} = (n, \mathbb{G}, \mathbb{G}_T, e, g, g_1 = g^{\gamma}, g_2, h, u' = g^{x'}, u_1 = g^{x_1}, \ldots, u_l = g^{x_l}, v', v_1, \ldots, v_m, w = (g_p^{\alpha} h)^{y}, H)$ and $\mathsf{MK} = g_2^{\gamma}$.

Queries: $\mathcal{B}_1$ can correctly response to $\mathcal{A}_1$'s various queries, since it knows the master secret key.

Output: Finally, $\mathcal{A}_1$ outputs a forged DNF signature (σ^*, M^*, ψ^*), where $\sigma^* = (\{(\sigma_{1,j}, \sigma_{2,j}, \sigma_{3,j})\}, \{(\{(C_{i,j}, \pi_{i,j}^{C})\}, \pi_i^{col})\}, \{(D_{j,k}, \pi_{j,k}^{D})\})$ and $\psi^* = \vee_{i=1}^{a} \wedge_{j=1}^{b} \mathsf{ID}_{i,j}$.

If (1) the corrupted identities set $C = \{\mathsf{ID}_i\}_{1 \leq i \leq q_E}$ by private key queries satisfies the DNF formula ψ^*; or (2) let S be the set of identity that was requested an individual signatures queries for (M^*, ψ^*), then $S \cup C$ satisfies the DNF formula ψ^*; or (3) $\mathcal{A}$ did request a signature for a pair (M^*, ψ^*); or (4) $\mathsf{Verify}(\sigma^*, M^*, \psi^*, \mathsf{PP}) \neq$ "accept", then $\mathcal{B}_1$ stops the simulation because $\mathcal{A}_1$ was not successful.

$\mathcal{B}_1$ can solve the given problem as follows: First, it recovers $\{f_i\}_{1 \leq i \leq a}$ by setting $f_i = 0$ if $C_{i,j}^{\delta_p} = 1$ or $f_i = 1$ otherwise. Let $f = \sum_{i=1}^{a} f_i$, then $f \neq 1$, because $\mathcal{A}_1$ is a type-1 adversary. Let I be a set of index i such that $f_i = 1$ and $Y_{i,j} = u' \prod_{k=1}^{l} u_k^{\kappa_{i,j,k}}$. Then we have $w \prod_{i=1}^{a} C_{i,j} = w \cdot \prod_{i \in I} (Y_{i,j}/w) \cdot h^{z_j^{row}} = w^{1-f} \cdot \prod_{i \in I} Y_{i,j} \cdot h^{z_j^{row}}$. Since the signature is valid one, it should satisfy the verification equation by unknown identity. That is, there exists $\mathsf{ID}_j^{+} = (\kappa_{j,1}^{+}, \ldots, \kappa_{j,l}^{+})$ such that $w \prod_{i=1}^{a} C_{i,j} = Y_j^{+} \cdot h^{z_j^{row}}$, where $Y_j^{+} = u' \prod_{k=1}^{l} u_k^{\kappa_{j,k}^{+}}$, because of the equation $w \prod_{i=1}^{a} C_{i,j} = u' \prod_{k=1}^{l} D_{j,k} = u' \prod_{k=1}^{l} u_k^{\kappa_{j,k}^{+}} \cdot h^{z_j^{row}} =$

$Y_j^+ \cdot h^{z_j^{row}}$. Thus $\mathcal{B}_1$ recovers the identity ID_j^+ from $\{(D_{j,k}, \pi_{j,k}^D)\}$ by setting $\kappa_{j,k}^+ = 0$ if $D_{j,k}^{\delta_p} = 1$ or $\kappa_{j,k}^+ = 1$ otherwise. Let $F(\mathsf{ID}_{i,j}) = x' + \sum_{k=1}^{l} \kappa_{i,j,k} \cdot x_k$, where $\mathsf{ID}_{i,j} = (\kappa_{i,j,1}, \ldots, \kappa_{i,j,1})$. We obtain the following equation from the above two equations by raising δ_p to both sides

$$\left(w^{(1-f)}\right)^{\delta_p} = \left(Y_j^+ / \prod_{i \in I} Y_{i,j}\right)^{\delta_p} = \left(g^{F(\mathsf{ID}_j^+) - \Sigma_{i \in I} F(\mathsf{ID}_{i,j})}\right)^{\delta_p}.$$

