computing@computingonline.net
www.computingonline.net

Print ISSN 1727-6209
On-line ISSN 2312-5381
International Journal of Computing

ATTRIBUTE-BASED AUTHENTICATION SCHEMES: A SURVEY

Huihui Yang [1], Vladimir Oleshchuk [2]

[1] University of Agder, Norway, huihui.yang@uia.no, http://www.uia.no/kk/profil/huihuiy
[2] University of Agder, Norway, Vladimir.Oleshchuk@uia.no, http://www.uia.no/kk/profil/vladimao

Abstract: Attribute-based authentication (ABA) is a way to authenticate users via attributes which are the properties of those to be authenticated, for example, resources, contextual information (time, location, etc.) or their combination. In ABA schemes, attributes instead of identity are requested to be presented or even evidence showing that users own the required attributes is enough, so it is more flexible and privacy-preserving compared with traditional identity-based authentication. In this paper, we first explain the general structure and security requirements of ABA schemes, and then give an example to demonstrate their cryptographic construction. Next, we analyze recent work and discuss future research topics on the construction of ABA schemes, including attribute tree building, cryptographic construction, security models, hierarchy, traceability and revocation.

Keywords: Authentication, attribute-based authentication, attribute tree, anonymity.

1. INTRODUCTION

Authentication is usually required before resource accessing and used as part of access control, such as attribute-based access control [1]. In traditional identity-based authentication [2], a user is required to show its identity specific information to get authenticated by another party, so the user accesses the resources at the cost of losing her anonymity. In some scenarios where resources are highly credential and strictly controlled, identity information is critical and a necessity, while in some situations a set of non-identity attributes is enough, for example, online shopping or location service. ABA [3, 4, 5] is an authentication approach where only necessary attributes instead of identity information are needed. In ABA schemes, the authentication of a user is usually triggered by the user's request for some service. Once the service provider receives the request, it sends back attribute requirements for the service. If the user owns the requested attributes, it sends back evidence to the service provider, where the evidence is usually a signature. The user's request will be granted if the evidence is valid and otherwise it will be denied. During the whole process, the user only needs to prove that it owns the requested attributes without revealing its identity. Therefore, ABA is a promising authentication approach to protect users' privacy and keep their identities anonymous.

Researches about ABA schemes can generally be divided into several fields, including system structures [4, 6], cryptographic construction and security requirements [3, 8], and policy specification [5] and so on. In this paper, we mainly focus on ABA scheme construction. There are two main contributions in this paper. First, we review recent work on ABA construction, in topics of attribute trees building, cryptographic construction, security models, hierarchy, traceability and revocation, with analysis of their advantages and disadvantages. Second, we discuss open problems in these fields and propose potential directions how to solve them.

This paper is organized as follows. In Section 2, we describe the general structure and workflow of ABA schemes first, and then define their main security properties. Followed in Section 3, a specific example about how attribute trees and ABA schemes are constructed is given. In Section 4, we analyze recent research results and discuss open problems in fields of attribute trees building, cryptographic construction, security models, hierarchy, traceability and revocation respectively. The last section is a brief conclusion about the work in this paper.

2. ABA SCHEME INTRODUCTION

In this section, we describe the general structure and workflow of ABA schemes. Based on the

structure, we summarize and explain the security requirements that ABA scheme should satisfy.

2.1. THE STRUCTURE AND WORKFLOW OF ABA SCHEMES

The structure and workflow of ABA schemes can be illustrated in Fig. 1. There are usually three types of entities in ABA schemes, authorities, users and verifiers, where authorities can be divided into central authority, attribute authority, revocation authority and opener. The way how they interact with each other and how the authentication is carried out can be described as follows.

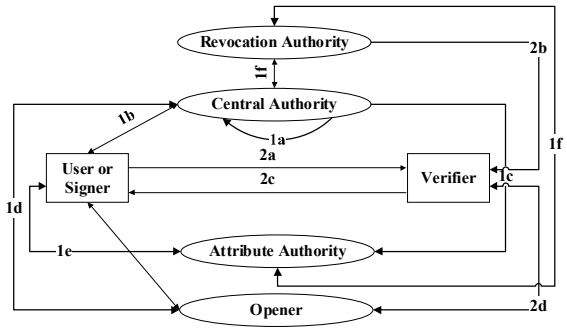

Fig. 1 – Structure and Workflow of ABA Schemes.

