Generic Framework for Attribute-Based Group Signature

Veronika Kuchta(✉), Gaurav Sharma, Rajeev Anand Sahu, and Olivier Markowitch

Universite Libre de Bruxelles, Brussels, Belgium
vkuchta@ulb.ac.be

Abstract. We first formalise a generic architecture for attribute-based signatures (ABS). Further we expand the design to the generic framework of an attribute-based group signature (ABGS), combining our generic structure of ABS with the efficient generic design of group signature proposed by Bellare et al. in Eurocrypt 2003. We also analyse security of the proposed constructions following the most standard and strong proof system, the Non-Interactive Zero Knowledge (NIZK) arguments. We emphasise that meanwhile in the process, we first achieve an attribute-based instantiation of the generic group signature scheme given by Bellare et al. and we provide a generic structure of ABGS on that block which has applications in cloud security and other cryptographic problems.

1 Introduction

In general, digital signature is a cryptographic primitive to provide signer's authentication. But there may be situations where signer's anonymity is desired for example, in anonymous electronic transaction system [24], anonymous key exchange protocol [39] etc. There have been constructions to achieve anonymity of the signer directly from the signature. The well known approaches are ring signature, group signature and blind signature. The more recent alternative attribute-based signature (ABS) [33,34] is attracting researchers due to its functionality. Their construction uses functionality of bilinear pairings but their most practical scheme is only proven secure in the generic group model. They envision their construction to readily use in multi-authority settings. In few of the extensions of ABS, attribute-based group signature (ABGS) is one of the most important and useful primitives currently being studied. In this paper, we aim to provide a generic frame to design an ABGS.

Attribute-Based Signatures. Attribute-based signature (ABS) is an extended alternative to identity-based signatures (IBS) having a set of attributes and satisfying a specific predicate.The anonymity of identity or attributes is the preliminary objective of this signature. Instead of the identity, users are associated (and specified) with certain attributes in ABS with compare to the IBS. The importance of attributes was first realized to design attribute-based encryption (ABE)

J. K. Liu and P. Samarati (Eds.): ISPEC 2017, LNCS 10701, pp. 814–834, 2017.
https://doi.org/10.1007/978-3-319-72359-4_51

to provide fine-grained access control over the encrypted data. Since the introduction of ABE [20], various proposals [4,7,9,11] have been formalized exploring different properties and advantages of ABE.

Other signature schemes have been combined to achieve the advantage of ABS with extended functionality for example attribute-based group signatures [28] and attribute-based ring signature [31] fulfill basic objectives of the underlying signature protocol with properties of ABS which yield the compact signature suitable for specific application. The proof of security in standard model is observed to be more realistic than that in random oracle. In [36] Okamoto and Takashima have formalized security setup for an ABS in the standard model. Attributes in their scheme are constrained to follow non-monotone predicates. Their scheme is based on dual pairing vector spaces and they follow the functional encryption proof technique of [30]. A threshold variation of the similar concept is presented in [38]. The threshold ABS restricts the signer to maintain a threshold number of attributes in common with the verification set of attributes. An additional featured ABS in standard model was proposed in [18] with full revocability. None of these submissions offer constant-sized signatures and usually, they all grow linearly in the number of attributes involved in the signing predicate. The first contribution with constant-sized signatures was given in [23].

Group Signatures. A group signature scheme allows an authorized member of a group to anonymously sign messages on behalf of the group. There is a group manager who can revoke the identity of the signer in case of misuse or conflict. The group manager is the only authority with this privilege. We also distinguish between static group signatures and dynamic group signatures. In a static group signature the set of members is frozen after the setup phase, whereas in the dynamic group signature, the group members can join even after the setup phase, and the setup is updated dynamically. Standard generic structures of group signature are presented by Bellare et al. in [5,6]. In [5] they formalized a generic framework of group signature addressing various properties of group signature in more standard and well defined way. They start with the *static* aspect of the group signature and initiate the idea of partially dynamic groups and fully dynamic groups. Later they proposed the generic structure of a *fully dynamic* group signature in [6]. The basic structure [6] requires a non-interactive zero knowledge (NIZK) proof system between the prover and the verifier during the signature protocol to address the verifier's witness on the signer's commitment.

The idea of group signature was introduced by Chaum and Van Heyst [15]. Ateniese et al. [3] presented an efficient and provably collision-resistant group signature scheme. In 2003, Bellare et al. [5] identified the security requirements of group signature and presented their, popularly known BMW (Bellare, Micciancio and Warinschi) security model. The two well accepted security properties for group signatures, full traceability and full anonymity were presented in this paper. Boneh et al. [10] designed short signatures in the random oracle model, using a variant of the security definition of BMW model. Security models of some well structured group signatures [13,32] are also motivated by the BMW model [5]. In these schemes, the adversary is restricted to ask queries on the tracing

of group signatures. Another efficient group signature scheme is proposed by Camenisch et al. [14] using bilinear maps. Later, Bellare et al. [6] escalated the security strength to include the group members dynamically.

Various proof techniques have been followed in different proposals of group signature. Kiayias and Yung [29] have presented a scheme which is scalable and allows dynamic adversarial joins. Security of their scheme was proved in random oracle model. Ateniese et al. [2] have proved security of their group signature in standard model. Their scheme is based on interactive assumptions. Boyen et al. [12] have followed the Groth-Ostrovsky-Sahai NIZK proof system [22], and have achieved crucial security properties viz. anonymity. In the initial proposals of group signature, the size of signature was directly dependent (linear in relation) on the number of group members. In 2008, Zhang et al. [40] presented an identity-based group signature scheme based on pairing. Size of their signature is independent of the size of group members. The group signature construction of Cheng et al. [16] has the advantages of concurrent join, immediate revocation, easy tracing and short signature length.

Attribute-Based Group Signatures. Attribute-based group signature (ABGS) is generated by a member of the group possessing certain attributes. The verifier can easily determine the role of the signer within the group. This approach is different than the usual group signatures because the signer needs to prove the ownership of certain attributes or properties. The ABGS was introduced in [28], though their primitive provides only the *anonymity* of the signer. Also, the algorithm reveals the attributes of the signer which satisfy the predicate. In a further version [27] they added the revocation property. For the practical application it is also desired to hide the attributes, used by the signer, from the verifier to achieve full anonymity. To achieve this property an ABGS scheme based on oblivious signature-based envelope (OSBE) protocol was proposed in [37]. In [17] a dynamic ABGS scheme was presented which can avoid the reissuing of attribute certificates and eliminates the pairing ratio increment depending on the number of attributes. They also discussed the application of ABGS in anonymous survey for collection of attribute statistics. Signature size is an important issue to be considered for implementation. Ali et al. [1] have suggested a constant signature sized ABGS scheme. Their scheme is independent of the number of attributes and secured in standard model. Though, there are a few constructions of the ABGS, but yet the existing literature does not cover any generic structure of ABGS for dynamic entry of the signers. In this paper, we try to put forward such a construction.

Our Contribution. We first formalize a generic architecture for ABS using the CCA-secure key encapsulation mechanism of [6] as building blocks. Further we expand the design to the generic framework of an ABGS, combining our generic structure of ABS with the efficient generic design of group signature proposed by Bellare et al. [5] in Eurocrypt 2003. Meanwhile in the process, we first achieve an attribute-based instantiation of the generic group signature scheme [5] and then construct the generic structure of our ABGS on that block. We emphasize that obtaining an attribute-based instantiation of the generic group signature

framework of [5] is itself a topic of interest since long and to the best of our knowledge our work provides first such instantiation. Furthermore, our generic ABGS system includes the dynamic setup of group members. Interestingly, unlike the all elementary constructions of dynamic group signature, our approach to dynamic ABGS does not use any access tree or credential bundle. We also analyse security of the proposed constructions following the most standard and strong proof system, the Non-Interactive Zero Knowledge (NIZK) protocol. Moreover, in contrast to the Maji's generic scheme of ABS [33,34] we achieve existential unforgeability of our scheme.

Applications. An attribute-based group signature has the following crucial applications which we propose in the following paragraphs:

Attribute-Based Messaging (ABM): As discussed in [34], the ABS schemes are useful for anonymous authentication of the sender of the attribute-based message [8]. Also, it has been described in [34] that the available classical techniques of ring signature, group signature, mesh signature are not adequate for the required security properties for such an objective. In this scenario the attribute-based signature offers desired support.

Anonymous Credential: In certain online purchase-sale activities, the merchant may want the customer to submit his/her personal details (credentials) to an external recipient (or the issuer, maybe sometimes the government). But at the same time it may be also desired that the content of these details should remain hidden from the merchant (verifier). In such circumstances, the user needs to protect his/her credentials. There have been efforts [19,25] to protect sensitive credentials in the scenarios where attributes are prime concerns. But such available schemes are either computationally expensive or can be used only when the content of certificates can be estimated. Hence, such schemes cannot be considered for the practical implementations. The attribute-based signatures following our construction can be an efficient alternate for such an objective, which offers the mechanism to convince the validity of the signature to the verifier without revealing the attributes of the signer.

Anonymous Survey: Anonymous survey is a well known practice in the electronic communication, for instance, authentication of an organizational server (which involves a group of users) before granting access to a confidential or protected resource. Approaches for such anonymous survey are proposed in [26,35] by exploiting the statistical information. For the purpose, the user sends ciphertext, encrypted with the attribute issuer's public key, to the verifier, but as it has been pointed out in [17], it is difficult to manage the statistical information for the different sets of attributes, because one attribute certificate is issued corresponding to an attribute type. It can be observed, with details in [17] that an ABGS is solution for the anonymous survey without the above difficulties.

Cloud Security: The Cloud storage services are provided by the third party hence the access to the data should be only with the legitimate user(s). Even not to the service provider. Most popular technique to achieve access control in the cloud computing is by outsourcing encrypted data over the cloud.

For the purpose, attribute-based encryption (ABE) technique has been highly suggested to be used for the encryption, due to it's functionality. But before access to the data, an authentication of the user, by the cloud server is desired. In case of group of users (in more regular situations), authentication of the appropriate user (with certain attributes) is required, specially to avoid collusion. For such authentication, ABGS offers a perfect application.

2 Preliminaries

Attribute-Based Key Encapsulation Mechanism. An attribute-based key encapsulation mechanism (AB-KEM) extends attribute-based encryption (ABE), where the ciphertext encapsulates a session key which is used to encrypt data in symmetric way.

Definition 1. *An AB-KEM consists of the following four algorithms:*

$\mathtt{Setup}(1^\lambda)$**:** *On input security parameter* 1^λ*, it outputs public parameters* $\mathtt{param}$ *and the master secret key* $\mathtt{msk}$*.*

$\mathtt{ABKKeyGen}(\mathtt{param}, \mathtt{msk}, \mathbb{A})$**:** *On input public parameters* $\mathtt{param}$*, master secret key* $\mathtt{msk}$ *and a set of attributes* $\mathbb{A}$ *it generates a corresponding secret key* $\mathtt{sk}_\mathbb{A}$*.*

$\mathtt{ABKEncaps}(\mathtt{param}, \Gamma)$**:** *On input the public parameters* $\mathtt{param}$*, a predicate* Γ*, it generates a key* $\mathtt{K}$ *and an encapsulation* $\mathtt{E}_\Gamma$ *of this key.*

$\mathtt{ABKDecaps}(\mathtt{sk}_\mathbb{A}, \mathtt{E})$**:** *On input a secret key* $\mathtt{sk}_\mathbb{A}$ *and encapsulation* $\mathtt{E}$*, it outputs either* $\mathtt{K}$ *or* $\perp$*.*

Definition 2 (ABKEM-IND-CCA Security). *The security notion of ABKEM scheme is defined for a bit* $b \in \{0,1\}$ *via the following experiment* $\mathtt{Exp}^{\mathtt{IND-CCA-b}}_{\mathcal{A}_{\mathtt{ind}},\mathtt{ABKEM}}$*:*

