Attribute-Based Signcryption : Signer Privacy, Strong Unforgeability and IND-CCA2 Security in Adaptive-Predicates Attack

Tapas Pandit[1], Sumit Kumar Pandey[2], and Rana Barua[1]

[1] Indian Statistical Institute, Kolkata, India
[2] C R RAO AIMSCS, Hyderbad, India

Abstract. An Attribute-Based Signcryption (ABSC) is a natural extension of Attribute-Based Encryption (ABE) and Attribute-Based Signature (ABS), where we have the message confidentiality and authenticity together. Since the signer privacy is captured in security of ABS, it is quite natural to expect that the signer privacy will also be preserved in ABSC. In this paper, first we propose an ABSC scheme which is *weak existential unforgeable, IND-CCA2* secure in *adaptive-predicates* attack and achieves *signer privacy*. Secondly, by applying strongly unforgeable one-time signature (OTS), the above scheme is lifted to an ABSC scheme to attain *strong existential unforgeability* in *adaptive-predicates* model. Both the ABSC schemes are constructed on common setup, i.e the public parameters and key are same for both the encryption and signature modules. Our first construction is in the flavor of $\mathcal{CE}\&\mathcal{S}$ paradigm, except one extra component that will be computed using both signature components and ciphertext components. The second proposed construction follows a new paradigm (extension of $\mathcal{CE}\&\mathcal{S}$), we call it "Commit then Encrypt and Sign then Sign" ($\mathcal{CE}\&\mathcal{SS}$). The last signature is done using a strong OTS scheme. Since the non-repudiation is achieved by $\mathcal{CE}\&\mathcal{S}$ paradigm, our systems also achieve the same.

Keywords: Attribute-based encryption, Attribute-based signature, Attribute-based signcryption, Commitment scheme.

1 Introduction

In the last couple of years, attribute-based encryption (ABE) has become a privilege way for encrypting a message for many users. In this encryption, a message is encoded with a policy and a key is labeled with a set of attributes. This form of ABE is known as ciphertext-policy attribute-based encryption (CP-ABE) and in its dual form, key-policy attribute-based encryption (KP-ABE), the role of policy and the set of attributes are interchanged. Since its introduction (Fuzzy Identity-Based Encryption) [35] till to date many schemes have been proposed, some of them are CP-ABE [4,22,30,39,21], some of them are KP-ABE [16,33,22,30,2], most of them are selectively secure under chosen plaintext attack (CPA) [16,39,33,2], few of them are adaptively secure under CPA [30,22,32] and

S.S.M. Chow et al. (Eds.): ProvSec 2014, LNCS 8782, pp. 274–290, 2014.

very few of them are secure under chosen ciphertext attack (CCA) [30] for general policies. But, there are techniques [9,7,40] to convert a CPA secure scheme to CCA secure scheme. However, the schemes that are adaptively secure under CCA in the standard model seem to be more powerful.

Side by side with ABE, attribute-based signature (ABS) also draws much attention due to its versatility. Unlike the traditional signature scheme, it captures unforgeability for a policy (group of users) and signer privacy. Similar to ABE, in attribute-based signature a message is signed under a policy and a key is associated with a set of attributes. We call this form of ABS as CP-ABS [31,26,23,27] and its dual form, where the role of the policy and the set of attributes are reversed, is called KP-ABS [36]. Similar to the traditional signature, ABS can be weak existential unforgeable[1] [31,26,27,23] or strong existential unforgeable under chosen message attack (CMA). Most of the ABS [36] proposed so far are weak existential unforgeable. But, by a simple technique [20] one can obtain strongly unforgeable signature scheme from weak unforgeable scheme. Since here the message is signed under a policy, similar to ABE there are two types of unforgeability, selective-predicate [36,23] and adaptive-predicate [31,26,27].

Zheng [41] introduced the concept of signcryption that provides an efficient way of achieving the message confidentiality and an authenticity together as compared to "Sign then Encrypt" approach. But they have not given any formal security proof as no formal security model was known to them. Then J.Baek et al. [3] first formalized the security notion for signcryption. Later An et al. [1] proposed the generic constructions of signcryption in three paradigm, "Sign then Encrypt ($\mathcal{S}t\mathcal{E}$)", "Encrypt then Sign ($\mathcal{E}t\mathcal{S}$)" and "Commit then Encrypt and Sign ($\mathcal{C}t\mathcal{E}\&\mathcal{S}$)". As compared to $\mathcal{S}t\mathcal{E}$ and $\mathcal{E}t\mathcal{S}$ paradigms, $\mathcal{C}t\mathcal{E}\&\mathcal{S}$ has an advantage that in Signcrypt (resp. Unsigncrypt) both the routines, Encrypt and Sign (resp. Decrypt and Ver) can be executed in parallel, i.e., in $\mathcal{C}t\mathcal{E}\&\mathcal{S}$ paradigm both Signcrypt and Unsigncrypt run faster as compared to other two paradigms. The generic constructions in [1] were proven in two users model in PKI setting, but using some minor modification one can have the same security in multi user setting. Since it's debut several signcryption schemes [29,28,24,25,13,11,8] have been proposed either in PKI setting or in IBE setting.

Meanwhile S.Haber et al. [17] first proposed the idea of combining public-key schemes, where an encryption scheme and a signature scheme are combined to have the common public parameters and the key. But the Encrypt and Decrypt (resp. Sign and Ver) of the encryption (resp. signature) scheme were kept unchanged in the combined scheme. The security model is called joint security of the combined public-key schemes, where in message confidentiality the adversary is given only the encryption component of the challenge message but not the signature and in authenticity the adversary is has to forge a signature. In both cases, the adversary will get access to some oracles. Later, Vasco et al. showed in [37] that the IBE scheme [6] and the IBS scheme [19] can be combined in the joint security model. However, in this joint security model semantic

[1] Unless stated, existential unforgeable means weak existential unforgeable throughout this paper.

security of the message is not possible if the signature of the challenge message is additionally given with the challenge ciphertext.

It is natural to ask whether signcryption can be extended to the context of attribute-based cryptography. It was Gagné et. al. [15] who first answered the question but the policy considered in their construction (called attribute-based signcryption) was a threshold policy. Basically in their construction, the structure of Fuzzy IBE in [35] and a new efficient threshold ABS were used as encryption primitive and signature primitive respectively. Subsequently, Emura et al. [14] proposed a dynamic attribute-based signcryption (ABSC), where access structures of encryptors can be changed without re-issuing the secret keys of the users. Both the signcryption scheme were shown to be secure (confidentiality and authenticity) under selective-predicate attack. Since ABSC is a natural

Table 1. Performance of our CP-ABSC scheme

Scheme	CS	Key size	Signcryption size	Signcrypt time	Unsigncrypt time
[15]	No	$2\|A_s\|, 3\|A_e\|$	$\mathcal{O}(\|\omega_s\| + \|\omega_e\|)$	$\mathcal{O}(\|\omega_s\| + \|\omega_e\|)$	$\mathcal{O}(\|\omega_s\| + d)$
[14]	No	$2\|A_s\|, \theta_e$	$\mathcal{O}(\ell_s + \|\mathcal{U}_e\| + \Im)$	$\mathcal{O}(\ell_s + \|\mathcal{U}_e\| + \Im)$	$\mathcal{O}(\ell_s + \|\mathcal{U}_e\| + \Im)$
[10]	Yes	$\mathfrak{M}\|A\| + 2$	$2\ell_s + \ell_e + 4$	$\mathcal{O}(\ell_s) + \mathcal{O}(\ell_e)$	$\mathcal{O}(\ell_s) + \mathcal{O}(\|\mathcal{I}_B\|)$
Our	Yes	$\mathfrak{M}\|A\| + 2$	$2\ell_s + 2\ell_e + 5 + \wp$	$Max\{\mathcal{O}(\ell_s), \mathcal{O}(\ell_e)\}$	$Max\{\mathcal{O}(\ell_s), \mathcal{O}(\|\mathcal{I}_B\|)\}$

In table 1, CS and $|A|$ stand for the common setup and cardinality of the set A respectively. The schemes supporting the common setup have the single key extraction algorithm and in this case, we use A to indicate the user set of attributes. Otherwise two set of attributes, A_s and A_e are used respectively for signcryption and unsigncryption. In later case, the individual key sizes are separated by comma (,). Let ℓ_s and ℓ_e respectively denote the size of the signer policy Γ_s and receiver policy Γ_e. $\mathfrak{M}$ stands for maximum # repetition of an attribute in an access policy. Let ω_s, ω_e and d respectively represent the signing set of attributes, encryption set of attributes and threshold value in [15]. $\mathcal{U}_e$ and $\Im$ respectively denote the attribute universe involved in encryption and length of verification key for OTS. $\theta_e = 2|A_e| + 2\Im + 1$. The sizes of the commitment and the one-time signature are described by $\wp$. Let $|\mathcal{I}_B|$ be the minimum # row in the receiver policy Γ_e labeled by the set B to compute the target vector $\mathbf{1}$. The key size and signcryption size are measured by # group elements involved in the key and signcryption respectively. The time for signcrypt is # exponentiations to construct a signcryption, whereas the time for unsigncrypt is both # exponentiations and # pairings.