1. The first stage is system set up.
   a) The central authority generates system public and private parameters.
   b) The user communicates with the central authority and gets its secret keys. There are two ways to generate users' secret keys. In the first way, the central authority chooses users' secret keys and sends them to users in a secret channel. In the second way, users negotiate with the authority to generate their secret keys by a "join in" protocol. Due to the ways how users' secret keys are generated, ABA schemes are divided into static and dynamic.
   c) The attribute authority retrieves these system parameters and generates private and public attribute key pairs.
   d) The opener communicates with the central authority and gains the tracking keys.
   e) The user communicates with the attribute authority and gets its private attribute keys.
   f) Revocation authority communicates with both the central authority and the attribute authority to establish a database of revocation information.
2. The second stage is signature generation, verification and possibly opening.
   a) After receiving a challenge or attribute requirements from the verifier, the signer sends its signature to the verifier, where the signature is generated by signing a message with the signer's attribute keys.
   b) The verifier retrieves revocation information from the revocation authority. If the signer and related attribute keys are not revoked, the verifier checks the validity of the signature and sends a response to the signer.
   c) If the identity of the signer needs to be revealed, the verifier delivers the signature to the opener. The opener uses its tracking keys to open the signature and reveal the signer's identity.

2.2. SECURITY REQUIREMENTS OF ABA SCHEMES

To our best knowledge, the first systematic description of ABA schemes is given by D. Khader in his PhD thesis [3]. In his thesis, Khader listed some properties and security requirements an ABA scheme should satisfy. Based on the descriptions in [3], we define the following five security requirements.

1. **Anonymity:** To achieve basic anonymity, identities of signers should be protected. Furthermore, even signers' attributes should be protected, and they only have to prove that they own the required attributes. This property is the main security requirement of an ABA scheme and is mandatory.
2. **Unforgeability:** The signer's signature should not be able to be forged by an outsider that does not belong to the system. In some systems, the signature is even required unforgeable for system authorities. However, in a system where authorities generate all keys and secrets, the authorities obviously can forge all signatures. Therefore, "unforgeable" is defined differently in different ABA schemes. However, any system should provide at least the basic level of "unforgeability", i.e, for the outsiders.
3. **Unlinkability:** Given two signatures, if it is impossible to decide whether they are generated by the same signer, the ABA scheme is unlinkable. If a system does not satisfy it, given enough signatures, there is a possibility to reveal the signer's identity.
4. **Coalition resistance:** The signer can only generate the signature if he or she has all the required attributes. It should be impossible for different users to collude and generate a valid

signature together if they as a whole have all the required attributes. If a system satisfies this requirement, it is coalition resistant.

5. **Traceability:** Given a valid signature, if the opener can successfully track the signer's identity, the system is traceable. It is a useful security requirement for some applications, for instance, obtaining evidence for legitimate issues.

3. CRYPTOGRAPHIC CONSTRUCTION OF ABA SCHEMES: AN EXAMPLE

Attribute trees represent what attribute requirements a signer should satisfy to get authenticated and the attribute tree building is part of the ABA scheme construction. In this section, we first describe how to build an attribute tree, and then give an example to demonstrate how to construct ABA schemes.

3.1. ATTRIBUTE TREE CONSTRUCTION

Attribute trees [3] are also called access trees [9] in attribute-based access control, and they are constructed in the same way. An attribute tree is a tree structure where leaves are attributes and interior nodes are threshold gates. For an interior node$x$, let l_x and k_x be the numbers of children and the threshold respectively. It represents logical "AND" and "OR" respectively when $k_x$is equal to or less than l_x. This rule applies for the two ways of constructing attribute trees, i.e., top-to-down and down-to-top.

Usually attribute trees are built from top to down. To the best of our knowledge, there is only one paper [10] in which attribute trees are built from down to top. However, since the example we will give in this section is top-to-down attribute tree based, we will not discuss the down-to-top approach here. More details will be covered later and can also be referred to in [10].

Suppose there is a system, the attribute set is $\Psi = \{att_1, \cdots, att_8\}$, where Ψ is the set of all system attributes and $att_j (1 \le j \le 8)$ are elements. Assume there is an attribute based logical statement $(att_1 \wedge att_3) \vee (att_4 \wedge (att_7 \vee att_8))$, which can be shown as the tree structure in Fig. 2. Attribute subset related to the logical statement is denoted by $\{att_1, att_3, att_4, att_7, att_8\} \subset \Psi$.