1. $(\mathtt{param}, \mathtt{msk}) \leftarrow \mathtt{Setup}(1^\lambda)$
2. $(\Gamma, \mathtt{state}) \leftarrow \mathcal{A}^{\mathcal{O}\mathtt{ABKKeyGen}, \mathcal{O}\mathtt{ABKDecaps}}_{\mathtt{ind}}(\mathtt{param})$
3. $(\mathtt{K}_1, \mathtt{E}^*_\Gamma) \leftarrow \mathtt{ABKEncaps}(\mathtt{param}, \Gamma)$
4. $\mathtt{K}_0 \leftarrow \mathcal{K}; \mathtt{b} \xleftarrow{\mathtt{r}} \{0,1\}$
5. $\mathtt{b}' \leftarrow \mathcal{A}^{\mathcal{O}\mathtt{ABKKeyGen}, \mathcal{O}\mathtt{ABKDecaps}}_{\mathtt{ind}}(\mathtt{state}, \mathtt{K}_\mathtt{b}, \mathtt{E}^*_\Gamma)$*. If* $b = b'$*, return* 1*, else* 0*.*

$\mathcal{O}\mathtt{ABKKeyGen}(\mathbb{A})$**:** *On input an attribute set* $\mathbb{A}$*, such that* $\Gamma(\mathbb{A}) \neq 1$ *the oracle runs* $\mathtt{sk}_\mathbb{A} \leftarrow \mathtt{ABKKeyGen}(\mathtt{param}, \mathtt{msk}, \mathbb{A})$*.*

$\mathcal{O}\mathtt{ABKDecaps}(\mathtt{E}_\Gamma, \mathbb{A})$**:** *On input an attribute set* $\mathbb{A}$ *and the encapsulation* E_Γ*, the oracle checks if* $E_\Gamma = E^*_\Gamma$*. If so it outputs* $\perp$*, otherwise it runs* $\mathtt{sk}_\mathbb{A} \leftarrow \mathtt{ABKKeyGen}(\mathtt{param}, \mathtt{msk}, \mathbb{A})$*. On input* $\mathtt{sk}_\mathbb{A}$*, it runs* $K \leftarrow \mathtt{ABKDecaps}(\mathtt{param}, \mathtt{sk}_\mathbb{A}, \mathtt{E}_\Gamma)$*. It outputs either* $\mathtt{K}$ *or* $\perp$*.*

An ABKEM scheme is indistinguishable against chosen-ciphertext attacks if for any PPT adversary $\mathcal{A}_\mathtt{ind}$ *the following advantage of is negligible:*

$$\mathtt{Adv}^{\mathtt{IND-CCA}}_{\mathcal{A}_\mathtt{ind},\mathtt{ABKEM}}(\lambda) = \left|\Pr\left[\mathtt{Exp}^{\mathtt{IND-CCA-1}}_{\mathcal{A}_\mathtt{ind},\mathtt{ABKEM}} = 1\right] - \Pr\left[\mathtt{Exp}^{\mathtt{IND-CCA-0}}_{\mathcal{A}_\mathtt{ind},\mathtt{ABKEM}} = 1\right]\right| \leq \epsilon(\lambda)$$

The ABKEM scheme has practical applications in combination with data encapsulation mechanism.

Definition 3. *A data encapsulation mechanism (DEM) consists of the following three algorithms* $\mathcal{DEM} = (\mathtt{KeyGen}, \mathtt{DEncaps}, \mathtt{DDecaps})$.

$\mathtt{DKeyGen}(1^\lambda)$**:** *On input a security parameter* 1^λ*, output the secret key* K.
$\mathtt{DEncaps}(\mathtt{K}, \mathtt{m})$**:** *On input a key* $\mathtt{K}$ *and a message* $\mathtt{m}$*, it generates a ciphertext* $\mathtt{C_D}$.
$\mathtt{DDecaps}(\mathtt{C_D}, \mathtt{K})$**:** *On input a key* $\mathtt{K}$*, ciphertext* $\mathtt{C_D}$*, it outputs either* $\mathtt{m}$ *or* $\perp$.

Definition 4 (DEM-IND-CCA Security). *The security notion of DEM scheme is defined for a bit* $b \in \{0,1\}$ *via the following experiment* $\mathtt{Exp}^{\mathtt{IND-CCA-b}}_{\mathcal{A}_{\mathtt{ind}},\mathtt{DEM}}$

1. $\mathtt{K} \leftarrow \mathtt{KeyGen}(1^\lambda)$
2. $(\mathtt{m}_0, \mathtt{m}_1, \mathtt{state}) \leftarrow \mathcal{A}^{\mathcal{O}\mathtt{DDecaps}}_{\mathtt{ind}}(1^\lambda)$
3. $\mathtt{b} \xleftarrow{r} \{0,1\};\ \mathtt{C}^*_\mathtt{D} \leftarrow \mathtt{DEncaps}(\mathtt{K}, \mathtt{m_b})$
4. $\mathtt{b}' \leftarrow \mathcal{A}^{\mathcal{O}\mathtt{ABKDecaps}}_{\mathtt{ind}}(\mathtt{state}, \mathtt{C}^*_\mathtt{D})$. *If* $b = b'$*, return* 1. *Else return* 0.

$\mathcal{O}\mathtt{DDecaps}(\mathtt{C_D})$**:** *On input secret key* $\mathtt{K}$ *and the ciphertext* $\mathtt{C_D}$*, the oracle checks whether* $\mathtt{C_D} = \mathtt{C}^*_\mathtt{D}$. *If so it returns* $\perp$*, otherwise it runs* $\mathtt{ABKDecaps}$ *and returns* $\mathtt{m}$. *A DEM scheme is indistinguishable against chosen-ciphertext attacks if for any PPT adversary* $\mathcal{A}^{\mathtt{DEM}}_{\mathtt{ind}}$ *the following advantage of is negligible:*

$$\mathtt{Adv}^{\mathtt{IND-CCA}}_{\mathcal{A}_{\mathtt{ind}},\mathtt{DEM}}(\lambda) = \left|\mathtt{Pr}\left[\mathtt{Exp}^{\mathtt{IND-CCA-1}}_{\mathcal{A}_{\mathtt{ind}},\mathtt{DEM}} = 1\right] - \mathtt{Pr}\left[\mathtt{Exp}^{\mathtt{IND-CCA-0}}_{\mathcal{A}_{\mathtt{ind}},\mathtt{DEM}} = 1\right]\right| \leq \epsilon(\lambda)$$

3 Generic Construction of Attribute-Based Signatures

In this section we provide a generic construction of attribute-based signatures employing attribute-based key encapsulation mechanism and attribute-based data encapsulation mechanism. In the following paragraph we recall the definition of the attribute-based signature scheme.

Definition 5. *An attribute-based signature (ABS) scheme consists of the following four algorithms* $\mathcal{ABS} = (\mathtt{ABSetup}, \mathtt{ABKeyGen}, \mathtt{ABSign}, \mathtt{ABVerify})$ *given an attribute universe* $\mathbb{A}$.

$\mathtt{ABSetup}(1^\lambda, 1^\mathtt{n})$**:** *This algorithm is performed by the key generation center which on input of security parameter* 1^λ *and the number of attributes* $\mathtt{n}$ *generates public parameters* $\mathtt{param}$ *and the master secret key* $\mathtt{msk}$.
$\mathtt{ABKeyGen}(\mathtt{param}, \mathtt{msk}, \mathbb{A})$**:** *This algorithm is performed by the attribute authority which takes as input public parameters param, master secret key* $\mathtt{msk}$*, user's attribute set* $\mathbb{A}$ *and generates the user's secret key* $sk_\mathbb{A}$ *corresponding to* $\mathbb{A}$.
$\mathtt{ABSign}(\mathtt{param}, \mathtt{sk}_\mathbb{A}, \mathtt{m}, \Gamma)$**:** *On input user's secret key* $\mathtt{sk}_\mathbb{A}$*, a message* $\mathtt{m}$ *and a predicate* Γ *the user generates a signature* σ.
$\mathtt{ABVerify}(\mathtt{param}, \sigma, \mathtt{m}, \Gamma)$**:** *On input* $\mathtt{param}$*, message* $\mathtt{m}$*, a signature* σ *and predicate* Γ*, the algorithm outputs either* $\mathtt{1}$ *if the signature is valid or* $\mathtt{0}$ *else.*

Security Definitions. In this paragraph we describe the main security definitions of an Attribute-Based Scheme. The first definition handles with existential unforgeability against adaptive CCA, which requires that any collusion of signers is not satisfiable to produce a signature forgery under a predicate which does not satisfy any of attribute sets in the collusion of signers. The other definition handles with privacy which guarantees that the signature does not reveal any information on the identity of the signer and on the attributes.

Definition 6 (Existential unforgeability against adaptive chosen-message attacks). *Let $\mathcal{A}_{euf}$ be a probabilistic polynomial time (PPT) adversary against chosen-message attacks who tries to make a forgery $(\mathtt{m}^*, \Gamma^*, \sigma^*)$, of a message, a predicate and a signature. Consider the following experiment* $\mathtt{Exp}^{\mathtt{EUF-CMA}}_{\mathcal{A}_{\mathtt{euf}},\mathtt{ABS}}$*:*

1. $(\mathtt{param}, \mathtt{msk}) \leftarrow \mathtt{ABSetup}(1^\lambda, 1^\mathtt{n})$
2. $(\mathtt{m}^*, \Gamma^*, \sigma^*, \mathbb{A}^*) \leftarrow \mathcal{A}^{\mathcal{O}\mathtt{ABKeyGen}(\mathtt{param},\mathtt{msk},\cdot),\mathcal{O}\mathtt{ABSign}(\mathtt{param},\mathtt{sk}_\cdot,\cdot)}_{\mathtt{euf}}(\mathtt{param})$
3. *Return* 1 *if: (a).* $\mathtt{ABVerify}(\mathtt{param}, (\mathtt{m}^*, \sigma^*), \Gamma^*) = 1$, *(b).* $\mathbb{A}^*$ *was never queried to the oracles, (c).* $\mathtt{m}^*, \Gamma^*$ *was never queried to the* $\mathcal{O}\mathtt{ABSign}$ *oracle. Else return* 0.

$\mathcal{O}\mathtt{ABKeyGen}(\mathtt{param}, \mathtt{msk}, \mathbb{A})$**:** *On input public parameters and master secret key, giving an attribute set* $\mathbb{A}$*, the oracle runs* $\mathtt{sk}_\mathbb{A} \leftarrow \mathtt{ABKeyGen}(\mathtt{param}, \mathtt{msk}, \mathbb{A})$.
$\mathcal{O}\mathtt{ABSign}(\mathtt{param}, \mathbb{A}, \mathtt{m})$**:** *On input public parameters* param*, an attribute set* $\mathbb{A}$ *and a message* m*, the oracle generates* $\mathtt{sk}_{\mathbb{A}'} \leftarrow \mathtt{ABKeyGen}(\mathtt{param}, \mathtt{msk}, \mathbb{A}')$*. Furthermore upon receiving* $\mathtt{sk}_{\mathbb{A}'}$ *it runs* $\sigma \leftarrow \mathtt{ABSign}(\mathtt{param}, \mathtt{sk}_\mathbb{A}, \mathtt{m}, \Gamma)$ *on some message* m *and some predicate* Γ' *such that* $\Gamma'(\mathbb{A}') = 1$*. It outputs a signature* σ*.*

An ABS scheme is existentially unforgeable against chosen-message attacks if for any PPT adversary $\mathcal{A}_{\mathtt{euf}}$ *the following advantage of is negligible:*

$$\mathtt{Adv}^{\mathtt{EUF-CMA}}_{\mathcal{A}_{\mathtt{euf}},\mathtt{ABS}}(\lambda) = \left|\Pr\left[\mathtt{Exp}^{\mathtt{EUF-CMA}}_{\mathcal{A}_{\mathtt{euf}},\mathtt{ABS}}(\lambda) = 1\right]\right| \leq \epsilon(\lambda)$$

Definition 7 (Attribute Privacy). *Let $\mathcal{A}_{pr}$ be a PPT adversary who tries to break the attribute privacy property of an ABS scheme. Consider the following experiment* $\mathtt{Exp}^{\mathtt{Att-Priv}}_{\mathcal{A}_{\mathtt{pr}},\mathtt{ABS}}$ *with* Γ *representing an attribute policy (=predicate):*