Table 2. Security features of our CP-ABSC scheme

Scheme	Type	SAS	EAS	Auth.	Conf.	NR	SP	APM
[15]	KP	Threshold	Threshold	wEUF-CMA	IND-CCA2	No	NK	No
[14]	CP	MAT	AGW	sEUF-CMA	IND-CCA2	Yes	No	No
[10]	CP	MSP	MSP	sEUF-CMA	IND-CCA2	Yes	Yes	No
Our	CP	MSP	MSP	sEUF-CMA	IND-CCA2	Yes	Yes	Yes

In table 2, the abbreviations SAS, EAS, Auth., Conf., NR, SP, APM, NK, MAT, MSP, AGW, KP and CP stand for signing access structure, encryption access structure, signcryption unforgeability, confidentiality of message, non-repudiation, signer-privacy, adaptive-predicates model, not known, monotone access tree, monotone span program, AND-gate with wildcard respectively, key-policy and ciphertext-policy.

extension of both ABE and ABS, and the signer privacy is preserved in ABS, so the signer privacy property is supposed to be inherited in ABSC as well. But the later ABSC scheme lacks the property of signer privacy.

Recently Chen et al. [10] proposed a scheme in combined public-key framework but in attribute-based flavor. In their scheme the ABE and ABS modules have the same public parameters and the key distribution. Their scheme is based on the construction of Waters [39] and was shown to be secure (selectively) in the joint security model. Then they extended it to have a combined attribute-based signcryption ($\mathcal{StE}$ paradigm).

1.1 Our Approach and Contribution

Our constructions are almost in the flavor of $\mathcal{CtE}\&\mathcal{S}$ paradigm. In $\mathcal{CtE}\&\mathcal{S}$ paradigm, a message m is first committed to $(\breve{c}, \breve{d})$, then the commitment part $\breve{c}$ and decommitment part $\breve{d}$ are respectively signed to σ and encrypted to ϱ in parallel to produce the signcryption $\Upsilon := (\breve{c}, \sigma, \varrho)$. Similarly, in unsigncryption the verification (to verify σ) and the decryption (to get the $\breve{d}$) run in parallel to extract the message as $m := \mathsf{Open}(\breve{c}, \breve{d})$. But this generalized construction [1] never achieves strong unforgeability (resp. CCA2 security) in the insider security model as long as the primitive encryption algorithm (resp. the primitive sign algorithm) is probabilistic.

Our first CP-ABSC construction achieves *signer privacy*, *adaptive-predicates weak unforgeability*, and *adaptive-predicates IND-CCA2* security in the standard model. Moreover, our constructions support the combined public-key environment of "Combined Public-Key scheme", viz, both the primitives, encryption and signature have a common setup, i.e., the public parameters and key are identical. Suppose we want a signcryption for a message m under the policies[2] (Γ_s, Γ_e). Let $\sigma := (\boldsymbol{S}_0, \ldots, \boldsymbol{S}_{\ell_s})$ be the signature for $(\breve{c}, \Gamma_s)$, generated by a primitive CP-ABS, where $(\breve{c}, \breve{d}) \longleftarrow \mathsf{Commit}(m)$. Let $\varrho_0 := (\boldsymbol{C}_0, \ldots, \boldsymbol{C}_{\ell_e})$ be the ciphertext generated by a primitive CP-ABE that conceals $\breve{d}$ under a policy Γ_e. To achieve the CCA2 security, we first bind all the components $\boldsymbol{S}_i$'s and $\boldsymbol{C}_i$'s through a collision resistant hash function $H_e : \{0,1\}^* \longrightarrow \mathbb{Z}_N$ to $h_e := H_e(\Gamma_e, \Gamma_s, \breve{c}, \varrho_0, \sigma)$. Then we encode h_e using a secret s_e involved in the encryption of the primitive CP-ABE and Boneh-Boyen hash technique [5] to an additional ciphertext component C_{ℓ_e+1}. This basically prevents the adversary $\mathscr{A}$ from changing the challenge signcryption except the component C_{ℓ_e+1}, but if it gets changed then it will be recognized via a verification process. If the primitive CP-ABS scheme is weak unforgeable and the commitment scheme has relaxed-binding property, then proposed CP-ABSC scheme is shown to be weak unforgeable.

[2] Γ_s and Γ_e are respectively signer policy (i.e., on whom behalf, signer signs m) and receiver policy (i.e., who will be eligible for this plaintext m).

Our second CP-ABSC scheme additionally achieves *strong unforgeability* in *adaptive-predicates* attack[3]. First notice that in the former scheme the adversary can modify the replied signcryption for a message (m, Γ_s, Γ_e) : since $\mathscr{A}$ has access to key $\mathcal{SK}_A$ with $\Gamma_e(A) = \mathsf{True}$, so it can extract $\breve{d}$ from ϱ and then re-encrypts it to get modified (new) signcryption for the same message (m, Γ_s, Γ_e). Therefore, the former scheme does not achieve the strong unforgeability. The later scheme is obtained by combining the former scheme and a strong one-time signature (OTS) scheme. Essentially, we sign $h_e||C_{\ell_e+1}$ using strong OTS scheme to guarantee that the signcryption for a message can not be altered even if the adversary knows the unsigncryption key. Surprisingly, the strong unforgeability of this CP-ABSC scheme relies only on the weak unforgeability of the primitive CP-ABS scheme and the strong unforgeability of the primitive strong OTS scheme, i.e., no more relaxed-binding property of the primitive commitment scheme is required.

The primitive CP-ABE scheme considered here is a (CCA2) variant[4] of CP-ABE scheme of Lewko et al. in [22]. Our primitive CP-ABS scheme (in section 3) has the similar structure as of ABS scheme in the combined public-key framework [10] except - (a) the encoding from hash of message to group element, and (b) the bilinear pairing groups. The ABS scheme of [10] was proven in selective-predicate model, whereas ours is shown to be secure in adaptive-predicate model. Since the adaptive security (confidentiality and authenticity) is one of the main motivations of our work, we must require the adaptive-unforgeability of the primitive CP-ABS scheme. Therefore, the ABS of [10] can not be applied directly to our CP-ABSC schemes. Another reason for moving prime to composite order pairing groups is to fit the ABS scheme to CP-ABE scheme of [22]. There are many commitment schemes [12,18,34] suitable for our systems, but we use them as a black box in our constructions.

Summary of Our Contribution. To the best of our knowledge, this is the first scheme having strong unforgeability and IND-CCA2 security in adaptive-predicates model. Since our solution supports $\mathcal{CE}\&\mathcal{S}$ paradigm, Signcrypt and Unsigncrypt run faster as compared to other paradigms, viz, $\mathcal{ES}$ and $\mathcal{SE}$. Our system is based on the common setup, i.e the public parameters and key are same for both the encryption and signature module. In addition it supports non-repudiation, dynamic property and signer privacy. A details comparisons of performance and the security features between our scheme and others are given in table 1 and table 2. The proofs of confidentiality and unforgeability are based on the dual system methodology of [38]. Due to space restriction, all the missing proofs will be given in full version of this paper.

Discussion. We remark that our proposed solution is not generic. One may think that applying the generic construction [1] it is possible but this is not the

[3] We remark that adaptive-predicates IND-CCA2 security (resp. existential unforgeability) and IND-CCA2 security (resp. existential unforgeability) in adaptive-predicates attack both carry the same meaning.

[4] This is not explicitly given but the signcryption scheme implicitly contains it.

case. Indeed, $\mathcal{CtE}\&\mathcal{S}$ paradigm preserves only weak unforgeability and[5] IND-gCCA2. But here our proposed scheme attains both strong unforgeability and IND-CCA2 security in adaptive-predicates attack. Further, our solution supports the common setup for encryption and signature, so the security proof can not carry through as in $\mathcal{CtE}\&\mathcal{S}$ paradigm. For our system, the considered form of ABS, where a signature is associated with a policy and key is labeled by a set of attributes, is called CP-ABS.