Based on the basic knowledge about attribute trees, we will explain how to construct a down-to-top attribute tree in the following. In an attribute tree, each node x is indexed with a random number $d(x)$, which is from 1 to 9 here. Each interior node (node 6, 7, 8, 9 in Fig. 2) is bound to a polynomial $q_i(x)(i = ind(x))$ (Refer to [3] for more details.) and its degree is at most the same as the number of its children, which is 2 here. Polynomials are constructed from top to down as follows.

1. Randomly choose a secret value s_9 and a polynomial $q_9(x)$ for the root, where $q_9(0) = s_9$ holds.
2. Compute $s_7 = q_9(7)$ and $s_8 = q_9(8)$. Choose $q_7(x)$ randomly such that $s_7 = q_7(0)$. $q_8(x)$ is computed in the same way.
3. Repeat Step 2 until all polynomials related to interior nodes are constructed.
4. There are no polynomials bonded to leaf nodes. Suppose the parent of a leaf node x is y, then $s_x = q_y(x)$.

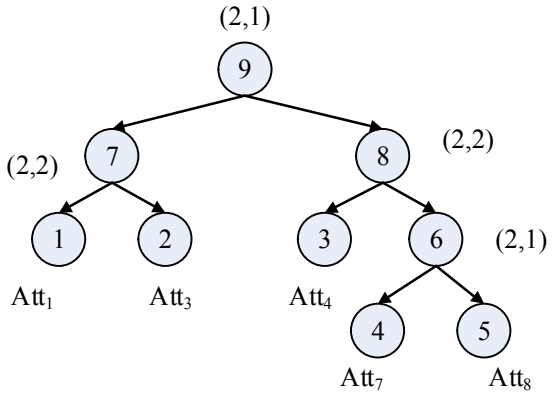

Fig. 2 – Top-to-down Attribute Tree Construction.

Once an attribute tree is built, we can combine it with the knowledge of bilinear group [12] and compute attribute keys. Suppose $e: G_1 \times G_2 \to G_3$ is a bilinear map, where G_1, G_2 and G_3 are of prime order q. A_i is user U_i's secret key, and the attribute subset owned by U_i is Ψ_i. Each attribute att_j is related to a master attribute key t_j and public key $bpk_j = w^{t_j}$, where $w \in G_2$ is public. For U_i's attribute att_j, the private attribute key is $T_{i,j} = A_i^{1/t_j}$. After all the polynomials are calculated, assign $D_j = bpk_j^{q_j(0)}$ to each leaf node or attribute att_j.

In the following, we give some definitions and get some conclusions that will be used later in the cryptographic construction of Khader's Scheme. Define $CT_j = T_{i,j}^{\beta} (\beta \in \mathbb{Z}_q^*)$ and

$$F_z = \begin{cases} e(CT_j, D_j), if\ j \in \Psi_i; \\ \perp, if\ j \notin \Psi_i. \end{cases}$$

For a non-leaf node x in attribute subtree Γ_i, let $\Delta_{\hat{S}_x, ind(z)} = \prod_{l \in \{\hat{S}_x - ind(z)\}} \frac{l}{ind(z) - l}$, where $\hat{S}_x$ is the subset of all x's children z belonging to Ψ_i. Suppose $root_i$ is the root of attribute subtree Γ_i.

Let
$\overline{D}_i = \{D_1, \cdots, D_{|\Psi_i|}\}$, $\overline{T}_i = \{T_{i,1}^{\beta}, T_{i,2}^{\beta}, \cdots, T_{i,|\Psi_i|}^{\beta}\}$,
and then we have

$$F_{root_i} = TVerify(\overline{D}_i, \Psi_i, \overline{T}_i) = \prod_{z \in \hat{S}_x} F_z^{q_z(0) \Delta_{\hat{S}_{root_i,j}}}$$
$$= \prod_{z \in \hat{S}_x} e(CT_j, D_j)^{q_z(0) \Delta_{\hat{S}_{root_i,j}}}$$
$$= \prod_{z \in \hat{S}_x} e(T_{i,j}^{\beta}, D_j)^{q_z(0) \Delta_{\hat{S}_{root_i,j}}}$$
$$= \prod_{z \in \hat{S}_x} e(A_i^{\beta}, w)^{q_z(0) \Delta_{\hat{S}_{root_i,j}}}$$
$$= e(A_i^{\beta}, w)^{q_{root_i}(0)}.$$

3.2. REVIEW OF KHADER'S SCHEME

Khader's scheme is described as an example to show how to construct a dynamic ABA scheme based on a group signature in [3]. The general structure and workflow of Khader's schemes are illustrated in Fig. 3. Compared with Fig. 1, the main difference is that an "Issuer" is separated from the central authority. The main responsibility of the issuer is to control users' joining in by running the "join in" protocol between itself and users. In step $1b.1$, the issuer retrieves system parameters from the central authority. Step $1b.2$ is the "join in" protocol, where the issuer and the user negotiate with each other to generate the user's secret key.