Additionally, we have $w = (g_p^{\alpha} h)^y$, $y \mod p \neq 0$, and $f \neq 1$. Therefore it solves the given discrete logarithm problem as follows:

$$\alpha = (F(\mathsf{ID}_j^+) - \sum_{i \in I} F(\mathsf{ID}_{i,j})) \cdot y^{-1} \cdot (1-f)^{-1} \mod p.$$

Let $\mathsf{Adv}_{\mathcal{B}_1}^{\mathsf{DL}}$ be the advantage of $\mathcal{B}_1$ that breaks the discrete logarithm problem. Since $\mathcal{B}_1$ succeeds whenever $\mathcal{A}_1$ does, we have $\mathsf{Adv}_{\mathcal{B}_1}^{\mathsf{DL}} \geq \mathsf{Adv}_{\mathcal{A}_1}^{\mathsf{IBDNF\text{-}UF}}$. □

Lemma 6.3. *If there exists a type-2 adversary $\mathcal{A}_2$, then there exists an algorithm $\mathcal{B}_2$ that breaks the unforgeability of Waters two-level signature scheme.*

Proof. Suppose there exists a type-2 adversary $\mathcal{A}_2$ that breaks unforgeability of the above scheme. The algorithm $\mathcal{B}_2$ that forges Waters two-level signature using $\mathcal{A}_2$ is given: The description of the bilinear group $\mathbb{G}$, the factorization p, q of order n, and the public parameter of Waters two-level signature as $\tilde{\mathsf{PP}} = (p, \mathbb{G}_p, \mathbb{G}_{T_p}, e, \tilde{g}, \tilde{g}_1, \tilde{g}_2, \tilde{u}', \tilde{u}_1, \ldots, \tilde{u}_l, \tilde{v}', \tilde{v}_1, \ldots, \tilde{v}_m, H)$, where all is in subgroups of order p. Then $\mathcal{B}_2$ that interacts with $\mathcal{A}_2$ is described as follows.

Setup: $\mathcal{B}_2$ selects random generators $(f, f_2, h, \gamma', \gamma_1, \ldots, \gamma_l, \nu', \nu_1, \ldots, \nu_m) \in \mathbb{G}_q^{l+m+5}$, $w \in \mathbb{G}$, and a random exponent $\beta \in \mathbb{Z}_q^*$. Next it sets the public parameters $\mathsf{PP} = (n, \mathbb{G}, \mathbb{G}_T, e, g = \tilde{g} f, g_1 = \tilde{g}_1 f^{\beta}, g_2 = \tilde{g}_2 f_2, h, u' = \tilde{u}' \gamma', u_1 = \tilde{u}_1 \gamma_1, \ldots, u_l = \tilde{u}_l \gamma_l, v' = \tilde{v}' \nu', v_1 = \tilde{v}_1 \nu_1, \ldots, v_m = \tilde{v}_m \nu_m, w, H)$ and gives PP to $\mathcal{A}_2$.

Queries: For a private key query on ID, $\mathcal{B}_2$ first asks the private key of Waters two-level signature and receives $\tilde{\mathsf{SK}}_{\mathsf{ID}} = (\tilde{K}_1, \tilde{K}_2) \in \mathbb{G}_p^2$, then it chooses a random $s \in \mathbb{Z}_q$ and constructs the private key as $\mathsf{SK}_{\mathsf{ID}} = (K_1 = \tilde{K}_1 \cdot f^{\beta} \cdot (\gamma' \prod_{i=1}^{l} \gamma_i^{\kappa_i})^s,\ K_2 = \tilde{K}_2 \cdot f^s,\ K_3 = h^s)$.

For an individual signature query on (M, ψ, ID), $\mathcal{B}_2$ first asks the signature of Waters two-level signature and receives $\tilde{\theta} = (\tilde{\theta}_1, \tilde{\theta}_2, \tilde{\theta}_3)$, then it chooses random $s, r \in \mathbb{Z}_q$ and constructs the signature as $\theta = (V = \tilde{\theta}_1 \cdot f^{\beta} \cdot (\gamma' \prod_{i=1}^{l} \gamma_i^{\kappa_i})^s \cdot (\nu' \prod_{i=1}^{m} \nu_i^{\mu_i})^r,\ S = \tilde{\theta}_2 \cdot f^s,\ T = h^s,\ R = \tilde{\theta}_3 \cdot f^r)$.