1.2 Organization

This paper is organized as follows. Section 2 contains the preliminaries. A CP-ABS scheme and its security are provided respectively in section 3 and 4. A adaptive-predicates weak unforgeable and IND-CCA2 secure CP-ABSC scheme and its security are given respectively in section 5 and 6. In section 7, our adaptive-predicates strongly unforgeable and IND-CCA2 secure CP-ABSC scheme and its security are demonstrated.

2 Preliminaries

Basic notation, definitions and hardness assumptions are provided in this section. For definition and security model of commitment scheme, ABS, strongly unforgeable OTS and CP-ABSC, refer to [1], [26], [30] and appendix A respectively. For access structure and linear secret sharing scheme, see [22].

Notation. Let $[\ell] := \{i \in \mathbb{N} : 1 \leq i \leq \ell\}$, $g_T := e(g, g)$, where e is a bilinear pairing. Let the vectors $\mathbf{1}$ and $\mathbf{0}$ respectively denote $(1, 0, \ldots, 0)$ and $(0, 0, \ldots, 0)$, where the length of the vectors will be understood from the context. Let $\boldsymbol{Y} := (y_1, \ldots, y_n)$ and $\boldsymbol{W} := (w_1, \ldots, w_n)$ be two vectors, then $\boldsymbol{Y}.\boldsymbol{W}$ denotes the dot product of $\boldsymbol{Y}$ and $\boldsymbol{W}$, i.e., $\boldsymbol{Y}.\boldsymbol{W} := \sum_{i=1}^{n} y_i w_i$. For $S \subset \mathbb{Z}_N^{\ell_s}$ and $\boldsymbol{\alpha} \in \mathbb{Z}_N^{\ell_s}$, we define $\boldsymbol{\alpha} + S := \{\boldsymbol{\alpha} + \boldsymbol{\beta} \mid \boldsymbol{\beta} \in S\}$. For a set X, $x \xleftarrow{\mathrm{R}} X$ denotes that x is randomly picked from X according to the distribution R. Likewise, $x \xleftarrow{\mathrm{U}} X$ indicates x is uniformly selected from X. To better understand the schemes, we use two subscripts, s and e respectively for encryption and signature. Through out this paper, we will use the symbol $\Gamma := (M, \rho)$ for monotone span programs, where $\ell \times n$ stands for the order of the matrix M. For an access structure Γ and a set attributes A, $\Gamma(A)$ stands for boolean variable, i.e, $\Gamma(A) = \mathsf{True}$ if A satisfies Γ, else $\Gamma(A) = \mathsf{False}$. For a matrix M_e (resp. M_s), the symbol $\boldsymbol{M}_e^{(i)}$ (resp. $\boldsymbol{M}_s^{(i)}$) represents the i^{th} row of the matrix M_e (resp. M_s).

Composite Order Bilinear Groups. Let $\mathcal{G}$ be an algorithm which takes 1^κ as a security parameter and returns a description of a composite order bilinear groups, $\mathcal{J} := (N := p_1 p_2 p_3, \mathbb{G}, \mathbb{G}_T, e)$, where p_1, p_2, p_3 are three distinct primes and $\mathbb{G}$ and $\mathbb{G}_T$ are cyclic groups of order N and $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ is a map such that

[5] IND-gCCA2 is a weaker security notion than IND-CCA2. For details refer to [1].

1. (Bilinear) $\forall g, h \in \mathbb{G}, a, b \in \mathbb{Z}_N, e(g^a, h^b) = e(g, h)^{ab}$
2. (Non-degenerate) $\exists g \in \mathbb{G}$ such that $e(g, g)$ has order N in $\mathbb{G}_T$

Let $\mathbb{G}_{p_1}, \mathbb{G}_{p_2}$ and $\mathbb{G}_{p_3}$ respectively denote the subgroups of $\mathbb{G}$ of order p_1, p_2 and p_3. Let $h_i \in \mathbb{G}_{p_i}$ and $h_j \in \mathbb{G}_{p_j}$ be arbitrary elements with $i \neq j$, then $e(h_i, h_j) = 1$. This property is called orthogonal property of $\mathbb{G}_{p_1}, \mathbb{G}_{p_2}, \mathbb{G}_{p_3}$.

Hardness Assumptions. We describe here three Decisional SubGroup (DSG) assumptions [22] for 3 primes, DSG1, DSG2 and DSS3 in composite order bilinear groups. Let $\mathcal{J} := (N = p_1p_2p_3, \mathbb{G}, \mathbb{G}_T, e) \xleftarrow{U} \mathcal{G}(1^\lambda)$ be the common parameters for each assumptions.
[DSG1]. Let $g \xleftarrow{U} \mathbb{G}_{p_1}, X_3 \xleftarrow{U} \mathbb{G}_{p_3}, T_0 \xleftarrow{U} \mathbb{G}_{p_1}$, $T_1 \xleftarrow{U} \mathbb{G}_{p_1p_2}$. Define $\mathcal{D} := (\mathcal{J}, g, X_3)$
[DSG2]. Let $g, X_1 \xleftarrow{U} \mathbb{G}_{p_1}$, $X_2, Y_2 \xleftarrow{U} \mathbb{G}_{p_2}$, $X_3, Y_3 \xleftarrow{U} \mathbb{G}_{p_3}, T_0 \xleftarrow{U} \mathbb{G}_{p_1p_3}$, $T_1 \xleftarrow{U} \mathbb{G}$. Then set $\mathcal{D} := (\mathcal{J}, g, X_1X_2, Y_2Y_3, X_3)$
[DSG3]. Let $\alpha, s \xleftarrow{U} \mathbb{Z}_N$, $g \xleftarrow{U} \mathbb{G}_{p_1}$, $X_2, Y_2, Z_2 \xleftarrow{U} \mathbb{G}_{p_2}$, $X_3 \xleftarrow{U} \mathbb{G}_{p_3}, T_0 := e(g, g)^{\alpha s}$, $T_1 \xleftarrow{U} \mathbb{G}_T$. Define $\mathcal{D} := (\mathcal{J}, g, g^\alpha X_2, g^s Y_2, Z_2, X_3)$

The advantage of an algorithm $\mathscr{A}$ in breaking DSGi, for $i = 1, 2, 3$ is defined by

$$\mathsf{Adv}^{\mathrm{DSGi}}_{\mathscr{A}}(\kappa) = |Pr[\mathscr{A}(\mathcal{D}, T_0) = 1] - Pr[\mathscr{A}(\mathcal{D}, T_1) = 1]|$$

We say that the DSGi assumption holds if for every PPT algorithm $\mathscr{A}$, the advantage $\mathsf{Adv}^{\mathrm{DSGi}}_{\mathscr{A}}(\kappa)$ is at most negligible in security parameter κ.

3 Basic Ciphertext-Policy Attribute-Based Signature

Illustrated here is a basic ciphertext-policy attribute-based signature (CP-ABS) scheme for monotone span program (MSP) in the composite order pairing groups $(N := p_1p_2p_3, \mathbb{G} := \mathbb{G}_{p_1} \times \mathbb{G}_{p_2} \times \mathbb{G}_{p_2}, \mathbb{G}_T, e)$, for 3 distinct primes p_1, p_2 and p_3. The subgroup $\mathbb{G}_{p_2}$ has no role in this scheme but it will be used to prove the security. As we mentioned earlier that the proposed CP-ABS scheme has the similar structure to that of [10] except some minor modifications, viz., the encoding function from hash of messages to group elements and pairing groups. To have the unforgeability of the ABS scheme in adaptive-predicate model, we allow such modifications. In this basic CP-ABS construction, the policies, i.e., MSPs are restricted to have each entry of row labeling function ρ_s to be distinct. In other word, the row labeling functions ρ_s of the monotone span programs $\Gamma_s := (M_s, \rho_s)$ are injective. From this basic CP-ABS construction one can easily lift to full CP-ABS construction by a mechanism described in appendix B.
$\mathsf{Setup}(1^\kappa, \mathcal{U})$: It executes $\mathcal{G}(1^\kappa)$ to have composite order bilinear groups descriptor, $\mathcal{J} := (N := p_1p_2p_3, \mathbb{G}, \mathbb{G}_T, e)$ with known factorization p_1, p_2 and p_3 of N. It chooses $g \xleftarrow{U} \mathbb{G}_{p_1}, X_3 \xleftarrow{U} \mathbb{G}_{p_3}$, $a, a_s, b_s, \alpha \xleftarrow{U} \mathbb{Z}_N$ and $t_i \xleftarrow{U} \mathbb{Z}_N$ for

each attribute $i \in \mathcal{U}$. It then sets $u_s := g^{a_s}, v_s := g^{b_s}, T_i := g^{t_i}$ for $i \in \mathcal{U}$. Let $H_s : \{0,1\}^* \longrightarrow \mathbb{Z}_N$ be a hash function. The public parameters and master secret are given by

$\mathcal{PP} := (\mathcal{J}, g, g^a, u_s, v_s, g_T^\alpha, \{T_i\}_{i\in\mathcal{U}}, X_3, H_s)$

$\mathcal{MSK} := (\alpha)$.