1. $(\mathtt{param}, \mathtt{msk}) \leftarrow \mathtt{ABSetup}(1^\lambda, 1^\mathtt{n})$
2. $(\mathbb{A}_0, \mathbb{A}_1, \Gamma^*) \leftarrow \mathcal{A}_{\mathtt{pr}}(\mathtt{param})$*, where* $|\mathbb{A}_0| = |\mathbb{A}_1|$ *such that* $(\Gamma^*(\mathbb{A}_0) = \Gamma^*(\mathbb{A}_1) = 1) \ \vee (\Gamma^*(\mathbb{A}_0) = \Gamma^*(\mathbb{A}_1) = 0)$
3. $\mathtt{sk}_{\mathbb{A}_0} \leftarrow \mathtt{ABKeyGen}(\mathtt{param}, \mathtt{msk}, \mathbb{A}_0),\ \mathtt{sk}_{\mathbb{A}_1} \leftarrow \mathtt{ABKeyGen}(\mathtt{param}, \mathtt{msk}, \mathbb{A}_1)$
4. *choose* $\mathtt{b} \in \{0,1\}, \mathtt{b}' \leftarrow \mathcal{A}^{\mathcal{O}\mathtt{ABSign}(\mathtt{param},\mathtt{sk}_{\mathbb{A}_\mathtt{b}},\cdot)}_{\mathtt{pr}}(\mathtt{param}, \mathtt{sk}_{\mathbb{A}_0}, \mathtt{sk}_{\mathbb{A}_1})$
5. *If* $\mathtt{b} = \mathtt{b}'$*, and* $|\mathbb{A}_0| = |\mathbb{A}_1|$ *return* 1*, else return* 0*.*

$\mathcal{O}\mathtt{ABSign}(\mathtt{param}, \mathbb{A})$**:** *On input public parameters* param *and an attribute set* $\mathbb{A}$ *the oracle runs* $\mathtt{sk}_\mathbb{A} \leftarrow \mathtt{ABKeyGen}(\mathtt{param}, \mathtt{msk}, \mathbb{A})$*. Furthermore upon receiving* $\mathtt{sk}_\mathbb{A}$ *it runs the signature algorithm* $\sigma \leftarrow \mathtt{ABSign}(\mathtt{param}, \mathtt{sk}_\mathbb{A}, \mathtt{m}, \Gamma)$ *on some message* m *and predicate* Gamma*. It outputs a signature* σ*.*

An ABS scheme is private if for any PPT adversary $\mathcal{A}_{\mathtt{pr}}$ *the following advantage is negligible:* $\mathtt{Adv}^{\mathtt{Attr-Priv}}_{\mathcal{A}_{\mathtt{pr}},\mathtt{ABS}}(\lambda) = \left|\Pr\left[\mathtt{Exp}^{\mathtt{Att-Priv}}_{\mathcal{A}_{\mathtt{pr}},\mathtt{ABS}}(\lambda) = 1\right] - 1/2\right| \leq \epsilon(\lambda)$.

3.1 Generic Construction of ABS Scheme

For our construction we use building blocks the attribute-based key encapsulation mechanism and the data encapsulation mechanism which are secure against chosen ciphertext attacks:

$\mathtt{ABSetup}(1^\lambda, 1^n)$**:** Given a security parameter 1^λ and the input size of the attribute set $\mathtt{n}$, it runs the `Setup` algorithm of the underlying AB-KEM scheme and public parameters `param` and master secret key `msk`. Furthermore it runs the $\mathtt{Setup}(1^\lambda)$ algorithm of the non-interactive proof system and outputs a common reference string `crs`.
$\mathtt{ABKeyGen}(\mathtt{param}, \mathtt{msk}, \mathbb{A})$**:** On input public parameters `param`, a master secret key `msk` and user's attribute set $\mathbb{A}$ it runs $\mathtt{sk}_\mathbb{A} \leftarrow \mathtt{ABKKeyGen}(\mathtt{param}, \mathtt{msk}, \mathbb{A})$ and outputs the received secret key $\mathtt{sk}_\mathbb{A}$.
$\mathtt{ABSign}(\mathtt{param}, \mathtt{crs}, \mathtt{sk}_\mathbb{A}, \mathtt{m}, \Gamma)$**:** On input public parameters `param`, common reference string `crs`, user's secret key $\mathtt{sk}_\mathbb{A}$, a message `m`, a predicate Γ, it runs $(\mathtt{E}_\Gamma, \mathtt{K}) \leftarrow \mathtt{ABKEncaps}(\mathtt{param}, \Gamma)$ and $\sigma \leftarrow \mathtt{DEncaps}(m, K)$. It uses a NIZK proof to prove the statement that a value `K` is a satisfiable output of the `ABKDecaps` algorithm under input of secret key $\mathtt{sk}_\mathbb{A}$, i.e. it shows that $\mathtt{sk}_\mathbb{A}$ is the correct key for the decapsulation algorithm on input $\mathtt{E}_\Gamma$. Note that, neither `K` nor $\mathtt{sk}_\mathbb{A}$ key will be revealed to the verifier. The output is $\hat{\sigma} = (\sigma, \pi)$.
$\mathtt{ABVerify}(\mathtt{param}, \hat{\sigma}, \Gamma)$**:** On input `param`, $\hat{\sigma} = (\sigma, \pi)$, it runs the verification part of NIZK proof, which proves the knowledge of `K` that is the output of `ABKDecaps` algorithm on input a secret key $\mathtt{sk}_\mathbb{A}$. Afterwards the verifier runs $\mathtt{m} \leftarrow \mathtt{DDecaps}(\mathtt{K}, \sigma)$. If the NIZK verification succeeds, the algorithm outputs 1, else it outputs `0`.

Description of NIZK. Let $\mathcal{P}$ and $\mathcal{V}$ be the prover and the verifier respectively of our simulation sound non-interactive zero-knowledge proof as recalled in Sect. 2. We describe the proof as follows: Our construction relies on the NIZK proof of membership in NP languages. Let `L` denote a NP language with NP-relation `R` denotes which is a subset of two arbitrary size bit strings $\{0,1\}^* \times \{0,1\}^*$ such that it requires a polynomial time algorithm to decide whether a set of a statement x and the corresponding witness `w` is an element of R or not. We specify this relation as follows: $(\mathtt{K}, \Gamma, \mathtt{ABKDecaps}(\cdot, \mathtt{E}_\Gamma), (\mathtt{sk}_\mathbb{A}, \mathbb{A}, \mathtt{R}))$, where $(\mathtt{K}, \Gamma, \mathtt{ABKDecaps}(\cdot, \mathtt{E}_\Gamma))$ is a statement of the proof and $(\mathtt{sk}_\mathbb{A}, \mathbb{A}, \mathtt{R})$ the corresponding witness with randomness R.

4 Security Analysis of ABS Scheme

Theorem 1. *Our ABS scheme is existentially UNF-CMA secure if the underlying $\mathcal{ABKEM}$ and $\mathcal{ABDEM}$ schemes are IND-CCA secure in the adaptive predicate model, the commitments used in the NIZK proof are binding and the NIZK proof itself is simulation sound.*

Proof. To prove the theorem, we assume there is an adversary $\mathcal{A}_{euf}$ against the existential UNF-CMA security of the ABS scheme. We design an adversary

$\mathcal{B}_\gamma \in (\mathcal{B}_\mathtt{K}, \mathcal{B}_\mathtt{D})$, where $\mathcal{B}_\mathtt{K}$ denotes a simulator against the IND-CCA security of ABKEM and $\mathcal{B}_\mathtt{D}$ is the corresponding algorithm against IND-CCA security of the underlying DEM scheme, respectively.

Setup: $\mathcal{B}_\gamma$ simulates $\mathcal{A}_{\mathtt{euf}}$. The simulator $\mathcal{B}_\mathtt{K}$ runs its $\mathtt{Setup}(1^\lambda)$ algorithm on input security parameter and outputs public parameters and a master secret key $\mathtt{param}, \mathtt{msk}$. $\mathcal{B}_\mathtt{K}$ forwards these values to $\mathcal{A}_{\mathtt{euf}}$.

Queries to $\mathcal{O}\mathtt{ABKeyGen}(\mathtt{param}, \mathtt{msk}, \cdot)$: Whenever $\mathcal{A}_{\mathtt{euf}}$ issues key generation queries corresponding to an attribute set $\mathbb{A}$, simulator $\mathcal{B}_\mathtt{K}$ answers to the queries using its own $\mathcal{O}\mathtt{ABKKeyGen}$ oracle on input attribute set $\mathbb{A}$. The simulator generates a list $\bar{\mathbb{A}}$ of all queried attribute sets $\mathbb{A}$. If a certain attribute set was already queried to the oracle, It forwards the received secret key $\mathtt{sk}_\mathbb{A}$ to $\mathcal{A}_{\mathtt{euf}}$.

Queries to $\mathcal{O}\mathtt{ABSign}(\mathtt{param}, \mathtt{sk}_\cdot, \mathtt{m}, \cdot)$: Whenever $\mathcal{A}_{\mathtt{euf}}$ issues signature queries on input $\mathtt{sk}_\cdot$, where "$\cdot$" describes some attribute set $\mathbb{A}'$ and a message m, simulator $\mathcal{B}_{\mathtt{K/D}}$ invokes $\mathcal{B}_\mathtt{K}$ which runs $\mathtt{sk}_\mathbb{A} \leftarrow \mathtt{ABKKeyGen}(\mathtt{param}, \mathtt{msk}, \mathbb{A})$. It chooses a predicate Γ, such that $\Gamma(\mathbb{A}) = 1$ and runs $(\mathtt{E}_\Gamma, \mathtt{K}) \leftarrow \mathtt{ABKEncaps}(\mathtt{param}, \Gamma)$. On input $\mathtt{K}$ simulator invokes $\mathcal{B}_\mathtt{D}$, which runs $\sigma \leftarrow \mathtt{DEncaps}(\mathtt{m}, \mathtt{K})$ on the received message $\mathtt{m}$. The simulator generates a list $\mathtt{M}$ of all received messages. If the received message $\mathtt{m}$ is already in the list, simulator aborts the simulation. Using the secret key $\mathtt{sk}_\mathbb{A}$ it runs the prover protocol of the NIZK proof and outputs $\hat{\sigma} = (\sigma, \pi)$, where π is the NIZK proof inspired by [21] and given by $\mathtt{P}(\mathtt{K}, \Gamma, \mathtt{ABKDecaps}(\cdot, \mathtt{E}_\Gamma), (\mathtt{sk}_\mathbb{A}, \mathbb{A}, \mathtt{R}))$. Finally, $\mathcal{B}_\gamma$ forwards $\hat{\sigma}$ to $\mathcal{A}_{\mathtt{euf}}$.

Output: Finally, $\mathcal{A}_{\mathtt{euf}}$ outputs $(\sigma^*, \mathtt{m}^*, \Gamma^*)$ s.t. the following properties hold:

(a) $\mathtt{ABVerify}(\mathtt{param}, \mathtt{m}^*, \sigma^*) = 1$. To check this equality, the simulator takes $\mathtt{m}^*, \sigma^*$, invokes the $\mathcal{B}_\mathtt{K}$ part of the simulation algorithm. $\mathcal{B}_\mathtt{K}$ queries its own $\mathcal{O}\mathtt{ABKeyGen}$ oracle on input previously chosen attribute set $\mathbb{A}$. Upon receiving the secret key $\mathtt{sk}_\mathbb{A}$ it queries its $\mathcal{O}\mathtt{ABKDecaps}$ oracle on input the secret key. The oracle outputs symmetric key $\mathtt{K}$. Taking the symmetric key the simulator invokes $\mathcal{B}_\mathtt{D}$ part of the algorithm to firstly run $\sigma \leftarrow \mathtt{DEncaps}(\mathtt{m}^*, \mathtt{K})$. It checks whether the received signature is equal to the received challenge signature σ^*. Furthermore it issues a query to its own $\mathcal{O}\mathtt{DDecaps}$ oracle on input $\mathtt{K}$ and checks the received message $\mathtt{m}$ is equal to the challenge message $\mathtt{m}^*$. If both are equal, the verification succeeds and either $\mathcal{B}_\mathtt{K}$ or $\mathcal{B}_\mathtt{D}$ outputs 1 to $\mathcal{A}_{euf}$.
(b) $\mathbb{A}^*$ was never queried to the both oracles.
(c) $(\mathtt{m}^*, \Gamma^*)$ was never queried to the $\mathcal{O}\mathtt{ABSign}$ oracle.