For a signature query on (M, ψ), where $\psi = \vee_{i=1}^{a} \wedge_{j=1}^{b_i} \mathsf{ID}_{i,j}$, $\mathcal{B}_2$ first selects an arbitrary index i^* and constructs individual signatures of $\mathsf{ID}_{i^*,j}$ for all j, then it creates the final signature using the merge algorithm.

Output: Finally, $\mathcal{A}_2$ outputs a forged DNF signature pair (σ^*, M^*, ψ^*), where $\sigma^* = (\{(\sigma_{1,j}, \sigma_{2,j}, \sigma_{3,j})\}, \{(\{(C_{i,j}, \pi^C_{i,j})\}, \pi^{col}_i)\}, \{(D_{j,k}, \pi^D_{j,k})\})$ and $\psi^* = \vee^a_{i=1} \wedge^b_{j=1} \mathsf{ID}_{i,j}$.

If (1) the corrupted identities set $C = \{\mathsf{ID}_i\}_{1 \le i \le q_E}$ by private key queries satisfies the DNF formula ψ^*; or (2) let S be the set of identity that was requested an individual signatures queries for (M^*, ψ^*), then $S \cup C$ satisfies the DNF formula ψ^*; or (3) $\mathcal{A}$ did request a signature for a pair (M^*, ψ^*); or (4) $\mathsf{Verify}(\sigma^*, M^*, \psi^*, \mathsf{PP}) \neq$ "accept", then $\mathcal{B}_2$ stops the simulation because $\mathcal{A}_2$ was not successful.

$\mathcal{B}_2$ can convert the signature to Waters two-level signature as follows: First, it recovers $\{f_i\}_{1 \le i \le a}$ by setting $f_i = 0$ if $C^{\delta_p}_{i,j} = 1$ or $f_i = 1$ otherwise. Since $\mathcal{A}_2$ is a type-2 adversary, there is exactly one index i^* such that $f_{i^*} = 1$. Using the index i^*, the signers identities $\{\mathsf{ID}_{i^*,j}\}_{1 \le j \le b}$ can be reconstructed from ψ. Let the index j^* be such that neither the private key for ID_{i^*,j^*} and an individual signature on (M^*, ψ^*) by ID_{i^*,j^*} was queried by $\mathcal{A}_2$. By the conditions of $\mathcal{A}_2$'s valid forgery, the index j^* always exists. We obtain from the verification equation by raising δ_p

$$\begin{aligned} e(\tilde{g}, \sigma^{\delta_p}_{1,j^*}) &= e(\tilde{g}_1, \tilde{g}_2) \cdot e(\sigma^{\delta_p}_{2,j^*}, (w \prod_{i=1}^{a} C_{i,j^*})^{\delta_p}) \cdot e(\sigma^{\delta_p}_{3,j^*}, (v' \prod_{i=1}^{m} v^{\mu_i}_{j^*})^{\delta_p}) \\ &= e(\tilde{g}_1, \tilde{g}_2) \cdot e(\sigma^{\delta_p}_{2,j^*}, \tilde{u}' \prod_{i=1}^{l} \tilde{u}^{\kappa_i}_{j^*}) \cdot e(\sigma^{\delta_p}_{3,j^*}, \tilde{v}' \prod_{i=1}^{m} \tilde{v}^{\mu_i}_{j^*}). \end{aligned}$$

Thus $(\sigma^{\delta_p}_{1,j^*}, \sigma^{\delta_p}_{2,j^*}, \sigma^{\delta_p}_{3,j^*})$ is a valid Waters two-level signature on (M^*, ψ^*) by the identity ID_{i^*,j^*}. Then $\mathcal{B}_2$ outputs it and halts.

Let $\mathsf{Adv}^{\mathsf{W\text{-}IBS}}_{\mathcal{B}_2}$ be the advantage of $\mathcal{B}_2$ that breaks Waters two-level signature scheme. Since $\mathcal{B}_2$ succeeds whenever $\mathcal{A}_2$ does, we have $\mathsf{Adv}^{\mathsf{W\text{-}IBS}}_{\mathcal{B}_2} \ge \mathsf{Adv}^{\mathsf{IBDNF\text{-}UF}}_{\mathcal{A}_2}$. □

Theorem 6.4. *The above ID-based DNF signature scheme satisfies anonymity under the SD assumption in a bilinear group $\mathbb{G}$ of composite order n.*