$\mathsf{KeyGen}(\mathcal{PP}, \mathcal{MSK}, A)$: It picks $t \xleftarrow{U} \mathbb{Z}_N, R, R'_0 \xleftarrow{U} \mathbb{G}_{p_3}$. For each attribute $i \in A$, the algorithm chooses $R_i \xleftarrow{U} \mathbb{G}_{p_3}$ and outputs the secret key

$\mathcal{SK}_A := [A,\ K := g^{\alpha+at}R,\ L := g^t R'_0,\ K_i := {T_i}^t R_i,\ \forall i \in A]$.

$\mathsf{Sign}(\mathcal{PP}, m, \mathcal{SK}_A, \Gamma_s := (M_s, \rho_s))$: Let M_s be an $\ell_s \times n_s$ matrix. Suppose $\Gamma_s(A) = \mathsf{True}$, then there exist $\mathcal{I}_A \subset [\ell_s]$ and $\{\alpha_s^{(i)}\}_{i\in\mathcal{I}_A}$ such that $\sum_{i\in\mathcal{I}_A} \alpha_s^{(i)} \boldsymbol{M}_s^{(i)} = \mathbf{1}$. It selects $\boldsymbol{\beta} \xleftarrow{U} \{\boldsymbol{\beta} = (\boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_{\ell_s}) \in \mathbb{Z}_N^{\ell_s} \mid \sum_{i\in[\ell_s]} \beta_i \boldsymbol{M}_s^{(i)} = \mathbf{0}\}$. Suppose $\mathcal{SK}_A := [A,\ K := g^{\alpha+at}R,\ L := g^t R'_0,\ K_i := {T_i}^t R_i,\ \forall i \in A]$, then it re-randomizes the key $\mathcal{SK}_A$ as follows: it picks $\hat{t} \xleftarrow{U} \mathbb{Z}_N$ and sets $\tilde{t} := t + \hat{t}$

$\tilde{\mathcal{SK}}_A := [A,\ \tilde{K} := K.g^{a\hat{t}},\ \tilde{L} := L.g^{\hat{t}},\ \tilde{K}_i := K_i.{T_i}^{\hat{t}},\ \forall i \in A]$

$:= [A,\ \tilde{K} := g^{\alpha+a\tilde{t}}R,\ \tilde{L} := g^{\tilde{t}}R'_0,\ \tilde{K}_i := {T_i}^{\tilde{t}}R_i,\ \forall i \in A]$

It picks $r_s, \tau \xleftarrow{U} \mathbb{Z}_N, \bar{R}, \bar{R}_0 \xleftarrow{U} \mathbb{G}_{p_3}$ and for each $i \in [\ell_s]$, it chooses $\bar{R}_i \xleftarrow{U} \mathbb{G}_{p_3}$. Then it computes $h_s := H_s(m||\Gamma_s)$. The components of signature are given by (for $i \notin \mathcal{I}_A$, it sets $\alpha_s^{(i)} := 0$)

$\boldsymbol{S}_0 := \left(\tilde{K}(u_s^{h_s} v_s)^{r_s} \bar{R},\ g^{r_s} \bar{R}_0 \right)$

$\boldsymbol{S}_i := \left(\tilde{L}^{\alpha_s^{(i)}} (g^\tau)^{\beta_i} \bar{R}_i,\ (\tilde{K}_{\rho_s(i)})^{\alpha_s^{(i)}} (T_{\rho_s(i)})^{\tau\beta_i} \bar{R'}_i \right)$ for $i \in [\ell_s]$.

After simplification, it gives

$\boldsymbol{S}_0 := \left(g^{\alpha+a\tilde{t}} (u_s^{h_s} v_s)^{r_s} \tilde{R},\ g^{r_s} \tilde{R}_0 \right)$, where $\tilde{R} := R\bar{R}$, $\tilde{R}_0 := \bar{R}_0$

$\boldsymbol{S}_i := \left((g^{\tilde{t}})^{\alpha_s^{(i)}} (g^\tau)^{\beta_i} \tilde{R}_i,\ \ (T_{\rho_s(i)}^{\tilde{t}})^{\alpha_s^{(i)}} (T_{\rho_s(i)}^{\tau})^{\beta_i} \tilde{R'}_i \right)$,

where $\tilde{R}_i := (R'_0)^{\alpha_s^{(i)}} \bar{R}_i$, $\tilde{R'}_i := R_{\rho_s(i)}^{\alpha_s^{(i)}+\tau\beta_i} \bar{R}'_i$

The final output (signature) is $\sigma := (\boldsymbol{S}_0, \{\boldsymbol{S}_i\}_{i\in[\ell_s]})$

$\mathsf{Ver}(\mathcal{PP}, m, \sigma, \Gamma_s)$: It first computes a verification text, then using this verification text it will verify the signature. The following is the construction of verification text: It picks $\boldsymbol{u}_s := (s, u_2, \ldots, u_{n_s}) \xleftarrow{U} \mathbb{Z}_N^{n_s}$ and $r_s^{(i)} \xleftarrow{U} \mathbb{Z}_N$ for $i \in [\ell_s]$. It computes $h_s := H_s(m||\Gamma_s)$. Let $M_s^{(i)}$ denote the i^{th} row of the matrix, M_s and let $\lambda_s^{(i)} := \boldsymbol{M}_s^{(i)}.\boldsymbol{u}_s$. The verification text is given by

$\boldsymbol{V}_0 := \left(g^s, (u_s^{h_s} v_s)^s, g_T^{\alpha s} \right)$

$\boldsymbol{V}_i := \left(g^{a\lambda_s^{(i)}} T_{\rho_s(i)}^{-r_s^{(i)}},\ \ g^{r_s^{(i)}} \right)$, for $i \in [\ell_s]$

The final verification text is $\mathcal{V} := (\boldsymbol{V}_0, \{\boldsymbol{V}_i\}_{i\in[\ell_s]})$

Now, it computes $\Delta_s := \frac{e(S_{01}, V_{01})}{e(S_{02}, V_{02}) \prod_{i=1}^{\ell_s} (e(S_{i1}, V_{i1}) e(S_{i2}, V_{i2}))}$ and checks $\Delta_s \stackrel{?}{=} V_{03}$. It returns 1 if $\Delta_s = V_{03}$, else returns 0.

Correctness.

$$\begin{aligned}
\Delta_s &= \frac{e(S_{01}, V_{01})}{e(S_{02}, V_{02}) \prod_{i=1}^{\ell_s} (e(S_{i1}, V_{i1}) e(S_{i2}, V_{i2}))} \\
&= \frac{g_T^{\alpha s + a\tilde{t}s} . e(u_s^{h_s} v_s, g)^{sr_s}}{e(u_s^{h_s} v_s, g)^{sr_s} \prod_{i=1}^{\ell_s} (e(S_{i1}, V_{i1}) e(S_{i2}, V_{i2}))} \\
&= \frac{g_T^{\alpha s + a\tilde{t}s}}{\prod_{i=1}^{\ell_s} (e(g^{\tilde{t}\alpha_s^{(i)} + \tau\beta_i}, g^{a\lambda_s^{(i)} - r_s^{(i)} t_{\rho_s(i)}}) . e(g^{\tilde{t}\alpha_i t_{\rho_s(i)} + \tau\beta_i t_{\rho_s(i)}}, g^{r_s^{(i)}}))} \\
&= \frac{g_T^{\alpha s + a\tilde{t}s}}{\prod_{i=1}^{\ell_s} g_T^{a\tilde{t}\lambda_s^{(i)}\alpha_s^{(i)} + a\tau\lambda_s^{(i)}\beta_i}} = \frac{g_T^{\alpha s + a\tilde{t}s}}{g_T^{a\tilde{t}\sum_{i=1}^{\ell_s}\lambda_s^{(i)}\alpha_s^{(i)} + a\tau\sum_{i=1}^{\ell_s}\lambda_s^{(i)}\beta_i}} = g_T^{\alpha s}
\end{aligned}$$