Otherwise, $\mathcal{B}_{\mathtt{K/D}}$ breaks the IND-CCA security as follows: If $\mathbb{A}^*$ was queried to the key generation oracle $\mathcal{O}\mathtt{ABKeyGen}$ the simulator would be able to recover the queried attribute set from the attribute set $\bar{\mathbb{A}}$, which would break the IND-CCA security of the underlying AB-KEM, DEM schemes. Assuming that $\mathbb{A}$ has been queried to the key generation oracle $\mathcal{O}\mathtt{ABKeyGen}$ and output a new secret key $\mathtt{sk}'_\mathbb{A}$ it would break the binding assumption of commitment scheme, which would mean that it is possible to find two different opening values aka randomizers to open the commitment to two different blinded secret keys. Since the binding

property of our commitments used in the NIZK proof is guaranteed, we claim that an adversary succeeds in breaking the binding property with a negligible probability. In order to prove simulation soundness of the NIZK proof, we consider the following game where an adversary $\mathcal{A}_{\mathtt{ss}}$ against simulation soundness of NIZK is playing against a challenger (who is represented by the adversary against our ABS scheme):

1. $(\mathtt{param}, \mathtt{crs}, \mathtt{msk}) \leftarrow \mathtt{Setup}(1^{\lambda}, 1^{\mathtt{n}})$
2. $\mathtt{sk}_{\mathbb{A}} \leftarrow \mathtt{ABKKeyGen}(\mathtt{param}, \mathtt{msk})$

   End for (a). $\mathtt{m}^*, \Gamma^*, \sigma^* \leftarrow \mathcal{A}_{\mathtt{euf}}^{\mathcal{O}\mathtt{ABKeyGen}(\mathtt{param},\mathtt{msk},\cdot),\mathcal{O}\mathtt{ABSign}(\mathtt{param},\mathtt{sk}_{\cdot},\cdot)}(\mathtt{param}, \mathtt{msk})$
   (b). $(\mathtt{K}, \mathtt{E}_{\Gamma^*}) \leftarrow \mathtt{ABKEncaps}(\mathtt{param}, \Gamma^*)$, (c). $(\mathtt{C}_{\mathtt{D}}) \leftarrow \mathtt{ABKEncaps}(\mathtt{K}, \mathtt{m}^*)$, $\mathtt{C}_{\mathtt{D}} := \sigma^*$
   (d). $\pi \leftarrow \mathtt{SIM}(\mathtt{prove}, \mathtt{crs}, \mathtt{param}, \mathtt{m}^*, \sigma^*, \mathtt{sk}_{\mathbb{A}}, \Gamma^*)$,
   Make oracle queries to $\mathcal{O}\mathtt{ABKeyGen}$ and $\mathcal{O}\mathtt{ABSign}$. Run $\mathtt{Verify}(\mathtt{param}, \sigma, \pi)$.
   If $\mathcal{A}_{\mathtt{euf}}$ outputs a valid σ, π', output $(\mathtt{param}, \mathtt{crs}, \sigma, \pi')$.

We say that $\mathcal{A}_{\mathtt{ss}}$ wins the experiment, if $\mathcal{A}_{\mathtt{euf}}$ did not query $\mathcal{O}\mathtt{DDecaps}$ on (σ, π'). Advantage of $\mathcal{A}_{\mathtt{ss}}$ is given by:

$$\begin{aligned}\mathtt{Adv}^{\mathtt{Sim-Sound}}_{\mathcal{A}_{\mathtt{ss}},\mathtt{ABS}} = &\left|\mathtt{Pr}\left[\mathtt{Exp}^{\mathtt{IND-CCA-1}}_{\mathcal{A}_{\mathtt{ind}},\mathtt{ABKEM}} = 1\right] - \mathtt{Pr}\left[\mathtt{Exp}^{\mathtt{IND-CCA-0}}_{\mathcal{A}_{\mathtt{ind}},\mathtt{ABKEM}} = 1\right]\right| \\ &+ \left|\mathtt{Pr}\left[\mathtt{Exp}^{\mathtt{IND-CCA-1}}_{\mathcal{A}_{\mathtt{ind}},\mathtt{DEM}} = 1\right] - \mathtt{Pr}\left[\mathtt{Exp}^{\mathtt{IND-CCA-0}}_{\mathcal{A}_{\mathtt{ind}},\mathtt{DEM}} = 1\right]\right|\end{aligned}$$

Finally we conclude that the advantage of an adversary $\mathcal{A}_{euf}$ is given by the following combined inequation:

$$\mathtt{Adv}^{\mathtt{E-UNF}}_{\mathcal{A}_{\mathtt{euf}},\mathtt{ABS}} \leq \mathtt{Adv}^{\mathtt{Sim-Sound}}_{\mathcal{A}_{\mathtt{ss}},\mathtt{ABS}} + \mathtt{Adv}^{\mathtt{IND-CCA}}_{\mathcal{A}_{\mathtt{ind}},\mathtt{ABKEM}} + \mathtt{Adv}^{\mathtt{IND-CCA}}_{\mathcal{A}_{\mathtt{ind}},\mathtt{DEM}}$$

Theorem 2. *Our ABS scheme is attribute anonymous if the underlying DEM scheme is IND-CCA secure and the underlying NIZK proof is simulation-sound and computationally zero-knowledge provable.*

Proof. Due to the page limit, we skip a detailed proof of this theorem and refer to the full version of this paper.

5 Generic Construction of Attribute-Based Group Signature

In this section we present a generic construction of attribute-based group signature (ABGS) scheme. We assume a scenario where the group manager is not involved in the key generation process for a new member. Provided by the group managers secret key, she is available to trace the malicious signer only. The key issuing functionality is processed by another entity, the key issuing entity. The reason for separating the roles of group manager and key issuer is to disable a group manager to create a signature forgery or to collude with other members.

Definition 8. *An ABGS scheme consists of the following six algorithms:*

$\mathtt{Setup}(1^\lambda, 1^n)$**:** *On input security parameter* 1^λ *and the size of attribute set* 1^n *the central authority runs this randomized algorithm to output public parameters* param *and master secret key* msk*.*

$\mathtt{ABGKeyGen}(\mathtt{param}, \mathtt{msk}, \mathbb{A}_i)$**:** *On input public parameters* param*, master secret key* msk *and an attribute set* $\mathbb{A}_i$ *of user* i*, it generates a group public key* gpk*, an issuing key* ik *for enrolling new group members by a certificate issuing entity and a group master secret key* gmsk *for opening the signature by the group manager to trace and identify the signers. Furthermore the algorithm generates user's* i *secret key* $\mathtt{sk}_{\mathbb{A}_i}$ *corresponding to the user's attribute set* $\mathbb{A}_i$ *and* $\mathtt{pk}_i$*.*

$\langle \mathtt{Join}(\mathtt{param}, \mathtt{gpk}, \mathtt{pk}_i, \mathtt{sk}_{\mathbb{A}_i}) \rangle, \langle \mathtt{Issue}(\mathtt{param}, \mathtt{pk}_i, \mathtt{ik}) \rangle$**:** *This is an interactive protocol allowing new members to join the group. The protocol is run between a user* $\mathtt{U}_i$ *and an certificate issuing entity* KIE*. The certificate issuing outputs a certificate* $\mathtt{cert}_i$ *for user* $\mathtt{U}_i$ *and stores user's public key* $\mathtt{pk}_i$ *in a registration table.*

$\mathtt{ABGSign}(\mathtt{param}, \mathtt{sk}_{\mathbb{A}_i}, \mathtt{m}, \Gamma)$**:** *On input public parameters* param*, member's secret key* $\mathtt{sk}_{\mathbb{A}_i}$*, a predicate* Γ *and a message* m *it returns a signature* σ*.*

$\mathtt{ABGVery}(\mathtt{param}, \mathtt{gpk}, \sigma, \Gamma)$**:** *On input public parameters* param*, group public key* gpk*, a signature* σ *and the predicate* Γ*, the deterministic algorithm verifies the validity of the signature and outputs* 1 *if the signature is valid, else outputs* 0*.*

$\mathtt{ABGOpen}(\mathtt{param}, \mathtt{gmsk}, \sigma)$**:** *On input public parameters* param*, group master secret key* gmsk *and a signature* σ *it outputs either the attribute set* $\mathbb{A}$ *or* $\perp$*.*

5.1 Security Definitions

In this section we provide the core security properties of an ABGS scheme. We are focusing in this paper on the following three security notions: attribute and user anonymity, traceability and non-frameability.

Fully anonymity of users. In general, anonymity property of an ABGS scheme means that it is hard for an adversary apart from the group manager to recover the identity of the signer. Similar to the construction in [6], we guarantee collusion incapacity of an adversary with group members by providing the secret keys of all group members to the adversary. Furthermore we give an adversary access to the open oracle in order to allow him to see the results of previous openings. In the following definition we consider an adversary $\mathcal{A}_{uan}$, who wants to break the fully user anonymity property, and a bit b associated with the security experiment. We assume an adversary acting in two stages where in the first stage - the so called find stage - it takes as input the user's secret keys $sk_{\mathbb{A}_i}$ and group public key *gpk* and outputs two identities i_0, i_1 and a message m.

Definition 9 (User anonymity). *An ABGS scheme preserves user anonymity if the advantage of an adversary in winning* $\mathtt{Exp}^{\mathtt{U-ANO-b}}_{\mathcal{A}_{\mathtt{uan}},\mathtt{ABGS}}(1^\lambda, 1^n)$ *is negligible:*

1. $(\mathtt{param}, \mathtt{msk}) \leftarrow \mathtt{Setup}(1^\lambda, 1^n)$
2. $(\mathtt{gpk}, \mathtt{ik}, \mathtt{gmsk}, \mathtt{sk}_{\mathbb{A}_i}, \mathtt{sk}_i, \mathtt{pk}_i) \leftarrow \mathtt{ABGKeyGen}(\mathtt{param}, \mathtt{msk})$, $\tilde{\mathtt{usk}} := \{\mathtt{sk}_{\mathbb{A}_i}, \mathtt{sk}_i\}_{i \in [n]}$

3. $(\mathtt{state}, \mathtt{i}_0, \mathtt{i}_1, \mathtt{m}, \Gamma) \leftarrow \mathcal{A}_{\mathtt{uan}}^{\mathcal{O}^{\mathtt{ABGOpen}(\cdot)}}(\mathtt{find}, \mathtt{param}, \mathtt{gpk}, \tilde{\mathtt{usk}})$
4. *Choose* $\mathtt{b} \in 0, 1$; $\sigma^* \leftarrow \mathtt{ABGSign}(\mathtt{param}, \mathtt{sk}_{\mathbb{A},\mathtt{i_b}}, \mathtt{m}, \Gamma)$
5. $\mathtt{b}' \leftarrow \mathcal{A}_{\mathtt{uan}}^{\mathcal{O}^{\mathtt{ABGOpen}(\cdot)}}(\mathtt{guess}, \mathtt{state}, \sigma^*)$

$\mathcal{O}\mathtt{ABGOpen}(m, \sigma)$**:** *The adversary calls this oracle with some message* m *and a signature* σ*. The oracle runs* $\mathtt{Open}(gmsk, \sigma)$ *to receive index* i *which allows to trace malicious signer.*

An ABGS scheme is fully anonymous if for any PPT adversary $\mathcal{A}_{\mathtt{uan}}$ *the following advantage is negligible:*

$$\mathtt{Adv}^{\mathtt{U-ANO}}_{\mathcal{A}_{\mathtt{uan}},\mathtt{ABGS}}(\lambda) = \left|\mathtt{Pr}\left[\mathtt{Exp}^{\mathtt{U-ANO-1}}_{\mathcal{A}_{\mathtt{uan}},\mathtt{ABGS}}(\lambda) = 1\right] - \mathtt{Pr}\left[\mathtt{Exp}^{\mathtt{U-ANO-0}}_{\mathcal{A}_{\mathtt{uan}},\mathtt{ABGS}}(\lambda) = 1\right]\right| \leq \epsilon(\lambda)$$

Attribute anonymity. This property means that a verifier should be able to verify a signature corresponding to a predicate without revealing the attribute set. Attribute anonymity is especially useful if there is only one group member with a certain attribute, which helps tracing back to the identity of the user.