Proof. Suppose there exists an adversary $\mathcal{A}$ that breaks anonymity of the above scheme. First we define two games G_0 and G_1 as the anonymity game with following differences. In the game G_0, h is chosen uniformly from $\mathbb{G}_q$; but in the game G_1, h is chosen uniformly from $\mathbb{G}$. Let $\mathsf{Adv}^{G_b}_{\mathcal{A}}$ be the advantage $\mathcal{A}$ has over $1/2$ in the game G_b for $b \in \{0, 1\}$. The proof can be obtained from following two lemmas. □

Lemma 6.5. *For all polynomially bounded adversary, $Adv^{IBDNFS\text{-}AN}_{\mathcal{A}} - Adv^{G_1}_{\mathcal{A}} \le 2Adv^{SD}_{\mathcal{B},\mathbb{G},\mathbb{G}_q}$.*

The proof is the same as the lemma 5.5.

Lemma 6.6. *For any adversary $\mathcal{A}$, we have that $Adv^{G_1}_{\mathcal{A}} = 0$.*

Proof. We argue that when h is chosen from $\mathbb{G}$ instead of $\mathbb{G}_q$, then the challenge signature is statistically independent of signer identity set, that is, $\mathsf{Adv}_{\mathcal{A}}^{G_1} = 0$. Consider the challenge signature $\sigma = (\{(\sigma_{1,j}, \sigma_{2,j}, \sigma_{3,j})\}, \{(\{(C_{i,j}, \pi_{i,j}^C)\}, \pi_i^{col})\}, \{(D_{j,k}, \pi_{j,k}^D)\})$ and determine what an adversary can deduce from it.

First, observe that $\sigma_{2,j}$ and $\sigma_{3,j}$ are unrelated to the choice of signer. Next, consider $\{(C_{i,j}, \pi_{i,j}^C)\}$, π_i^{col} and $\{(D_{j,k}, \pi_{j,k}^D)\}$ for all i, j, k. When h is a generator of $\mathbb{G}$, $C_{i,j}$ and $D_{j,k}$ are perfect commitments, and $\pi_{i,j}^C$, π_i^{col} and $\pi_{j,k}^D$ are witness-indistinguishable proofs as shown in theorem 4.2 and lemma 5.6. So these pairs give no information to the adversary. Last we consider $\sigma_{1,j}$. If $\sigma_{2,j}$, $\sigma_{3,j}$ and $\{(C_{i,j}, \pi_{i,j}^C)\}$ are fixed, $\sigma_{1,j}$ is the unique value satisfying the verification equation. Specifically, letting $g_1 = g^\alpha$, $\sigma_{2,j} = g^s$, $\sigma_{3,j} = g^r$, and $w_j \prod_{i=1}^a C_{i,j} = g^c$ (all of which a computationally unbounded adversary can obtain), we have $\sigma_1 = g_2^\alpha \cdot g^{cs} \cdot (v' \prod_{k=1}^m v_k'^{\mu_k})^r$. Thus this value gives no information about the signer identity set. This establishes $\mathsf{Adv}_{\mathcal{A}}^{G_1} = 0$. □

6.3. Removing the restriction

The restriction that the number of identities in all conjunctions should be the same can be removed by adding dummy private keys of dummy identities to the public parameters. Suppose that ψ is an original DNF formula such that the number of identities in conjunctions are not the same, then we define ψ' as the number of identities in conjunctions are the same by adding dummy identities to ψ. Note that we should not expand the number of disjunctions by adding dummy identities, because it is trivial to forge the signature of ψ' that contains a conjunction of dummy identities only. Since private keys for dummy identities are known to everyone, the individual signatures for dummy identities can be generated by the merge algorithm. Unforgeability and anonymity are follows from the facts that dummy private keys can be regarded as extracted private keys, dummy private keys alone can't satisfy ψ', and security models considers insider corruption and full key exposure.