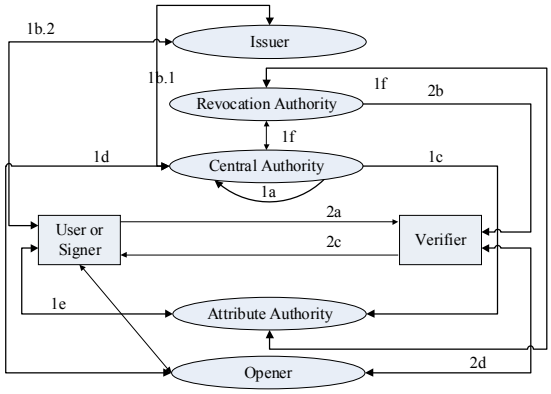

Fig. 3 – Structure and Workflow of Khader's schemes.

The algorithm to setup Khader's scheme runs as follows.

- **System setup** G_1,G_2 and G_3 are three multiplicative groups of prime order $q$and $G_1 \times G_2 \to G_3$ is a bilinear map.$H: \{0,1\}^* \to \mathbb{Z}_q^*$ is a hash function. Randomly select $\xi_1, \xi_2, \gamma \in \mathbb{Z}_q^*$, $h \in G_2$. Select generators $g_1, g_2, g_3, g_4 \in G_1$ such that $g_1 = {g_4}^{\xi_1}$ and $g_2 = {g_4}^{\xi_2}$. Then the system private parameter is $S_{pri} =< \gamma, tk >$ $(tk = \{\xi_1, \xi_1\})$ where γ is the issuer's key, and the public parameter S_{pub} is $< G_1, G_2, G_3, e, H, g_1, g_2, g_3, g_4, h, w >$ $(w = h^{\gamma})$.
- **User key generation** U_i generates a public and private key pair $upk[i]$ and $usk[i]$ for itself.
- **Join in protocol** The protocol runs as follows.
  1. The issuer selects $v \in \mathbb{Z}_q^*$ and sends g_1^v to U_i.
  2. U_i selects $y_i \in \mathbb{Z}_q^*$ and sends $(g_1^v)^{y_i}$ to the issuer.
  3. The issuer selects $x_i \in \mathbb{Z}_q^*$, computes $A_i = (g_1{}^{y_i} g_3)^{1/(x_i+\gamma)}$ and sends $A_i g_1{}^{y_i}$ to U_i.
  4. U_i extracts A_i from $A_i g_1{}^{y_i}$ and checks whether $A_i \in G_1$. If so, U_i uses its private key to sign A_i, get the signature $S = Sign(A_i, usk[i])$ and sends it to the issuer.
  5. The issuer uses U_i's public key $upk[i]$ to verify the signature. If it is valid, it saves $(upk[i], A_i, x_i, S)$ to a database, where S is considered as U_i's commitment to the private key part selected by itself. The issuer sends $x_i g_1{}^{y_i}$ to U_i.
  6. U_i computes x_i and verifies ${A_i}^{(x_i+\gamma)} = g_3 g_1{}^{y_i}$. If the equation holds, A_i is the registration key.U_i's secret key base is $bsk[i] =< A_i, x_i, y_i >$.
- **User attribute key generation** Suppose the attribute set owned byU_i is Ψ_i. For $att_j \in \Psi_i$, the private attribute key is $T_{i,j} = {A_i}^{1/t_j}$and its general secret key is $gsk =< bsk[i], T_{i,1}, \cdots, T_{i,|\Psi_i|} >$.
- **Signature generation and verification** Suppose $M$is the message to sign, and then protocol runs as follows:
  1. Suppose V is the verifier. It collects all public key bases of attributes in Ψ, randomly selects $\alpha \in \mathbb{Z}_q^*$, computes $\overline{D} = \{D_1, \cdots, D_{|\Psi|}\}$ and constructs attribute tree Γ, where $root$ is its root. Then V sends $(\Psi, \overline{D})$ to U_i.
  2. U_i first calculates $F_{root} = TVerify(\overline{D}, \Psi, \overline{T})$ as described in the process of attribute tree construction. Here we have $\overline{T} = \{T_{i,1}^{\beta}, T_{i,2}^{\beta}, \cdots, T_{i,|\Psi|}^{\beta}\}$. Next U_i selects $\beta_1, \beta_2 \in \mathbb{Z}_q^*$, computes $\beta = \beta_1 \beta_2$ and computes F_{root} as described before. Next U_i selects $\zeta, \delta, r_\zeta, r_\delta, r_x, r_z \in \mathbb{Z}_q^*$ and computes