Definition 10 (Attribute anonymity)
$\mathtt{Exp}^{\mathtt{At-Ano-b}}_{\mathcal{A}_{\mathtt{at-ano}},\mathtt{ABGS}}(1^\lambda, 1^n)$*:*

1. $(\mathtt{param}, \mathtt{msk}) \leftarrow \mathtt{Setup}(1^\lambda, 1^\mathtt{n})$
2. $(\mathbb{A}_0, \mathbb{A}_1, \Gamma^*) \leftarrow \mathcal{A}_{\mathtt{at-ano}}(\mathtt{param})$*, where* $|\mathbb{A}_0| = |\mathbb{A}_1|$ *such that* $(\Gamma^*(\mathbb{A}_0) = \Gamma^*(\mathbb{A}_1) = 1) \vee (\Gamma^*(\mathbb{A}_0) = \Gamma^*(\mathbb{A}_1) = 0)$
4. $\mathtt{sk}_{\mathbb{A}_0} \leftarrow \mathtt{ABGKeyGen}(\mathtt{param}, \mathtt{msk}, \mathbb{A}_0)$, $\mathtt{sk}_{\mathbb{A}_1} \leftarrow \mathtt{ABGKeyGen}(\mathtt{param}, \mathtt{msk}, \mathbb{A}_1)$
5. $\mathtt{b}' \leftarrow \mathcal{A}_{\mathtt{at-ano}}^{\mathcal{O}^{\mathtt{ABGSign}(\mathtt{param}, \mathtt{sk}_{\mathbb{A}_\mathtt{b}}, \cdot)}}(\mathtt{param}, \mathtt{sk}_{\mathbb{A}_0}, \mathtt{sk}_{\mathbb{A}_1})$
6. *If* $\mathtt{b} = \mathtt{b}'$ *and* $|\mathbb{A}_0| = |\mathbb{A}_1|$ *return* 1*, else return* 0*.*

$\mathcal{O}\mathtt{ABGSign}(\mathtt{param}, \mathbb{A}, \cdot)$**:** *On input public parameters* $\mathtt{param}$ *and an attribute set* $\mathbb{A}'$ *the oracle runs* $\mathtt{sk}_{\mathbb{A}} \leftarrow \mathtt{ABKeyGen}(\mathtt{param}, \mathtt{msk}, \mathbb{A}')$*. Furthermore upon receiving* $\mathtt{sk}_{\mathbb{A}'}$ *it runs* $\sigma \leftarrow \mathtt{ABSign}(\mathtt{param}, \mathtt{sk}_{\mathbb{A}}, \mathtt{m})$ *on some message* $\mathtt{m}$*. It outputs a signature* σ*. An ABGS scheme is attribute-anonymous if for any PPT adversary* $\mathcal{A}_{\mathtt{at-ano}}$ *the following advantage is negligible:*

$$\mathtt{Adv}^{\mathtt{At-ANO}}_{\mathcal{A}_{\mathtt{at-ano}},\mathtt{ABGS}}(\lambda) = \left|\mathtt{Pr}\left[\mathtt{Exp}^{\mathtt{At-ANO-1}}_{\mathcal{A}_{\mathtt{at-ano}},\mathtt{ABGS}}(\lambda) = 1\right] - \mathtt{Pr}\left[\mathtt{Exp}^{\mathtt{At-ANO-0}}_{\mathcal{A}_{\mathtt{at-ano}},\mathtt{ABGS}}(\lambda) = 1\right]\right|.$$

Full-Traceability. We assume that in case of malicious behavior, signer's identity can be revealed by the group manager using manager's secret key. In other words it means that no collusion of group members should enable to create a valid signature which cannot be opened by the group manager. As mentioned in [5], the group manager could be dishonest and accuse an user in malicious behavior. In order to avoid this dishonest behavior of the user we can ask the group manager to also output a proof together with the identity $\mathtt{i}$, after running the $\mathtt{Open}$ algorithm. The verification of the proof can take place by running an additional algorithm - $\mathtt{Judge}$ - on input a signature σ, identity i ant proof π.

Definition 11 (Full-Traceability). *We say that an ABGS scheme is fully traceable if the advantage of an adversary* $\mathcal{A}_{\mathtt{tr}}$ *to win the following experiment* $\mathtt{Exp}^{\mathtt{Full-Trace}}_{\mathcal{A}_{\mathtt{f-trace}},\mathtt{ABGS}}(1^\lambda, 1^n)$ *is negligible.*

1. $(\mathtt{param}, \mathtt{msk}) \leftarrow \mathtt{Setup}(1^{\lambda}, 1^{n})$
2. $(\mathtt{gpk}, \mathtt{ik}, \mathtt{gmsk}, \mathtt{sk}_{\mathbb{A}_i}, \mathtt{sk}_i, \mathtt{pk}_i) \leftarrow \mathtt{ABGKeyGen}(\mathtt{param}, \mathtt{msk})$, $\tilde{\mathtt{usk}} := \{\mathtt{sk}_{\mathbb{A}_i}, \mathtt{sk}_i\}_{i \in [n]}$
3. $(\mathtt{m}, \sigma) \leftarrow \mathcal{A}_{\mathtt{f-trace}}^{\mathcal{O}\mathtt{ABGSign}(\cdot), \mathcal{O}\mathtt{ABGKeyGen}(\cdot), \mathcal{O}\mathtt{Open}}(\mathtt{gpk}, \mathtt{gmsk})$
   If $\mathtt{ABGVery}(\mathtt{param}, \mathtt{gpk}, \sigma) = 0$, *return* 0. *If* $\mathtt{Open}(\mathtt{param}, \mathtt{gms}, \sigma) = \perp$, *return* 1.
   Let $\mathcal{C}$ *denote the list of all opened identities. If* $\mathtt{Open}(\mathtt{param}, \mathtt{gmsk}, \sigma) = \mathtt{i}$ *and* $\mathtt{i} \notin \mathcal{C}$, *then return* 1, *else return* 0.

$\mathcal{O}\mathtt{ABGSign}(\mathtt{param}, \mathbb{A}, \cdot)$: *On input public parameters* param *and an attribute set* $\mathbb{A}$ *the oracle runs* $\mathtt{sk}_{\mathbb{A}'} \leftarrow \mathtt{ABKeyGen}(\mathtt{param}, \mathtt{msk}, \mathbb{A}')$. *Furthermore upon receiving* $\mathtt{sk}_{\mathbb{A}}$ *it runs* $\sigma \leftarrow \mathtt{ABSign}(\mathtt{param}, \mathtt{sk}_{\mathbb{A}}, \mathtt{m})$ *on some message* m. *It outputs a signature* σ.
$\mathcal{O}\mathtt{ABGKeyGen}(\mathtt{param}, \mathtt{msk}, \mathbb{A})$: *On input public parameters and master secret key, giving an attribute set* $\mathbb{A}$, *the oracle runs* $(pk, \mathtt{sk}_{\mathbb{A}}) \leftarrow \mathtt{ABGKeyGen}(\mathtt{param}, \mathtt{msk}, \mathbb{A})$, *where pk denotes all the public key of the* ABGKeyGen. *It outputs a tuple consisting of public keys and secret key* $\mathtt{sk}_{\mathbb{A}}$.
$\mathcal{O}\mathtt{Open}(\mathtt{param}, \mathtt{gmsk}, \sigma)$: *On input* $\mathtt{param}, \mathtt{gmsk}, \sigma$, *returns* $\mathtt{i} \leftarrow \mathtt{Open}(\mathtt{param}, \mathtt{gmsk}, \sigma)$

An ABGS scheme is attribute-anonymous if for any PPT adversary $\mathcal{A}_{\mathtt{at-ano}}$ *the following advantage is negligible:*

$$\mathtt{Adv}^{\mathtt{Full-Trace}}_{\mathcal{A}_{\mathtt{f-trace}}, \mathtt{ABGS}}(\lambda) = \left| \Pr\left[\mathtt{Exp}^{\mathtt{Full-Trace}}_{\mathcal{A}_{\mathtt{f-trace}}, \mathtt{ABGS}}(\lambda) = 1 \right] \right| \leq \epsilon(\lambda).$$

Non-frameability. This security notion means that an adversary is not able to prove that some honest user created a valid signature. This property requires that it is impossible for two or more colluding users to produce a signature which would trace back to the non-colluded group member. As showed by Bellare et al. [5], non-frameability property is considered to be a version of collusion resistance. The two properties are the same in the sense that non-frameability prevents to create a signature which would be opened by a group manager and trace to a different member of the group. An ABGS scheme that is fully-traceable, is automatically secure against framing. Bellare et al. [5] showed how to convert an adversary against framing into an adversary against full-traceability.

5.2 Construction

Using such building blocks as attribute-based key encapsulation and data encapsulation mechanisms, public key encryption scheme, digital signature scheme and strong one-time signature scheme we merge the two generic constructions (of an ABS and a GS schemes) recalled and constructed in this paper and introduce a new generic construction of an attribute-based group signature scheme. We achieve the first instantiation of the construction technique from [6] applied to the attribute-based groups signature scheme. Furthermore, we use the NIZK proof from [21] which was successfully implemented in the construction of Bellare's static [5] and dynamic group [6]

schemes. We not that for a proper implementation of the NIZK proof we require and additional building block of a secure strong one-time group signature (SOTS) scheme. Assume that a SOTS scheme consists of three algorithms $\mathtt{KeyGen}_{sots}, \mathtt{Sign}_{sots}, \mathtt{Verify}_{sots}$ with the corresponding outputs $(\mathtt{vk_{sots}}, \mathtt{sk_{sots}}) \leftarrow \mathtt{KeyGen_{sots}}(1^\lambda)$; $\sigma_{\mathtt{sots}} \leftarrow \mathtt{Sign_{sots}}(\mathtt{m}, \mathtt{sk_{sots}})$; $1/0 \leftarrow \mathtt{Verify_{sots}}(\mathtt{vk_{sots}}, \mathtt{m}, \sigma_{\mathtt{sots}})$.

Definition 12. *A generic attribute-based groups signature scheme consists of the following six algorithms:*

$\mathtt{Setup}(1^\lambda, 1^n)$**:** *The algorithm is run by a key generation center. It runs the setup algorithm of AB-KEM algorithm* $(\mathtt{param}, \mathtt{msk}) \leftarrow \mathtt{Setup}(1^\lambda, 1^n)$.

$\mathtt{ABGKeyGen}(\mathtt{param}, \mathtt{msk}, \mathbb{A}_i)$**:** *On input public parameters* param *and master secret key* msk *user's attribute set* $\mathbb{A}_i$ *it runs* $\mathtt{sk}_{\mathbb{A}_i} \leftarrow \mathtt{ABKeyGen}(\mathtt{param}, \mathtt{msk})$. *Furthermore it runs the key generation algorithm of the underlying digital signature scheme and outputs a pair of secret and public key* $(\mathtt{sk_i}, \mathtt{pk_i}) \leftarrow \mathtt{KeyGen_s}(1^\lambda)$ *where the secret key represents the other part of user's secret key. The algorithm sets user's secret key equal to* $\mathtt{usk[i]} = (\mathtt{sk}_{\mathbb{A}_i}, \mathtt{sk_i})$ *and user's public key as* $\mathtt{pk_i}$. *The algorithm runs the key generation algorithm of the signature scheme for the second time to generate a secret and a public key for the certificate issuing entity,* $(\mathtt{sk_s}, \mathtt{pk_s}) \leftarrow \mathtt{KeyGen}(1^\lambda)$. *Lastly it runs the key generation algorithm of the underlying public key encryption scheme* $(\mathtt{sk_e}, \mathtt{pk_e}) \leftarrow \mathtt{KeyGen}(1^\lambda)$, *where the secret key* sk_e *represents the group manager's secret key to open the signature and to trace malicious signers. The algorithm also runs the* Setup *algorithm of the underlying NIZK proofs and outputs a common reference string* crs *with randomness* r. *Group public key is set equal to* $gpk = (param, crs, r, pk_e, pk_s)$.