Theorem 6.7. *The modified ID-based DNF signature scheme with dummy identities satisfies unforgeability and anonymity if the original ID-based DNF signature scheme in the section 6.1 satisfies unforgeability and anonymity.*

Proof. UNFORGEABILITY. Suppose there exists an adversary $\mathcal{A}$ that breaks unforgeability of the modified ID-based DNF signature scheme (with dummy identities). The algorithm $\mathcal{B}$ that breaks unforgeability of the original ID-based DNF signature scheme is given: the public parameters of the original one as $\tilde{\mathsf{PP}}$. Let $\mathsf{ID}_{D,1}, \ldots, \mathsf{ID}_{D,\hat{b}-1}$ be the dummy identities of the modified scheme, where $\hat{b}$ is the maximum number of identities in a conjunction. $\mathcal{B}$ asks private keys of dummy identities to the original scheme and receives $\tilde{\mathsf{SK}}_{D,1}, \ldots, \tilde{\mathsf{SK}}_{D,\hat{b}-1}$. The public parameters for the modified scheme is constructed as $\mathsf{PP} = \{\tilde{\mathsf{PP}}, \{(\mathsf{ID}_{D,i}, \tilde{\mathsf{SK}}_{D,i})\}_{1 \le i \le \hat{b}-1}\}$ and given to $\mathcal{A}$.

For $\mathcal{A}$'s private key query or individual signature query, $\mathcal{B}$ asks to the original schemes and receives $\tilde{\mathsf{SK}}$ or $\tilde{\theta}$, then gives it to $\mathcal{A}$. For $\mathcal{A}$'s signature query on DNF formula ψ' with dummy identities, $\mathcal{B}$ selects a signer set that satisfies ψ' and asks individual signature query for signer to the original scheme except dummy identities, then it construct the signature using the merge algorithm. Note that individual signatures for dummy identities can be generated from private keys of dummy identities in public parameters. Finally, $\mathcal{A}$ outputs a forgery (σ^*, M^*, ψ^*).

If $\mathcal{A}$'s forgery satisfies the conditions of valid forgery in unforgeability definition and a subset of dummy identities in the public parameters does not satisfy ψ^*, then $\mathcal{B}$ outputs (σ^*, M^*, ψ^*) as a forgery. Since the dummy private keys can be regarded as extracted private keys and a subset of dummy identities does not satisfy ψ^*, $\mathcal{B}$'s forgery is also satisfies the conditions of valid forgery. Therefore if $\mathcal{A}$ success, then $\mathcal{B}$ also success.

Anonymity. Let $\mathcal{A}$ be an adversary that breaks anonymity of the modified ID-based DNF signature scheme (with dummy identities). The algorithm $\mathcal{B}$ that breaks anonymity of the original ID-based DNF signature scheme is also simulated like above simulation for unforgeability. Additionally, for $\mathcal{A}$'s challenge query, $\mathcal{B}$ receives the response from the original scheme using $\mathcal{A}$'s challenge value, then it gives the response to $\mathcal{A}$.

Since the original scheme satisfies anonymity against full key exposure and the dummy private keys can be regarded as extracted private keys, the private keys of dummy identities does not affect anonymity of the modified scheme. Therefore if $\mathcal{A}$ success, then $\mathcal{B}$ also success. □

7. Extensions

In this section, we present two extensions of our ID-based DNF signature with random oracles.

Multiple KGCs. One drawback of ID-based system is that the master secret key is only kept in the Key Generation Center (KGC). This lags the scalability of the system, thus multiple KGCs will be needed to overcome the scalability problem. Our construction with random oracles can be modified to support multiple KGCs. The idea is extending ID-based multi-signature to support multiple KGCs and using our extended GOS proof for zero or one bit-strings to guarantee that the hidden identities come from the same signers group.

Suppose that there are n number of KGCs. Each KGC with index j publishes its public parameters as $g_j = g^{s_j}$ by selecting a random master secret key s_j, but it use the same bilinear group of composite order with other KGCs. We can restate a DNF formula $\psi = \vee_{i=1}^{a} \wedge_{j=1}^{b_i} \mathsf{ID}_{i,j}$ as $\psi' = \vee_{i=1}^{a} \wedge_{j=1}^{n} \wedge_{k=1}^{c_{i,j}} \mathsf{ID}_{i,j,k}$, where $b_i = \sum_{j=1}^{n} c_{i,j}$. The keygen and sign algorithm are the same as our ID-based DNF signature. To generate an ID-based DNF signature for multiple KGCs, individual signatures are aggregated as $(\prod_{j=1}^{n}(\prod_{k=1}^{c_{i^*,j}} H_1(\mathsf{ID}_{i^*,j,k})^{s_j}) \cdot$

$H_2(M, \psi')^{\Sigma r},\ g^{\Sigma r})$, then BGN encryptions, GOS proofs, and our extended GOS proofs are added. Next the aggregated signature converted to anonymous one for hiding the signer's identities. Note that our extended GOS proof is essential to guarantee that the hidden identities by BGN encryption come from the same signers group.