4 Security Proof of CP-ABS

Theorem 1. *The proposed attribute-based signature scheme in section 3 is perfectly private.*

Theorem 2. *The proposed basic CP-ABS scheme is adaptive-predicate existential unforgeable if DSG1, DSG2 and DSG3 assumptions hold and H_s is a collision resistant hash function.*

5 Basic Ciphertext-Policy Attribute-Based Signcryption

In this section, we present our basic ciphertext-policy attribute-based signcryption (CP-ABSC) supporting monotone span programs. The scheme is based on the composite order bilinear pairing groups. Here we consider two policies, sender policy $\Gamma_s := (M_s, \rho_s)$ and receiver policy $\Gamma_e := (M_e, \rho_e)$. Similar to section 3, in our basic CP-ABSC scheme, both the row labeling functions ρ_s and ρ_e are assumed to be injective. By applying the mechanism illustrated in appendix B, a full CP-ABSC construction is easily obtained.

This construction is almost in the flavor of $\mathcal{CE}\&\mathcal{S}$ paradigm. To construct our scheme, we use any commitment scheme with hiding and relaxed-binding properties, CCA2 version encryption scheme of [22] and the ABS scheme described in section 3. Let $\Pi_{\mathsf{ABS}} := (\mathsf{ABS.Setup}, \mathsf{ABS.KeyGen}, \mathsf{ABS.Sign}, \mathsf{ABS.Ver})$ and $\Pi_{\mathsf{Commit}} := (\mathsf{C.Setup}, \mathsf{Commit}, \mathsf{Open})$ be respectively the ABS scheme described in section 3 and commitment scheme.

$\mathsf{Setup}(1^\kappa, \mathcal{U})$: It runs $\mathcal{CK} \longleftarrow \mathsf{C.Setup}(1^\kappa)$, $(\mathcal{ABS}.\mathcal{PP}, \mathcal{ABS}.\mathcal{MSK}) \longleftarrow \mathsf{ABS.Setup}(1^\kappa, \mathcal{U})$. It chooses $a_e, b_e \xleftarrow{\mathrm{U}} \mathbb{Z}_N$ and sets $u_e := g^{a_e}, v_e := g^{b_e}$. Let $H_e : \{0,1\}^* \longrightarrow \mathbb{Z}_N$ be a hash functions. The public parameters (combining $\mathcal{ABS}.\mathcal{PP}, \mathcal{CK}$ and u_e, v_e, H_e) and master secret are given by

$\mathcal{PP} := (\mathcal{I}, g, g^a, u_s, u_e, v_s, v_e, g_T^\alpha, \{T_i\}_{i\in\mathcal{U}}, X_3, H_s, H_e, \mathcal{CK})$

$\mathcal{MSK} := \mathcal{ABS}.\mathcal{MSK} = (\alpha)$

$\mathsf{KeyGen}(\mathcal{PP}, \mathcal{MSK}, A)$: $\mathcal{SK}_A \longleftarrow \mathsf{ABS.KeyGen}(\mathcal{ABS}.\mathcal{PP}, \mathcal{MSK}, A)$
$\mathsf{Signcrypt}(\mathcal{PP}, m, \mathcal{SK}_A, \Gamma_s := (M_s, \rho_s), \Gamma_e := (M_e, \rho_e))$: Let M_s (resp. M_e) be an $\ell_s \times n_s$ (resp. $\ell_e \times n_e$) matrix. It runs $(\breve{c}, \breve{d}) \longleftarrow \mathsf{Commit}(m)$ (see footnote [6]). The Signcrypt algorithm has two part, Sign and Encrypt, both run in parallel except the part C_{ℓ_e+1}.

***Sign*:** $\sigma := (\boldsymbol{S}_0, \{\boldsymbol{S}_i\}_{i\in[\ell_s]}) \longleftarrow \mathsf{ABS.Sign}(\mathcal{ABS}.\mathcal{PP}, (\breve{c}||\Gamma_e), \mathcal{SK}_A, \Gamma_s := (M_s, \rho_s))$, where the components are given by

$$\boldsymbol{S}_0 := \left(g^{\alpha+a\tilde{t}}(u_s^{h_s}v_s)^{r_s}\tilde{R},\ g^{r_s}\tilde{R}_0 \right), \text{ where } h_s := H_s((\breve{c}||\Gamma_e)||\Gamma_s)$$
$$\boldsymbol{S}_i := \left((g^{\tilde{t}})^{\alpha_s^{(i)}}(g^{\tau})^{\beta_i}\tilde{R}_i,\ \ (T^{\tilde{t}}_{\rho_s(i)})^{\alpha_s^{(i)}}(T^{\tau}_{\rho_s(i)})^{\beta_i}\tilde{R'}_i \right)$$

***Encrypt*:** It picks $\boldsymbol{u}_e := (s_e, u_2, \ldots, u_{n_e}) \xleftarrow{\mathrm{U}} \mathbb{Z}_N^{n_e}$ and $r_e^{(i)} \xleftarrow{\mathrm{U}} \mathbb{Z}_N$ for $i \in [\ell_e]$. Let $M_e^{(i)}$ denote the i^{th} row of the matrix, M_e and let $\lambda_e^{(i)} := \boldsymbol{M}_e^{(i)}.\boldsymbol{u}_e$. The ciphertext components of the signcryption are given by

$$\boldsymbol{C}_0 := \left(g^{s_e},\ \breve{d}.g_T^{\alpha s_e} \right)$$
$$\boldsymbol{C}_i := \left(g^{a\lambda_e^{(i)}}T^{-r_e^{(i)}}_{\rho_e(i)},\quad g^{r_e^{(i)}} \right), \text{ for } i \in [\ell_e]$$

It sets $\varrho_0 := (\boldsymbol{C}_0, \{\boldsymbol{C}_i\}_{i\in[\ell_e]})$ and computes $h_e := H_e(\Gamma_e, \Gamma_s, \breve{c}, \varrho_0, \sigma)$. Then, it calculates the last component $C_{\ell_e+1} := (u_e^{h_e}v_e)^{s_e}$. Then it sets the ciphertext part of the signcryption as $\varrho := (\varrho_0, C_{\ell_e+1})$.

It outputs the signcryption $\Upsilon := (\breve{c}, \sigma, \varrho)$
$\mathsf{Unsigncrypt}(\mathcal{PP}, \Upsilon, \mathcal{SK}_B, \Gamma_s := (M_s, \rho_s), \Gamma_e := (M_e, \rho_e))$: Let M_s (resp. M_e) be an $\ell_s \times n_s$ (resp. $\ell_e \times n_e$) matrix. This algorithm consists of two routines, Ver and Decrypt run in parallel.
Ver: $\mathsf{flag} \longleftarrow \mathsf{ABS.Ver}(\mathcal{PP}, (\breve{c}||\Gamma_e), \sigma, \Gamma_s)$. If $\mathsf{flag} = 0$, it returns $\perp$
Decrypt: It computes $h_e := H_e(\Gamma_e, \Gamma_s, \breve{c}, \varrho_0, \sigma)$. Then it checks $e(g, C_{\ell_e+1}) \stackrel{?}{=} e(u_e^{h_e}v_e, C_{01})$ and if the equality does not hold, it returns $\perp$. If $\Gamma_e(B) \neq \mathsf{True}$, it returns $\perp$, else there exist $\mathcal{I}_B \subset [\ell_e]$ and $\{\alpha_e^{(i)}\}_{i\in\mathcal{I}_B}$ such that $\sum_{i\in\mathcal{I}_B}\alpha_e^{(i)}\boldsymbol{M}_e^{(i)} = \mathbf{1}$. Then, it picks $r \xleftarrow{\mathrm{U}} \mathbb{Z}_N,\ R_0 \xleftarrow{\mathrm{U}} \mathbb{G}_{p_3}$ and computes

$$\Delta_e := \frac{e(K.(u_e^{h_e}v_e)^r, C_{01})}{e(g^rR_0, C_{\ell_e+1})\prod_{i\in\mathcal{I}_B}(e(L, C_{i1}).e(K_{\rho_e(i)}, C_{i2}))^{\alpha_e^{(i)}}}$$

Finally it returns the message $m := \mathsf{Open}(\breve{c}, C_{02}/\Delta_e)$

Correctness. It follows from the correctness of Ver and Decrypt routines. Since, the correctness of Ver is immediate from that of ABS in section 3, we illustrate here only the correctness of Decrypt.