$\mathtt{Join}(\langle \mathtt{param}, \mathtt{gpk}, \mathtt{ik}, \mathtt{pk_i}, \mathbb{A}_i \rangle, \langle \mathtt{param}, \mathtt{gpk}, \mathtt{pk_i}, \mathtt{usk[i]} \rangle)$**:** *This interactive protocol is initiated by the user* $\mathtt{U_i}$ *who takes its verification key* $\mathtt{pk_i}$ *and signs it by running the signature algorithm of the underlying digital signature scheme, using its secret key* $\mathtt{usk[i]}$, *s.t.* $\sigma_\mathtt{i} \leftarrow \mathtt{Sign}(\mathtt{usk[i]}, \mathtt{pk_i})$. *This signing procedure guarantees non-frameability against corrupt users. The user sends then her public key* $\mathtt{pk_i}$ *and the signature* $\sigma_\mathtt{i}$ *to the certificate issuing entity (CIE) in order to receive a certificate which would provide eligibility of a group member. CIE signs the public key using its own secret key,* $\mathtt{cert_i} \leftarrow \mathtt{Sign}(\mathtt{sk_s}, \langle \mathtt{i}, \mathtt{pk_i} \rangle)$, *such that the final signature serves as a certificate for user* $\mathtt{U_i}$. *The issuer stores* $(\mathtt{pk_i}, \sigma_i)$ *in the registration table.*

$\mathtt{ABGSign}(\mathtt{param}, \mathtt{usk[i]}, \mathtt{m}, \Gamma)$**:** *On input public parameters* param, *user's secret key* $\mathtt{usk[i]}$, *a message* m *and a predicate* Γ *the user runs* $(vk_{sots}, sk_{sots}) \leftarrow \mathtt{KeyGen}_{sots}(1^\lambda)$. *The verification key* $\mathtt{vk_{sots}}$ *will be a part of the NIZK proof. The user signs* $\mathtt{vk_{sots}}$ *using it's secret key* $\mathtt{usk[i]}$ *as follows: First, it runs* $(\mathtt{K}, \mathtt{E}_\Gamma) \leftarrow \mathtt{ABKEncaps}(\mathtt{param}, \Gamma)$ *of the underlying AB-KEM scheme. Taking* K *and a message* m *it runs the encapsulation algorithm of the underlying DEM scheme,* $\hat{\sigma} = \mathtt{DEncaps}(\mathtt{vk_{sots}}, \mathtt{K})$. *Using encryption algorithm of the underlying encryption scheme it outputs a ciphertext encrypting user's certificate, and signature* $\hat{\sigma}$, *i.e.* $\mathtt{C} \leftarrow \mathtt{Encrypt}(\mathtt{pk_e}, \langle \mathtt{i}, \mathtt{pk_i}, \mathtt{cert_i}, \hat{\sigma}, \mathtt{R} \rangle)$, *where* R *is a randomness used for the witness of NIZK proof. This encryption procedure prevents someone to create its own public and secret key pair* $\mathtt{pk'_i}, \mathtt{sk'_i}$. *The user runs*

NIZK1 proof π_1 from the ABS scheme described in Sect. 3.1. to prove the statement that $\mathtt{K}$ *is a satisfiable output of* $\mathtt{ABKDecaps}$ *algorithm on input a secret key* $\mathtt{sk}_{\mathbb{A}_i}$*. Furthermore, user runs NIZK2 proof π_2 which proves that the certificate* $\mathtt{cert_i}$ *is a signature under CIE's public key* $\mathtt{pk_s}$*. Lastly, taking as input a message m, verification key* $\mathtt{vk_{sots}}$*, ciphertext π and the corresponding proof $\tilde{\pi} = (\pi_1, \pi_2)$, user runs the signature algorithm of the underlying SOTS scheme and outputs* $\sigma_{\mathtt{sots}} \leftarrow \mathtt{Sign}(\mathtt{m}, \mathtt{vk_{sots}}, \mathtt{C}, \tilde{\pi})$*. The final signature is equal to* $\tilde{\sigma} = (\mathtt{C}, \tilde{\pi} = (\pi_1, \pi_2), \sigma_{\mathtt{sots}})$.

$\mathtt{ABGVery}(\mathtt{param}, \mathtt{gpk}, \tilde{\sigma})$**:** *On input* $\mathtt{param}$*,* $\mathtt{gpk} = (\mathtt{pk_e}, \mathtt{pk_s})$ *and* $\tilde{\sigma} = (\mathtt{C}, \tilde{\pi}, \sigma_{\mathtt{sots}})$*, the verification is followed by the* $\mathtt{Verify}$ *algorithm of NIZK proof verifying the SOTS signature* $\sigma_{\mathtt{sots}}$ *on input* $(\mathtt{m}, \mathtt{vk_{sots}}, \mathtt{C}, \tilde{\pi}, \sigma_{\mathtt{sots}})$ *as input.*

$\mathtt{ABGOpen}(\mathtt{param}, \mathtt{gmsk}, \tilde{\sigma})$**:** *Parse* $\mathtt{gmsk} = (\lambda, \mathtt{pk_e}, \mathtt{sk_e}, \mathtt{pk_s})$ *and* $\tilde{\sigma} = (\mathtt{C}, \tilde{\pi}, \sigma_{\mathtt{sots}})$*. If* $\mathtt{Verify} = 0$ *in both of the NIZK proofs, return* 0*, else decrypt the ciphertext by running* $\mathtt{Decrypt}(\mathtt{sk_e}, \mathtt{C})$ *and receive the string* $\langle \mathtt{i}, \mathtt{pk_i}, \mathtt{cert_i}, \hat{\sigma} \rangle$*. Return* $\mathtt{i}$.

Description of NIZK. Since the first NIZK proof π_1 is inherited from our generic ABS construction, we present the details of the second NIZK proof π_2 only. The witness relation of this proof π_2 which is used in our construction, is specified as $\mathtt{P}((\mathtt{pk_e}, \mathtt{vk_{sots}}, \mathtt{m}, \mathtt{C}), (\mathtt{sk}_{\mathbb{A}_i}, \mathbb{A}_\mathtt{i}, \mathtt{cert_i}, \sigma_\mathtt{i}, \mathtt{R}))$, where $(\mathtt{pk_e}, \mathtt{vk_{sots}}, \mathtt{m}, \mathtt{C})$ is a proof statement and $(\mathtt{sk}_{\mathbb{A}_i}, \mathbb{A}_\mathtt{i}, \mathtt{cert_i}, \sigma_\mathtt{i}, \mathtt{R})$ the corresponding witness with randomness R. Simulation soundness of this proof is guaranteed due to the following justification: The prover who is also the signer picks random keys of SOTS scheme $(\mathtt{vk_{sots}}, \mathtt{sk_{sots}})$, where $\mathtt{vk_{sots}}$ becomes a part of π_2. The corresponding SOTS signature $\sigma_{\mathtt{sots}}$ defined above becomes a part of the verifier algorithm of π_2. The common reference string of this proof contains user's public key pk_i. In the porver part a user proves that the above defined statement is an element of NP language $\mathtt{L}$ or he knows the signature $\sigma_\mathtt{i}(\mathtt{vk_{sots}})$. It will be guaranteed that an adversary cannot forge a signature on a new $\mathtt{vk_{sots}}$, which means that the creation of a valid NIZK proof fails. Since it is obvious to distinguish whether a NIZK proof is real or simulated we need to hide the signature $\hat{\sigma}(\mathtt{vk_{sots}})$, defined in the $\mathtt{ABGSign}$ algorithm. To achieve perfect soundness and the scenarios where a computationally unbounded adversary would be able to forge signatures under its public key $\mathtt{pk_i}$, we need to provide an encryption of some random element in CRS. For a valid NIZK proof both need to be encrypted, a signature $\hat{\sigma}(\mathtt{vk_{sots}})$ and a trivial element, which encrypts to $\mathtt{C_{triv}}$. The encryption of the witness $(\mathtt{sk}_{\mathbb{A}_i}, \mathbb{A}_\mathtt{i}, \mathtt{cert_i}, \hat{\sigma}, \mathtt{R})$ guarantees zero-knowledge property.

6 Security Analysis

Theorem 3. *Our generic ABGS scheme is fully-anonymous and fully traceable if the underlying NIZK proof is simulation sound and zero-knowledge provable*

Proof. In order to prove the theorem we are using the following lemmas:

Lemma 1. *If the underlying AB-KEM, DEM and public key encryption systems are IND-CCA secure and the NIZK1 and NIZK2 proofs are simulation sound and zero-knowledge, then our ABGS scheme is fully-anonymous.*

Lemma 2. *Our ABGS scheme is attribute anonymous if the underlying DEM scheme is IND-CCA secure and the underlying NIZK proofs is simulation-sound and computationally zero-knowledge provable.*

Lemma 3. *If the underlying AB-KEM, DEM systems are IND-CCA secure, digital signature scheme is unforgeable against chosen message attacks and the NIZK1 and NIZK2 proofs are simulation sound, then our ABGS scheme is fully-traceable.*

Proof of Lemma 1. Let $\mathcal{A}_{\mathtt{uan}}$ be an adversary against the user's full-anonymity in the ABGS scheme. We design an adversary $\mathcal{B}_\gamma \in (\mathcal{B}_{\mathtt{K}}, \mathcal{B}_{\mathtt{D}}, \mathcal{B}_{\mathtt{pke}}, \mathcal{B}_{\mathtt{SOTS}})$ against the IND-CCA security of the ABKEM or IND-CCA security of the DEM schemes, respectively, where γ indicates that the adversary is either running against the IND-CCA security of the ABKEM scheme or against the IND-CCA security of the DEM scheme. We show how to construct $\mathcal{B}_\gamma$ to simulate $\mathcal{A}_{uan}$.

Setup: $\mathcal{B}_\gamma$ simulates $\mathcal{A}_{\mathtt{uan}}$. Simulator $\mathcal{B}_{\mathtt{K}}$ runs its $\mathtt{Setup}(1^\lambda)$ algorithm on input security parameter and outputs public parameters and a master secret key $\mathtt{param}, \mathtt{msk}$. $\mathcal{B}_{\mathtt{K}}$ forwards these values to $\mathcal{A}_{\mathtt{uan}}$. To simulate the remained public and secret keys of user, issuer and group manager, $\mathcal{A}_{\mathtt{uan}}$ invokes an adversary against the underlying public key encryption scheme $\mathcal{B}_{\mathtt{pke}}$. The detailed description of this adversary is given in the following experiment:

1. $(\mathtt{vk_{sots}}, \mathtt{sk_{sots}}) \leftarrow \mathtt{KeyGen_{sots}}(1^\lambda)$
2. $(\mathtt{pk_e}, \mathtt{sk_e}) \leftarrow \mathtt{KeyGen_e}(1^\lambda)$
3. $(\mathtt{pk_s}, \mathtt{sk_s}) \leftarrow \mathtt{KeyGen_s}(1^\lambda)$
4. $(\mathtt{crs}, \mathtt{R}) \leftarrow \mathtt{SIM}(\mathtt{generate}, \lambda)$
5. Set $\mathtt{gpk} = (\lambda, \mathtt{R}, \mathtt{pk_e}, \mathtt{pk_s}, \mathtt{vk_{sots}})$
   For all $\mathtt{i} \in [\mathtt{n}]$ run $(\mathtt{pk_i}, \mathtt{sk_i}) \leftarrow \mathtt{KeyGen_s}(1^\lambda), \mathtt{cert_i} \leftarrow \mathtt{Sign}(\mathtt{sk_s}, \langle \mathtt{i}, \mathtt{pk_i} \rangle)$.
   Make oracle queries to $\mathcal{O}\mathtt{KeyGen}$ and $\mathcal{O}\mathtt{Decrypt}$ of the PKE scheme.