Different Messages. In ID-based DNF signatures, all actual signers should generate individual signatures on the same message. However it is natural to allow each signer to generate an individual signature for it's own message. Recently, Boyen proposed a similar signature scheme in public key system [6]. The idea to construct an ID-based DNF signature for different messages is using Gentry-Ramzan's ID-based aggregate signature scheme [11] as a building block. In the ID-based aggregate signature scheme, given n signatures on n distinct messages from n distinct users, all these signatures can be aggregated into a single short signature.

For an ID-based DNF signature for different messages, a DNF formula is newly defined as $\phi = \vee_{i=1}^{a} \wedge_{j=1}^{b_i} (\mathsf{ID}_{i,j}, M_{i,j})$, where $\mathsf{ID}_{i,j}$ is an identity and $M_{i,j}$ is a message to be signed by $\mathsf{ID}_{i,j}$. The construction is described as follows: First, the setup algorithm is almost same as our ID-based DNF signature scheme, except that it require additional hash function $H_3 : \{0,1\}^* \to \mathbb{Z}_n$. The keygen and sign algorithms are the same as Gentry and Ramzan's scheme. That is, a private key is $\mathsf{SK}_{\mathsf{ID}} = (Q^s_{\mathsf{ID},0}, Q^s_{\mathsf{ID},1})$, where $Q_{\mathsf{ID},k} = H_1(\mathsf{ID}, k)$, and an individual signature of $\mathsf{ID}_{i,j}$ is $\theta_{i,j} = (Q^s_{\mathsf{ID}_{i,j},0} \cdot (Q^s_{\mathsf{ID}_{i,j},1})^h \cdot H_w^r,\ g^r,\ w)$, where w is a shared random, $H_w = H_2(w)$ and $h = H_3(\mathsf{ID}_{i,j}, M_{i,j}, w)$. The merge algorithm first aggregates the individual signatures, then constructs BGN encryptions and GOS proofs as our ID-based DNF signature scheme. However, this scheme does not provide non-interactive property because of the shared random w.

8. Conclusion

We presented the first non-interactive ID-based DNF signatures that are secure under the CDH and subgroup decision assumptions. Our first construction uses random oracles, but it is efficient and the size of signature is compact. Our second construction does not use random oracles, but the size of signature is not compact. We note that the second construction directly yields the first ID-based ring signature to achieve signer anonymity against full key exposure without random oracles, because an ID-based ring signature scheme is a special case of an ID-based DNF signature scheme. Additionally we presented extensions of our scheme that support multiple KGCs and different messages. One interesting open problem is to construct a compact ID-based DNF signature without random oracles.

References

[1] A. Beimel, *Secure schemes for secret sharing and key distribution*, Ph. D. thesis, Israel Institute of Technology, Technion, Haifa, Israel, 1996.

[2] M. Bellare, C. Namprempre, and G. Neven, *Security proofs for identity-based identification and signature schemes*, Advances in cryptology–EUROCRYPT 2004, 268–286, Lecture Notes in Comput. Sci., 3027, Springer, Berlin, 2004.

[3] A. Bender, J. Katz, and R. Morselli, *Ring signatures*: *Stronger definitions, and constructions without random oracles*, In *TCC 2006*, 60–79, Lecture Notes in Comput. Sci., 3876, Springer-Verlang, 2007.

[4] J. Bethencourt, A. Sahai, and B. Waters, *Ciphertext-policy attribute-based encryption*, Proceedings of the IEEE Symposium on Security and Privacy, 321–334, 2007.

[5] D. Boneh, E.-J. Goh, and K. Nissim, *Evaluating 2-DNF formulas on ciphertexts*, Theory of cryptography, 325–341, Lecture Notes in Comput. Sci., 3378, Springer, Berlin, 2005.

[6] X. Boyen, *Mesh signatures*: *how to leak a secret with unwitting and unwilling participants*, Advances in cryptology–EUROCRYPT 2007, 210–227, Lecture Notes in Comput. Sci., 4515, Springer, Berlin, 2007.

[7] E. Bresson, J. Stern, and M. Szydlo, *Threshold ring signatures and applications to ad-hoc groups*, Advances in cryptology–CRYPTO 2002, 465–480, Lecture Notes in Comput. Sci., 2442, Springer, Berlin, 2002.