[6] For brevity, we just omit $\mathcal{CK}$ in Open and Commit algorithm throughout this paper.

$$\Delta_e = \frac{e(K.(u_e^{h_e}v_e)^r, C_{01})}{e(g^r, C_{\ell_e+1})\prod_{i\in\mathcal{I}_B}(e(L,C_{i1}).e(K_{\rho_e(i)},C_{i2}))^{\alpha_e^{(i)}}}$$
$$= \frac{g_T^{\alpha s_e+ats_e}.e(u_e^{h_e}v_e,g)^{rs_e}}{e(u_e^{h_e}v_e,g)^{rs_e}\prod_{i\in\mathcal{I}_B}((g_T^{at\lambda_e^{(i)}-t\rho_e(i)r_e^{(i)}})(g_T^{t\rho_e(i)r_e^{(i)}}))^{\alpha_e^{(i)}}}$$
$$= \frac{g_T^{\alpha s_e+ats_e}}{\prod_{i\in\mathcal{I}_B} g_T^{at\alpha_e^{(i)}\lambda_e^{(i)}}} = \frac{g_T^{\alpha s_e+ats_e}}{g_T^{at\sum_{i\in\mathcal{I}_B}\alpha^{(i)}{}_e\lambda_e^{(i)}}} = g_T^{\alpha s_e}$$

$$\mathsf{Open}(\check{c}, C_{02}/\Delta_e) = \mathsf{Open}(\check{c}, \breve{d}) = m$$

Non-Repudiation (Publicly Verifiability). Since it is achieved by $\mathcal{CtE}\&\mathcal{S}$ paradigm, our systems also achieve the same.

Dynamic property. In dynamic attribute-based system, a new attribute can be added dynamically to the system without re-issuing the whole secret key of the user. Here a user sends it's one secret key component, viz, $L := g^t R_0'$ to the PKG and then PKG will send the secret key component corresponding to the new attribute : Suppose *att* is a new attribute, then PKG computes $T_{att} := g^{t_{att}}$ by choosing $t_{att} \xleftarrow{\mathrm{U}} \mathbb{Z}_N$, keeps t_{att} to itself and adds T_{att} to $\mathcal{PP}$. Then, it sets $K_{att} := L^{t_{att}} R_{att}$ by picking $R_{att} \xleftarrow{\mathrm{U}} \mathbb{G}_{p_3}$ and returns it to the user.

6 Security Proof of CP-ABSC

Theorem 3. *The proposed attribute-based signcryption scheme in section 5 is perfectly private. (The Signer Privacy for CP-ABSC can be defined in similar manner as in CP-ABS. The details will be found in full version.)*

Theorem 4. *If DSG1, DSG2 and DSG3 assumptions hold, H_e is a collision resistant hash function and Π_{Commit} has hiding property, then our proposed basic CP-ABSC scheme in section 5 is adaptively secure.*

Theorem 5. *If DSG1, DSG2 and DSG3 assumptions hold for $\mathcal{J}$, the primitive commitment scheme Π_{Commit} has relaxed-binding property and H_s is a collision resistant hash function, then the proposed basic CP-ABSC scheme in section 5 is adaptive-predicates existential unforgeable.*

7 Extension to Strongly Unforgeable CP-ABSC

Here in this section, we describe our strongly unforgeable and IND-CCA2 secure CP-ABSC scheme for access policies represented by the monotone span programs. This scheme follows almost the same structure of weak unforgeable and IND-CCA2 secure CP-ABSC described in section 5. But to protect the signcryption from forging, we bind all the components by a strongly unforgeable OTS scheme which we call "Commit then Encrypt and Sign then Sign" ($\mathcal{CtE}\&\mathcal{StS}$)

paradigm. Although the similar type of generic constructions using strongly unforgeable OTS scheme are available in the literature [9,20] in the context of ABE and ABS, here we do not apply the OTS scheme in straightforward way because of the following reasons: (a) we no more assume the relaxed-binding property of the commitment scheme for strong unforgeability, and (b) to reuse the part of IND-CCA2 security proof of the construction described in section 5 for the current CP-ABSC construction.

We just give a short description of our strongly unforgeable and IND-CCA2 secure CP-ABSC construction, since it follows the CP-ABSC in section 5 and the idea of strongly unforgeable CP-ABS stated above. Let $\Pi_{\mathsf{Commit}} := (\mathsf{C.Setup}, \mathsf{Commit}, \mathsf{Open})$ be a commitment scheme. Let $\Pi_{\mathsf{wABS}} := (\mathsf{wABS.Setup}, \mathsf{wABS.KeyGen}, \mathsf{wABS.Sign}, \mathsf{wABS.Ver})$ and $\Pi_{\mathsf{ABE}} := (\mathsf{ABE.Setup}, \mathsf{ABE.KeyGen}, \mathsf{ABE.Encrypt}, \mathsf{ABE.Decrypt})$ be the CP-ABS scheme and CP-ABE scheme respectively used in section 5. Let $\Pi_{\mathsf{OTS}} := (\mathsf{Gen}, \mathsf{OTS.Sign}, \mathsf{OTS.Ver})$ be a strong unforgeable OTS scheme. Demonstrated below are only two routines, Signcrypt and Unsigncrypt as rest are same as in section 5.

–$\mathsf{Signcrypt}(\mathcal{PP}, m, \mathcal{SK}_A, \Gamma_s, \Gamma_e)$: It first runs $(\breve{c}, \breve{d}) \longleftarrow \mathsf{Commit}(m)$, $(\mathsf{verk}, \mathsf{signk}) \longleftarrow \mathsf{Gen}(1^\kappa)$. Then, it executes in parallel $\sigma_w := (\boldsymbol{S}_0, \ldots, \boldsymbol{S}_{\ell_s}) \longleftarrow \mathsf{wABS.Sign}(\mathcal{PP}, \breve{c}||\mathsf{verk}||\Gamma_e, \mathcal{SK}_A, \Gamma_s)$ and $\varrho_0 := (\boldsymbol{C}_0, \ldots, \boldsymbol{C}_{\ell_e}) \longleftarrow \mathsf{ABE.Encrypt}(\mathcal{PP}, \breve{d}, \Gamma_e)$. Then it computes $h_e := H_e(\Gamma_e, \Gamma_s, \breve{c}, \mathsf{verk}, \varrho_0, \sigma_w)$, $C_{\ell_e+1} := (u_e^{h_e} v_e)^{s_e}$ and $\sigma_o \longleftarrow \mathsf{OTS.Sign}(h_e||C_{\ell_e+1}, \mathsf{signk})$. Now it sets the signature part of the signcryption $\sigma_s := (\sigma_w, \sigma_o, \mathsf{verk})$ and the ciphertext part of the signcryption $\varrho := (\varrho_0, C_{\ell_e+1})$. It returns the signcryption $\Upsilon := (\breve{c}, \sigma_s, \varrho)$.

–$\mathsf{Unsigncrypt}(\mathcal{PP}, \Upsilon, \mathcal{SK}_B, \Gamma_s, \Gamma_e)$: It first parses Υ as $(\breve{c}, \sigma_s, \varrho)$, where $\sigma_s := (\sigma_w, \sigma_o, \mathsf{verk})$. Then it runs in parallel $\mathsf{flag}_o \longleftarrow \mathsf{OTS.Ver}(\sigma_w, \sigma_o, \mathsf{verk})$, $\mathsf{flag}_w \longleftarrow \mathsf{wABS.Ver}(\mathcal{PP}, \breve{c}||\mathsf{verk}||\Gamma_e, \sigma_w, \Gamma_s)$ and $\breve{d} \longleftarrow \mathsf{ABE.Decrypt}(\mathcal{PP}, \varrho, \mathcal{SK}_B, \Gamma_e)$. If $\mathsf{flag}_o = 1$ and $\mathsf{flag}_w = 1$, it returns $\mathsf{Open}(\breve{c}, \breve{d})$ else $\perp$.

Correctness, Dynamic property and Non-repudiation. These are immediate from that of section 5.