Queries to $\mathcal{O}\mathtt{ABGOpen}(\cdot,\cdot)$: Whenever $\mathcal{A}_{\mathtt{uan}}$ calls its opening oracle on input a message $\mathtt{m}$ and a signature σ, algorithm $\mathcal{B}_\gamma$ simulates these opening queries and sets $\sigma = C_D$ of the underlying DEM scheme. $\mathcal{B}_{\mathtt{D}}$ runs its key generation algorithm on input security parameter λ and outputs a symmetric key $\mathtt{K} \leftarrow \mathtt{KeyGen}(1^\lambda)$. Taking the key $\mathtt{K}$ and the received message $\mathtt{m}$, $\mathcal{B}_{\mathtt{D}}$ runs its data encapsulation algorithm $\mathtt{C_D} \leftarrow \mathtt{DEncaps}(\mathtt{vk_{sots}}, \mathtt{K})$. It compares whether $\mathtt{C_D} = \sigma$, if so it forwards this query on $\mathtt{C_D}$ to its own $\mathcal{O}\mathtt{DDecaps}$ oracle and receives either $\mathtt{m}$ or $\perp$. In case the oracle's output is $\mathtt{m}$, it returns 1 to $\mathcal{A}_{uan}$ adversary.

To simulate user's attribute-based secret key, algorithm $\mathcal{B}_{\mathtt{K}}$ is invoked and queries it's own $\mathcal{O}\mathtt{ABKKeyGen}$ on input public parameters $\mathtt{param}$ and master secret key $\mathtt{msk}$. The output is $\mathtt{sk}_{\mathbb{A}_\mathtt{i}} \leftarrow \mathcal{O}\mathtt{ABKKeyGen}$. The simulator sets $\mathtt{usk}[\mathtt{i}] = (\mathtt{sk_i}, \mathtt{sk}_{\mathtt{A_i}})$.

Challenge: When adversary $\mathcal{A}_{\mathtt{uan}}$ outputs $(\mathtt{state}, \mathtt{i_0}, \mathtt{i_1}, \mathtt{m})$, it picks $\mathtt{b} \in \{0, 1\}$, computes signature $\sigma_\mathtt{b} \leftarrow \mathtt{ABGSign}(\mathtt{param}, \mathtt{usk}[\mathtt{i_b}], \mathtt{m}, \Gamma)$, simulator invokes its $\mathcal{B}_{\mathtt{pke}}$, who randomly creates two messages m. Simulator invokes $\mathcal{B}_{\mathtt{K}}$ of the key encapsulation algorithm on input $(\mathtt{param}, \Gamma)$, i.e. $(\mathtt{E}_\Gamma,) \leftarrow \mathtt{ABKEncaps}(\mathtt{param}, \Gamma)$.

Furthermore $\mathcal{A}_{uan}$ invokes the $\mathcal{B}_{SOTS}$ algorithm to simulates the keys of SOTS scheme by running $(\mathtt{vk}_{\mathtt{sots}}, \mathtt{sk}_{\mathtt{sots}}) \leftarrow \mathtt{KeyGen}_{\mathtt{SOTS}}$. The verification key $\mathtt{vk}_{\mathtt{sots}}$ will be a part of the NIZK proof. $\mathcal{B}_{\mathtt{K}}$ signs $\mathtt{vk}_{\mathtt{sots}}$ using simulated secret key $\mathtt{usk}[\mathtt{i}]$, where the secret key simulation is given by a random guess with probability $1/|\mathcal{K}|$. The guessing probability reduces $\mathcal{B}'_{\mathtt{K}}$s advantage to win the game. If the guess of the keys does not match with the real secret key, the simulation aborts. The signature procedure continues as follows: Taking $\mathtt{K}$ and the verification key $\mathtt{vk}_{\mathtt{sots}}$ as a message, it runs encapsulation algorithm of the underlying DEM scheme, $\hat{\sigma} = \mathtt{DEncaps}(\mathtt{vk}_{\mathtt{sots}}, \mathtt{K})$. Furthermore $\mathcal{B}_{\mathtt{pke}}$ of the underlying encryption scheme is invoked, which outputs a ciphertext encrypting user's certificate $\mathtt{cert}_{\mathtt{i_b}}$, and signature $\hat{\sigma}$, i.e. $\mathtt{C} \leftarrow \mathtt{Encrypt}(\mathtt{pk}_{\mathtt{e}}, \langle \mathtt{i_b}, \mathtt{pk}_{\mathtt{i_b}}, \mathtt{cert}_{\mathtt{i_b}}, \hat{\sigma}, \mathtt{R} \rangle)$, where $\mathtt{R}$ is a randomness used in the NIZK proof. Finally taking as input a message m, verification key $\mathtt{vk}_{\mathtt{sots}}$, ciphertext π and the corresponding proof $\tilde{\pi} = (\pi_1, \pi_2)$, $\mathcal{B}_p$ runs the signature algorithm of the underlying SOTS scheme and outputs $\sigma_{\mathtt{sots}} \leftarrow \mathtt{Sign}(\mathtt{m}, \mathtt{vk}_{\mathtt{sots}}, \mathtt{C}, \tilde{\pi})$. Furthermore, simulator runs the NIZK proof π_1 from the ABS scheme to prove the knowledge of $\mathtt{K}$ that is the output of $\mathtt{ABKDecaps}$ algorithm on input a secret key $\mathtt{sk}_{\mathbb{A}}$. $\mathcal{B}_{\mathtt{pke}}$ runs the NIZK proof π_2 that the certificate $\mathtt{cert}_{\mathtt{i}}$ is a signature under CIE's public key $\mathtt{pk}_{\mathtt{s}}$. The final signature is equal to $\tilde{\sigma} = (\mathtt{C}, \tilde{\pi} = (\pi_1, \pi_2))$. We note that whenever $\mathcal{A}_{\mathtt{uan}}$ submits a query $(\mathtt{C}, \pi')$ to the opening oracle, simulator invokes $\mathcal{B}_{\mathtt{pke}}$ and forwards the query to its decryption oracle. Finally it outputs a bit $\mathtt{b}$ and terminates the simulation.

Distinguisher for Zero-Knowledge. Distinguisher involved in the NIZK proof is given in the following description of the algorithm $\mathcal{D}(\mathtt{choose}, \lambda, \mathtt{R})$:

1. $(\mathtt{vk}_{\mathtt{sots}}, \mathtt{sk}_{\mathtt{sots}}) \leftarrow \mathtt{KeyGen}_{\mathtt{sots}}(1^{\lambda})$
2. $(\mathtt{pk}_{\mathtt{e}}, \mathtt{sk}_{\mathtt{e}}) \leftarrow \mathtt{KeyGen}_{\mathtt{e}}(1^{\lambda})$
3. $(\mathtt{pk}_{\mathtt{s}}, \mathtt{sk}_{\mathtt{s}}) \leftarrow \mathtt{KeyGen}_{\mathtt{s}}(1^{\lambda})$
4. $(\mathtt{crs}, \mathtt{R}) \leftarrow \mathtt{SIM}(\mathtt{generate}, \lambda)$
5. Set $\mathtt{gpk} = (\lambda, \mathtt{R}, \mathtt{pk}_{\mathtt{e}}, \mathtt{pk}_{\mathtt{s}}, \mathtt{vk}_{\mathtt{sots}})$

   End for **(a).** $(\mathtt{state}, \mathtt{i}_0, \mathtt{i}_1, \mathtt{m}^*, \mathtt{vk}^*_{\mathtt{sots}}, \Gamma^*) \leftarrow \mathcal{A}^{\mathcal{O}^{\mathtt{ABGOpen}(\cdot)}}_{\mathtt{uan}}(\mathtt{param}, \mathtt{msk}, \cdot)$;
   (b). $\mathtt{b} \in \{0,1\}, \mathtt{R} \in \{0,1\}^{\lambda}$; **(c).** $\mathtt{C}_{\mathtt{D}} \leftarrow \mathtt{ABKEncaps}(\mathtt{K}, \mathtt{vk}^*_{\mathtt{sots}})$, $\mathtt{C}_{\mathtt{D}} := \hat{\sigma}^*$;
   (d). $\mathtt{C}^* \leftarrow \mathtt{Encrypt}(\mathtt{pk}_{\mathtt{e}}, \langle \mathtt{i_b}, \mathtt{pk}_{\mathtt{i_b}}, \mathtt{cert}_{\mathtt{i_b}}, \hat{\sigma}^*, \mathtt{R} \rangle)$;
   (e). $\sigma_{\mathtt{sots}} \leftarrow \mathtt{Sign}_{\mathtt{sots}}(\mathtt{m}^*, \mathtt{vk}^*_{\mathtt{sots}}, \mathtt{C}^*, \tilde{\pi}^*)$.

We note that distinguisher $\mathcal{D}$ can answer any queries submitted by $\mathcal{A}_{\mathtt{uan}}$, because it is in possession of group manager's secret key, which can be used to open the signatures. The output of the challenge phase is a signature gives as $(\mathtt{pk}_{\mathtt{e}}, \mathtt{pk}_{\mathtt{s}}, \mathtt{m}, \mathtt{C})$ together with a witness. In the second stage, distinguisher takes as input a proof $\tilde{\pi} = (\pi_1, \pi_2)$ and creates a groups signature $\tilde{\sigma} = (\mathtt{C}, \tilde{\pi}, \sigma_{\mathtt{sots}})$ and outputs it to the adversary $\mathcal{A}_{\mathtt{uan}}$. Finally, $\mathcal{D}$ outputs the same value as $\mathcal{A}_{\mathtt{uan}}$.

Soundness of NIZK proof. In order to prove simulation soundness of the NIZK proof, we consider the following game where an adversary $\mathcal{A}_{\mathtt{ss}}$ against simulation soundness of NIZK is playing against a challenger, who is represented by the adversary against our ABGS scheme:

1. $(\mathtt{vk}_{\mathtt{sots}}, \mathtt{sk}_{\mathtt{sots}}) \leftarrow \mathtt{KeyGen}_{\mathtt{sots}}(1^{\lambda})$
2. $(\mathtt{pk}_{\mathtt{e}}, \mathtt{sk}_{\mathtt{e}}) \leftarrow \mathtt{KeyGen}_{\mathtt{e}}(1^{\lambda})$
3. $(\mathtt{pk}_{\mathtt{s}}, \mathtt{sk}_{\mathtt{s}}) \leftarrow \mathtt{KeyGen}_{\mathtt{s}}(1^{\lambda})$
4. $(\mathtt{crs}, \mathtt{R}) \leftarrow \mathtt{SIM}(\mathtt{generate}, \lambda)$
5. Set $\mathtt{gpk} = (\lambda, \mathtt{R}, \mathtt{pk}_{\mathtt{e}}, \mathtt{pk}_{\mathtt{s}}, \mathtt{vk}_{\mathtt{sots}})$

End for **(a).** $\mathtt{m}^*, \Gamma^*, \sigma^* \leftarrow \mathcal{A}_{\mathtt{uan}}^{\mathcal{O}^{\mathtt{ABGOpen}(\mathtt{param},\mathtt{gmsk},\cdot)}}(\mathtt{param}, \mathtt{msk}, \cdot)$;
(b). $(\mathtt{K}, \mathtt{E}_{\Gamma^*}) \leftarrow \mathtt{ABKEncaps}(\mathtt{param}, \Gamma^*)$; **(c).** $\mathtt{C}_{\mathtt{D}} \leftarrow \mathtt{ABKEncaps}(\mathtt{K}, \mathtt{vk}^*_{\mathtt{sots}})$, $\mathtt{C}_{\mathtt{D}} = \hat{\sigma}^*$; **(d).** $\mathtt{C} \leftarrow \mathtt{Encrypt}(\mathtt{pk}_{\mathtt{e}}, \langle \mathtt{i}_{\mathtt{b}}, \mathtt{pk}_{\mathtt{i}_{\mathtt{b}}}, \mathtt{cert}_{\mathtt{i}_{\mathtt{b}}}, \sigma_{\mathtt{b}}, \mathtt{R} \rangle)$;
(e). $\sigma_{\mathtt{sots}} \leftarrow \mathtt{Sign}_{\mathtt{sots}}(\mathtt{m}^*, \mathtt{vk}^*_{\mathtt{sots}}, \mathtt{C}^*, \tilde{\pi}^*)$;
(f). $\pi \leftarrow \mathtt{SIM}(\mathtt{prove}, \mathtt{crs}, \mathtt{param}, \mathtt{m}^*, \sigma^*, \mathtt{sk}_{\mathbb{A}}, \Gamma^*)$.
Make oracle queries to $\mathcal{O}\mathtt{ABKeyGen}$ to simulate user's attribute-based secret key $\mathtt{sk}_{\mathbb{A}_{\mathtt{i}}}$. Run $\mathtt{Verify}(\mathtt{param}, \sigma_{\mathtt{sots}}, \pi, \mathtt{C})$. If $\mathcal{A}_{\mathtt{uan}}$ outputs a valid $\sigma_{\mathtt{sots}}, \pi', \mathtt{C}$, output $(\mathtt{param}, \mathtt{crs}, \sigma_{\mathtt{sots}}, \pi', \mathtt{C})$.