[8] J. C. Cha and J. H. Cheon, *An identity-based signature from gap Diffie-Hellman groups*, Public key cryptography–PKC 2003, 18–30, Lecture Notes in Comput. Sci., 2567, Springer, Berlin, 2002.

[9] S. S. M. Chow, S. M. Yiu, and L. C. K. Hui, *Efficient identity based ring signature*, ACNS 2005, 499–512, Lecture Notes in Comput. Sci., 3531, Springer-Verlang, 2005.

[10] W. Diffe and M. E. Hellman, *New directions in cryptography*, IEEE Trans. Information Theory **IT-22** (1976), no. 6, 644–654.

[11] C. Gentry and Z. Ramzan, *Identity-based aggregate signatures*, Public key cryptography–PKC 2006, 257–273, Lecture Notes in Comput. Sci., 3958, Springer, Berlin, 2006.

[12] V.Goyal, O. Pandey, A. Sahai, and B. Waters, *Attribute based encryption for fine-graned access control of encrypted data*, ACM conference on Computer and Communications Security (ACM CCS), 89–98, 2006.

[13] J. Groth, R. Ostrovsky, and A. Sahai, *Perfect non-interactive zero knowledge for NP*, Advances in cryptology–EUROCRYPT 2006, 339–358, Lecture Notes in Comput. Sci., 4004, Springer, Berlin, 2006.

[14] J. Herranz and G. Sáez, *New identity-based ring signature schemes*, ICICS 2004, 27–39, Lecture Notes in Comput. Sci., 3269, Springer-Verlang, 2004.

[15] F. Hess, *Efficient identity based signature schemes based on pairings*, Selected areas in cryptography, 310–324, Lecture Notes in Comput. Sci., 2595, Springer, Berlin, 2003.

[16] K. Lee, J. Y. Hwang, and D. H. Lee, *Non-interactive identity-based DNF signature scheme and its extensions*, ICISC 2008, Lecture Notes in Comput. Sci., Springer-Verlang, 2008.

[17] L. Nguyen, *Accumulators from bilinear pairings and applications*, Topics in cryptology–CT-RSA 2005, 275–292, Lecture Notes in Comput. Sci., 3376, Springer, Berlin, 2005.

[18] R. Ostrovsky, A. Sahai, and B. Waters, *Attribute-based encryption with non-monotonic access structures*, ACM conference on Computer and Communications Security (ACM CCS), 195–203, 2007.

[19] R. Rivest, A. Shamir, and Y. Tauman, *How to leak a secret*, Advances in cryptology–ASIACRYPT 2001 (Gold Coast), 552–565, Lecture Notes in Comput. Sci., 2248, Springer, Berlin, 2001.

[20] A. Sahai and B. Waters, *Fuzzy identity-based encryption*, Advances in cryptology–EUROCRYPT 2005, 457–473, Lecture Notes in Comput. Sci., 3494, Springer, Berlin, 2005.

[21] R. S. Sandhu, E. J. Coyne, and C. E. Youman, *Role-based access control models*, IEEE Computer **29** (1996), no 2, 38–47.
[22] H. Shacham and B. Waters, *Efficient ring signatures without random oracles*, Public key cryptography–PKC 2007, 166–180, Lecture Notes in Comput. Sci., 4450, Springer, Berlin, 2007.
[23] A. Shamir, *Identity-based cryptosystems and signaure shcemes*, CRYPTO 1984, 47–53, Lecture Notes in Comput. Sci., 196, Springer-Verlang, 1984.
[24] B. Waters, *Efficient identity-based encryption without random oralces*, EUROCRYPT 2005, 114–127, Lecture Notes in Comput. Sci., 3494, Springer-Verlang, 2005.
[25] F. Zhang and K. Kim, *ID-based blind signature and ring signature from pairings*, Advances in cryptology–ASIACRYPT 2002, 533–547, Lecture Notes in Comput. Sci., 2501, Springer, Berlin, 2002.

Kwangsu Lee
Graduate School of Information Management and Security
Korea University
Seoul 136-701, Korea
E-mail address: guspin@korea.ac.kr

Jung Yeon Hwang
Electronics and Telecommunications Research Institute
Daejeon 305-700, Korea
E-mail address: videmot@etri.re.kr

Dong Hoon Lee
Graduate School of Information Management and Security
Korea University
Seoul 136-701, Korea
E-mail address: donghlee@korea.ac.kr