Theorem 6. *The proposed CP-ABSC scheme in section 7 is perfectly private.*

Theorem 7. *If DSG1, DSG2 and DSG3 assumptions hold, H_e is a collision resistant hash function, Π_{Commit} has hiding property and Π_{OTS} is a strong unforgeable OTS scheme, then our proposed CP-ABSC scheme in section 7 is adaptively secure.*

Proof. The proof can be obtained by the similar approach as in proof of theorem 4 and the argument used for proving CCA2 security in [9].

Theorem 8. *If DSG1, DSG2 and DSG3 assumptions hold for $\mathcal{J}$, Π_{OTS} is a strong unforgeable OTS scheme and H_s, H_e are collision resistant hash functions, then the proposed basic CP-ABSC scheme in section 7 is existential strong existential unforgeable.*

Acknowledgements. One of the authors would like to thank R C Bose Centre for Cryptology and Security, ISI Kolkata for the financial support. Moreover, authors pay their sincere thanks to Dr. Mridul Nandi, ISI Kolkata and anonymous reviewers for their comments and suggestions that helped in polishing the technical and editorial content of this paper.

References

1. An, J.H., Dodis, Y., Rabin, T.: On the security of joint signature and encryption. In: Knudsen, L.R. (ed.) EUROCRYPT 2002. LNCS, vol. 2332, pp. 83–107. Springer, Heidelberg (2002)
2. 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)
3. Baek, J., Steinfeld, R., Zheng, Y.: Formal proofs for the security of signcryption. In: Naccache, D., Paillier, P. (eds.) PKC 2002. LNCS, vol. 2274, pp. 80–98. Springer, Heidelberg (2002)
4. Bethencourt, J., Sahai, A., Waters, B.: Ciphertext-policy attribute-based encryption. In: IEEE Symposium on Security and Privacy, pp. 321–334. IEEE Press (2007)
5. 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)
6. Boneh, D., Franklin, M.: Identity-Based Encryption from the Weil Pairing. In: Kilian, J. (ed.) CRYPTO 2001. LNCS, vol. 2139, pp. 213–229. Springer, Heidelberg (2001)
7. Boneh, D., Katz, J.: Improved efficiency for CCA-secure cryptosystems built using identity-based encryption. In: Menezes, A. (ed.) CT-RSA 2005. LNCS, vol. 3376, pp. 87–103. Springer, Heidelberg (2005)
8. Boyen, X.: Multipurpose identity-based signcryption. In: Boneh, D. (ed.) CRYPTO 2003. LNCS, vol. 2729, pp. 383–399. Springer, Heidelberg (2003)
9. Canetti, R., Halevi, S., Katz, J.: Chosen-ciphertext security from identity-based encryption. In: Cachin, C., Camenisch, J.L. (eds.) EUROCRYPT 2004. LNCS, vol. 3027, pp. 207–222. Springer, Heidelberg (2004)
10. Chen, C., Chen, J., Lim, H.W., Zhang, Z., Feng, D.: Combined public-key schemes: The case of ABE and ABS. In: Takagi, T., Wang, G., Qin, Z., Jiang, S., Yu, Y. (eds.) ProvSec 2012. LNCS, vol. 7496, pp. 53–69. Springer, Heidelberg (2012)
11. Chen, L., Malone-Lee, J.: Improved identity-based signcryption. In: Vaudenay, S. (ed.) PKC 2005. LNCS, vol. 3386, pp. 362–379. Springer, Heidelberg (2005)
12. Damgård, I., Fujisaki, E.: A statistically-hiding integer commitment scheme based on groups with hidden order. In: Zheng, Y. (ed.) ASIACRYPT 2002. LNCS, vol. 2501, pp. 125–142. Springer, Heidelberg (2002)
13. Dent, A.W., Fischlin, M., Manulis, M., Stam, M., Schröder, D.: Confidential signatures and deterministic signcryption. In: Nguyen, P.Q., Pointcheval, D. (eds.) PKC 2010. LNCS, vol. 6056, pp. 462–479. Springer, Heidelberg (2010)
14. Emura, K., Miyaji, A., Rahman, M.S.: Dynamic attribute-based signcryption without random oracles. International Journal of Applied Cryptography 2(11), 199–211 (2012)

15. Gagné, M., Narayan, S., Safavi-Naini, R.: Threshold attribute-based signcryption. In: Garay, J.A., De Prisco, R. (eds.) SCN 2010. LNCS, vol. 6280, pp. 154–171. Springer, Heidelberg (2010)
16. Goyal, V., Pandey, O., Sahai, A., Waters, B.: Attribute-based encryption for fine-grained access control of encrypted data. In: ACM Conference on Computer and Communications Security, pp. 89–98. ACM (2006)
17. Haber, S., Pinkas, B.: Securely combining public-key cryptosystems. In: ACM Conference on Computer and Communications Security, pp. 215–224. ACM (2001)
18. Halevi, S., Micali, S.: Practical and provably-secure commitment schemes from collision-free hashing. In: Koblitz, N. (ed.) CRYPTO 1996. LNCS, vol. 1109, pp. 201–215. Springer, Heidelberg (1996)
19. Hess, F.: Efficient identity based signature schemes based on pairings. In: Nyberg, K., Heys, H.M. (eds.) SAC 2002. LNCS, vol. 2595, pp. 310–324. Springer, Heidelberg (2003)
20. Huang, Q., Wong, D.S., Zhao, Y.: Generic transformation to strongly unforgeable signatures. In: Katz, J., Yung, M. (eds.) ACNS 2007. LNCS, vol. 4521, pp. 1–17. Springer, Heidelberg (2007)
21. Lewko, A., Waters, B.: New proof methods for attribute-based encryption: Achieving full security through selective techniques. In: Safavi-Naini, R., Canetti, R. (eds.) CRYPTO 2012. LNCS, vol. 7417, pp. 180–198. Springer, Heidelberg (2012)
22. 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)
23. Li, J., Au, M.H., Susilo, W., Xie, D., Ren, K.: Attribute-based signature and its applications. In: ACM Conference on Computer and Communications Security, pp. 60–69. ACM (2010)
24. Libert, B., Quisquater, J.-J.: Efficient signcryption with key privacy from gap diffie-hellman groups. In: Bao, F., Deng, R., Zhou, J. (eds.) PKC 2004. LNCS, vol. 2947, pp. 187–200. Springer, Heidelberg (2004)
25. Libert, B., Quisquater, J.-J.: Improved signcryption from q-diffie-hellman problems. In: Blundo, C., Cimato, S. (eds.) SCN 2004. LNCS, vol. 3352, pp. 220–234. Springer, Heidelberg (2005)
26. Maji, H., Prabhakaran, M., Rosulek, M.: Attribute-based signatures: Achieving attribute-privacy and collusion-resistance. Cryptology ePrint Archive, Report 2008/328 (2008), `http://eprint.iacr.org/`
27. Maji, H., Prabhakaran, M., Rosulek, M.: Attribute-based signatures. Cryptology ePrint Archive, Report 2010/595 (2010), `http://eprint.iacr.org/`
28. Malone-Lee, J., Mao, W.: Two birds one stone: Signcryption using RSA. In: Joye, M. (ed.) CT-RSA 2003. LNCS, vol. 2612, pp. 211–226. Springer, Heidelberg (2003)
29. Matsuda, T., Matsuura, K., Schuldt, J.C.N.: Efficient constructions of signcryption schemes and signcryption composability. In: Roy, B., Sendrier, N. (eds.) INDOCRYPT 2009. LNCS, vol. 5922, pp. 321–342. Springer, Heidelberg (2009)
30. Okamoto, T., Takashima, K.: Fully secure functional encryption with general relations from the decisional linear assumption. In: Rabin, T. (ed.) CRYPTO 2010. LNCS, vol. 6223, pp. 191–208. Springer, Heidelberg (2010)
31. 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)