Due to the page limit we provide only the final result of adversary's success. For the detailed analysis of this proof, we refer to the later full version of this paper. Finally we conclude that the advantage of an adversary $\mathcal{A}_{\mathtt{uan}}$ is given by the following combined inequation:

$$\mathtt{Adv}^{\mathtt{U-ANO}}_{\mathcal{A}_{\mathtt{uan}},\mathtt{ABGS}} \leq \mathtt{Adv}^{\mathtt{Sim-Sound}}_{\mathcal{A}_{\mathtt{ss}},\mathtt{ABGS}} + \mathtt{Adv}^{\mathtt{IND-CCA}}_{\mathcal{A}_{\mathtt{ind}},\mathtt{KEM}} + + \mathtt{Adv}^{\mathtt{IND-CCA}}_{\mathcal{A}_{\mathtt{ind}},\mathtt{DEM}} + \mathtt{Adv}^{\mathtt{IND-CCA}}_{\mathcal{A}_{\mathtt{ind}}\mathtt{PKE}} + \mathtt{Adv}^{\mathtt{ZK}}_{\mathcal{A}_{\mathtt{zk}},\mathtt{NIZK}}$$

7 Conclusion

In this paper, we first presented a generic design for Attribute-Based Signatures (ABS). Further we have extended our construction to the generic scheme of any Attribute-Based Group Signature (ABGS), combining our generic structure of ABS with an existing proposal of generic group signature. We have also analyzed security of the proposed constructions following the most standard and comparatively efficient proof system, the Non-Interactive Zero Knowledge Proof of Knowledge approach.

References

1. Ali, S.T., Amberker, B.B.: Short attribute-based group signature without random oracles with attribute anonymity. In: Thampi, S.M., Atrey, P.K., Fan, C.-I., Perez, G.M. (eds.) SSCC 2013. CCIS, vol. 377, pp. 223–235. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-40576-1_22
2. Ateniese, G., Camenisch, J., Hohenberger, S., de Medeiros, B.: Practical group signatures without random oracles. IACR Cryptology ePrint Archive, 2005:385 (2005)
3. Ateniese, G., Camenisch, J., Joye, M., Tsudik, G.: A practical and provably secure coalition-resistant group signature scheme. In: Bellare, M. (ed.) CRYPTO 2000. LNCS, vol. 1880, pp. 255–270. Springer, Heidelberg (2000). https://doi.org/10.1007/3-540-44598-6_16

4. Attrapadung, N., Libert, B., de Panafieu, E.: Expressive key-policy attribute-based encryption with constant-size ciphertexts. In: Catalano, D., Fazio, N., Gennaro, R., Nicolosi, A. (eds.) PKC 2011. LNCS, vol. 6571, pp. 90–108. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-19379-8_6
5. Bellare, M., Micciancio, D., Warinschi, B.: Foundations of group signatures: formal definitions, simplified requirements, and a construction based on general assumptions. In: Biham, E. (ed.) EUROCRYPT 2003. LNCS, vol. 2656, pp. 614–629. Springer, Heidelberg (2003). https://doi.org/10.1007/3-540-39200-9_38
6. Bellare, M., Shi, H., Zhang, C.: Foundations of group signatures: the case of dynamic groups. In: Menezes, A. (ed.) CT-RSA 2005. LNCS, vol. 3376, pp. 136–153. Springer, Heidelberg (2005). https://doi.org/10.1007/978-3-540-30574-3_11
7. Bethencourt, J., Sahai, A., Waters, B.: Ciphertext-policy attribute-based encryption. In: Security and Privacy, SP 2007, pp. 321–334. IEEE (2007)
8. Bobba, R., Fatemieh, O., Khan, F., Gunter, C.A., Khurana, H.: Using attribute-based access control to enable attribute-based messaging. In: ACSAC 2006, pp. 403–413. IEEE (2006)
9. Boneh, D., Boyen, X.: Efficient selective-ID secure identity-based encryption without random oracles. In: Cachin, C., Camenisch, J.L. (eds.) EUROCRYPT 2004. LNCS, vol. 3027, pp. 223–238. Springer, Heidelberg (2004). https://doi.org/10.1007/978-3-540-24676-3_14
10. Boneh, D., Boyen, X.: Short signatures without random oracles. In: Cachin, C., Camenisch, J.L. (eds.) EUROCRYPT 2004. LNCS, vol. 3027, pp. 56–73. Springer, Heidelberg (2004). https://doi.org/10.1007/978-3-540-24676-3_4
11. Boneh, D., Boyen, X., Goh, E.-J.: Hierarchical identity based encryption with constant size ciphertext. In: Cramer, R. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 440–456. Springer, Heidelberg (2005). https://doi.org/10.1007/11426639_26
12. Boyen, X., Waters, B.: Compact group signatures without random oracles. In: Vaudenay, S. (ed.) EUROCRYPT 2006. LNCS, vol. 4004, pp. 427–444. Springer, Heidelberg (2006). https://doi.org/10.1007/11761679_26
13. Boyen, X., Waters, B.: Full-domain subgroup hiding and constant-size group signatures. In: Okamoto, T., Wang, X. (eds.) PKC 2007. LNCS, vol. 4450, pp. 1–15. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-71677-8_1
14. Camenisch, J., Lysyanskaya, A.: Signature schemes and anonymous credentials from bilinear maps. In: Franklin, M. (ed.) CRYPTO 2004. LNCS, vol. 3152, pp. 56–72. Springer, Heidelberg (2004). https://doi.org/10.1007/978-3-540-28628-8_4
15. Chaum, D., van Heyst, E.: Group signatures. In: Davies, D.W. (ed.) EUROCRYPT 1991. LNCS, vol. 547, pp. 257–265. Springer, Heidelberg (1991). https://doi.org/10.1007/3-540-46416-6_22
16. Cheng, X., Zhou, S., Guo, L., Yu, J., Ma, H.: An ID-based short group signature scheme. J. Softw. **8**(3), 554–559 (2013)
17. Emura, K., Miyaji, A., Omote, K.: A dynamic attribute-based group signature scheme and its application in an anonymous survey for the collection of attribute statistics. Inf. Media Technol. **4**(4), 1060–1075 (2009)
18. Escala, A., Herranz, J., Morillo, P.: Revocable attribute-based signatures with adaptive security in the standard model. In: Nitaj, A., Pointcheval, D. (eds.) AFRICACRYPT 2011. LNCS, vol. 6737, pp. 224–241. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-21969-6_14
19. Frikken, K., Atallah, M., Li, J.: Attribute-based access control with hidden policies and hidden credentials. IEEE Trans. Comput. **55**(10), 1259–1270 (2006)

20. Goyal, V., Pandey, O., Sahai, A., Waters, B.: Attribute-based encryption for fine-grained access control of encrypted data. In: Proceedings of the 13th ACM Conference on Computer and Communications Security, pp. 89–98. ACM (2006)
21. Groth, J.: Simulation-sound NIZK proofs for a practical language and constant size group signatures. In: Lai, X., Chen, K. (eds.) ASIACRYPT 2006. LNCS, vol. 4284, pp. 444–459. Springer, Heidelberg (2006). https://doi.org/10.1007/11935230_29
22. Groth, J., Ostrovsky, R., Sahai, A.: Perfect non-interactive zero knowledge for NP. In: Vaudenay, S. (ed.) EUROCRYPT 2006. LNCS, vol. 4004, pp. 339–358. Springer, Heidelberg (2006). https://doi.org/10.1007/11761679_21
23. Herranz, J., Laguillaumie, F., Libert, B., Ràfols, C.: Short attribute-based signatures for threshold predicates. In: Dunkelman, O. (ed.) CT-RSA 2012. LNCS, vol. 7178, pp. 51–67. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-27954-6_4
24. Herreweghen, E.: Secure anonymous signature-based transactions. In: Cuppens, F., Deswarte, Y., Gollmann, D., Waidner, M. (eds.) ESORICS 2000. LNCS, vol. 1895, pp. 55–71. Springer, Heidelberg (2000). https://doi.org/10.1007/10722599_4
25. Holt, J.E., Bradshaw, R.W., Seamons, K.E., Orman, H.: Hidden credentials. In: Proceedings of the 2003 ACM Workshop on Privacy in the Electronic Society, pp. 1–8. ACM (2003)
26. Kazue, S.: Generating statistical information in anonymous surveys. IEICE Trans. Fund. Electron. Commun. Comput. Sci. **79**(4), 507–512 (1996)
27. Khader, D.: Attribute based group signature with revocation. IACR Cryptology ePrint Archive, 2007:241 (2007)
28. Khader, D.: Attribute based group signatures. IACR Cryptology ePrint Archive, 2007:159 (2007)
29. Kiayias, A., Yung, M.: Secure scalable group signature with dynamic joins and separable authorities. Int. J. Secur. Netw. **1**(1–2), 24–45 (2006)
30. Lewko, A., Okamoto, T., Sahai, A., Takashima, K., Waters, B.: Fully secure functional encryption: attribute-based encryption and (Hierarchical) inner product encryption. In: Gilbert, H. (ed.) EUROCRYPT 2010. LNCS, vol. 6110, pp. 62–91. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-13190-5_4
31. Li, J., Kim, K.: Attribute-based ring signatures. IACR Cryptology EPrint Archive, 2008:394 (2008)
32. Liang, X., Cao, Z., Shao, J., Lin, H.: Short group signature without random oracles. In: Qing, S., Imai, H., Wang, G. (eds.) ICICS 2007. LNCS, vol. 4861, pp. 69–82. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-77048-0_6
33. Maji, H.K., Prabhakaran, M., Rosulek, M.: Attribute-based signatures: Achieving attribute-privacy and collusion-resistance. IACR Cryptology ePrint Archive, 2008:328 (2008)
34. Maji, H.K., Prabhakaran, M., Rosulek, M.: Attribute-based signatures. In: Kiayias, A. (ed.) CT-RSA 2011. LNCS, vol. 6558, pp. 376–392. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-19074-2_24
35. Nakanishi, T., Sugiyama, Y.: An efficient anonymous survey for attribute statistics using a group signature scheme with attribute tracing. IEICE Trans. Fund. Electron. Commun. Comput. Sci. **86**(10), 2560–2568 (2003)
36. Okamoto, T., Takashima, K.: Efficient attribute-based signatures for non-monotone predicates in the standard model. In: Catalano, D., Fazio, N., Gennaro, R., Nicolosi, A. (eds.) PKC 2011. LNCS, vol. 6571, pp. 35–52. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-19379-8_3

37. Patel, B.K., Jinwala, D.: Anonymity in attribute-based group signatures. In: Thilagam, P.S., Pais, A.R., Chandrasekaran, K., Balakrishnan, N. (eds.) ADCONS 2011. LNCS, vol. 7135, pp. 495–504. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-29280-4_58
38. Shahandashti, S.F., Safavi-Naini, R.: Threshold attribute-based signatures and their application to anonymous credential systems. In: Preneel, B. (ed.) AFRICACRYPT 2009. LNCS, vol. 5580, pp. 198–216. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-02384-2_13
39. Yang, G., Wong, D.S., Deng, X., Wang, H.: Anonymous signature schemes. In: Yung, M., Dodis, Y., Kiayias, A., Malkin, T. (eds.) PKC 2006. LNCS, vol. 3958, pp. 347–363. Springer, Heidelberg (2006). https://doi.org/10.1007/11745853_23
40. Zhang, J., Geng, Q.: A novel ID-based group signature scheme. In: Wireless Communications, Networking and Mobile Computing, pp. 1–4. IEEE (2008)