32. Okamoto, T., Takashima, K.: Fully secure unbounded inner-product and attribute-based encryption. In: Wang, X., Sako, K. (eds.) ASIACRYPT 2012. LNCS, vol. 7658, pp. 349–366. Springer, Heidelberg (2012)
33. Ostrovsky, R., Sahai, A., Waters, B.: Attribute-based encryption with non-monotonic access structures. In: ACM Conference on Computer and Communications Security, pp. 195–203 (2007)
34. Pedersen, T.P.: Non-interactive and information-theoretic secure verifiable secret sharing. In: Feigenbaum, J. (ed.) CRYPTO 1991. LNCS, vol. 576, pp. 129–140. Springer, Heidelberg (1992)
35. Sahai, A., Waters, B.: Fuzzy identity-based encryption. In: Cramer, R. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 457–473. Springer, Heidelberg (2005)
36. 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)
37. Vasco, M.I.G., Hess, F., Steinwandt, R.: Combined (identity-based) public key schemes. Cryptology ePrint Archive, Report 2008/466 (2008), `http://eprint.iacr.org/`
38. Waters, B.: Dual system encryption: Realizing fully secure IBE and HIBE under simple assumptions. In: Halevi, S. (ed.) CRYPTO 2009. LNCS, vol. 5677, pp. 619–636. Springer, Heidelberg (2009)
39. Waters, B.: Ciphertext-policy attribute-based encryption: An expressive, efficient, and provably secure realization. In: Catalano, D., Fazio, N., Gennaro, R., Nicolosi, A. (eds.) PKC 2011. LNCS, vol. 6571, pp. 53–70. Springer, Heidelberg (2011)
40. Yamada, S., Attrapadung, N., Hanaoka, G., Kunihiro, N.: Generic constructions for chosen-ciphertext secure attribute based encryption. In: Catalano, D., Fazio, N., Gennaro, R., Nicolosi, A. (eds.) PKC 2011. LNCS, vol. 6571, pp. 71–89. Springer, Heidelberg (2011)
41. Zheng, Y.: Digital signcryption or how to achieve cost (signature & encryption) << cost(signature) + cost(encryption). In: Kaliski Jr., B.S. (ed.) CRYPTO 1997. LNCS, vol. 1294, pp. 165–179. Springer, Heidelberg (1997)

A Ciphertext-Policy Attribute-Base Signcryption

A.1 Definition

A ciphertext-policy Attribute-Base Signcryption(CP-ABSC) scheme consists of four PPT algorithms - Setup, KeyGen, Signcrypt and Unsigncrypt.
Setup: Input: a security parameter κ and a universe of attributes $\mathcal{U}$. Output: public parameters $\mathcal{PP}$ and a master secret $\mathcal{MSK}$.
KeyGen: Input: a set of attributes A, $\mathcal{PP}$ and $\mathcal{MSK}$. Output: a secret key $\mathcal{SK}_A$ corresponding to A.
Signcrypt: Input: $\mathcal{PP}$, a message m, $\mathcal{SK}_A$, a predicate Γ_s (signer policy) with $\Gamma_s(A) = \mathsf{True}$ and another predicate Γ_e (receiver policy). Output: a signcryption Υ for (Γ_e, Γ_s).
Unsigncrypt: Input: $\mathcal{PP}$, a signcryption Υ, $\mathcal{SK}_B$, Γ_s and Γ_e with $\Gamma_e(B) = \mathsf{True}$. Output: a message m if Υ is valid else $\perp$.

A.2 Adaptive-Predicates IND-CCA2 Security of CP-ABSC

A CP-ABSC scheme is *adaptively secure (adaptive-predicates IND-CCA2 secure)* if no PPT adversary A has non-negligible advantage in this game:
Setup: The challenger $\mathscr{B}$ runs $(\mathcal{PP}, \mathcal{MSK}) \longleftarrow \mathsf{Setup}(1^\kappa, \mathcal{U})$ and gives $\mathcal{PP}$ to $\mathscr{A}$.
Query: The adversary $\mathscr{A}$ is given access to the oracles $\mathsf{KeyGen}(\mathcal{PP}, \mathcal{MSK}, .)$, $\mathsf{Signcrypt}(\mathcal{PP}, ., .)$ and $\mathsf{Unsigncrypt}(\mathcal{PP}, ., .)$.
Challenge: $\mathscr{A}$ provides two equal length messages m_0, m_1 and the challenge access policies (Γ_s^*, Γ_e^*) s.t for each set of attributes A queried to $\mathsf{KeyGen}(\mathcal{PP}, \mathcal{MSK}, .)$ oracle, $\Gamma_e^*(A) = \mathsf{False}$. $\mathscr{B}$ picks $b \xleftarrow{\mathrm{U}} \{0, 1\}$. Then, it signcrypts the challenge message m_b using the challenge policies Γ_s^* and Γ_e^* and gives the challenge signcryption Υ_b to $\mathscr{A}$.
Query: Again, $\mathscr{A}$ is given access to $\mathsf{KeyGen}(\mathcal{PP}, \mathcal{MSK}, .)$, $\mathsf{Signcrypt}(\mathcal{PP}, ., .)$ and $\mathsf{Unsigncrypt}(\mathcal{PP}, ., .)$ oracles but if A is a set of attributes queried to $\mathsf{KeyGen}(\mathcal{PP}, \mathcal{MSK}, .)$ oracle and Υ is a unsigncryption query to $\mathsf{Unsigncrypt}(\mathcal{PP}, ., .)$ oracle, then $\Gamma_e^*(A) = \mathsf{False}$ and $\Upsilon \neq \Upsilon_b$.
Guess: $\mathscr{A}$ sends a guess b' to $\mathscr{B}$.
The advantage of $\mathscr{A}$ in above is $\mathsf{Adv}_{\mathscr{A}}^{\mathrm{ABSC-CCA}}(\kappa) = \left|\Pr[b = b'] - \frac{1}{2}\right|$.

A.3 Adaptive-Predicates Unforgeability of CP-ABSC

A CP-ABSC scheme is *adaptive-predicates strong existential unforgeable* if no PPT adversary A has non-negligible advantage in this game:
Setup: Same as in A.2.
Query: The adversary $\mathscr{A}$ is given access to the oracles $\mathsf{KeyGen}(\mathcal{PP}, \mathcal{MSK}, .)$ and $\mathsf{Signcrypt}(\mathcal{PP}, ., .)$.
Forgery: The adversary outputs a signcryption Υ^* for $(m^*, \Gamma_s^*, \Gamma_e^*)$.
$\mathscr{A}$ succeeds in this game if $(\Upsilon^*, m^*, \Gamma_s^*, \Gamma_e^*) \neq (\Upsilon^{(i)}, m^{(i)}, \Gamma_s^{(i)}, \Gamma_e^{(i)})$, where $\Upsilon^{(i)}$ is the reply by $\mathsf{Signcrypt}$ oracle for $(m^{(i)}, \Gamma_s^{(i)}, \Gamma_e^{(i)})$, Γ_s^* does not accept any set of attributes queried to KeyGen oracle and $\mathsf{Unsigncrypt}(\mathcal{PP}, \Upsilon^*, \mathcal{SK}_B, \Gamma_s^*, \Gamma_e^*) = m^*$, where $\Gamma_e^*(B) = \mathsf{True}$.
The advantage of $\mathscr{A}$ in above game is the success probability of $\mathscr{A}$.

B Mechanism for Full Construction

Although the technique is available in [22] but for self-containment, in this section we briefly demonstrate it. The mechanism described here is for both CP-ABS and CP-ABSC supporting MSPs. For full construction, the row labeling functions of span programs are not assumed to be injective. If we allow an attribute to repeat in the span programs at most $\mathfrak{M}$ time and the size of the universe $\mathcal{U}$ is n, then the size of new universe $\mathcal{U}'$ for the full construction will be $n\mathfrak{M}$. Basically in this full construction, for each attribute $\chi \in \mathcal{U}$, we consider $\mathfrak{M}$ copies of χ in $\mathcal{U}'$. To enumerate each copy, we assign a label say j to the attribute say χ, i.e., $\mathcal{U}' := \{(\chi, j) | \chi \in \mathcal{U}, j \in [\mathfrak{M}]\}$. Similarly, for any access

policy $\Gamma := (M, \rho)$ if $\rho(i) = \chi$ and the attribute χ appears j^{th} time, then we label the i^{th} row by (χ, j), i.e., we have a new row labeling function ρ' defined by $\rho'(i) := (\chi, j)$. Likewise if A is a set of attributes corresponding to $\mathcal{U}$, then $A' := \{(\chi, j) | \chi \in A, j \in [\mathfrak{M}]\}$ is the set of attributes for $\mathcal{U}'$. Then, we have that the set of attributes A satisfies the policy (M, ρ) if and only if A' satisfies (M, ρ'). Due to this technique, the sizes of public parameters and key increase by a factor linear to $\mathfrak{M}$, but the sizes of signature (resp. signcryption) and the cost of sign and ver (resp. signcrypt and unsigncrypt) for CP-ABS (resp. CP-ABSC) remain unchanged.