$C_1 = {g_4}^{\zeta}, C_2 = A_i {g_1}^{\zeta}, C_3 = {g_4}^{\delta},$
$C_4 = A_i {g_2}^{\delta}, C_5 = e({g_2}^{\delta} {A_i}^{1-\beta_2}, w)^{\beta_1},$
$C_6 = w^{\beta_1}, R_1 = {g_4}^{r_\delta},$
$R_2 = e(C_2, h)^{r_x} e(g_1, w)^{-r_\zeta} e(g_1, h)^{-r_z},$
$R_3 = {g_4}^{r_\delta}, R_4 = {g_1}^{r_\zeta} {g_2}^{-r_\delta},$
$c = H(M, C_1, C_2, C_3, C_4, C_5, C_6, R_1, R_2, R_3, R_4)$
$s_\zeta = r_\zeta + c\zeta, s_\delta = r_\delta + c\delta,$
$s_x = r_x + cx_i, z = x_i\zeta + y_i,$
$s_z = r_z + cz$. Finally, U_i sends $\delta = (C_1, C_2, C_3, C_4, C_5, C_6, F_{root}, c, s_\zeta, s_\delta, s_x, s_z)$ to V.

3. First of all, V verifies the attributes by checking $F_{root}^{1/\alpha} C_5 = e(C_4, C_6)$. Next V verifies that U_i is not revoked. If U_i is not revoked, the signature verification by V is carried out by the following calculations.
$\bar{R}_1 = {g_4}^{s_\zeta} {C_1}^{-c},$
$\bar{R}_2 = e(C_2, h)^{s_x} e(g_1, w)^{-s_\zeta} e(g_1, h)^{-s_z} (\frac{e(C_2, w)}{e(g_2, h)})^c,$
$\bar{R}_3 = {g_4}^{s_\delta} {C_3}^{-c},$
$\bar{R}_4 = {g_1}^{s_\zeta} {g_2}^{-s_\delta} / (C_2 {C_4}^{-1})^c.$
Finally, U_i checks whether $c = H(M, C_1, C_2, C_3, C_4, C_5, C_6, \bar{R}_1, \bar{R}_2, \bar{R}_3, \bar{R}_4)$ holds. If not, the signature will be rejected.

- **Signature open** The opener uses its tracking key $tk = \{\xi_1, \xi_2\}$ to open a signature δ by computing $A_i = \frac{C_2}{(C_1)^{\xi_1}} = C_4 / (C_3)^{\xi_2}$.
- **Judge** To prove that A_i has been used in the opened signature δ. A zero knowledge proof runs as follows.
  1. V randomly picks $rnd \in \mathbb{Z}_q^*$ and sends $td = ({C_1}^{rnd}, {C_2}^{rnd})$ to the opener.
  2. The opener calculates $P = \frac{{C_2}^{rnd}}{{C_1}^{rnd\xi_1}} = {A_i}^{rnd}$ and sends the proof P to V.
  3. V extracts A_i from P.

Khader's scheme satisfies several security requirements described earlier in Section 2 and they are proved as theorems in [3]. The satisfied security requirements include anonymity, traceability, unforgeability and non-frameability. One thing to notice is that "non-frameability" here means even the authority cannot impersonate an honest user to generate a valid signature on behalf of the honest user.

4. OPEN PROBLEM DISCUSSION

After getting a view how an ABA scheme can be constructed, we discuss several topics about ABA schemes and existing issues in this part. We briefly review recent approaches in attribute tree building, general framework for ABA scheme construction, cryptographic assumptions, hierarchy, traceability and revocation respectively, and then we analyze both advantages and disadvantages of existing approaches in these areas. Finally, we point out open problems in each field and propose some directions how to solve these problems.

4.1. ATTRIBUTE TREE BUILDING

As mentioned earlier, attribute trees can also be built from down to top. The down-to-top approach [10, 37] is designed for a dynamic ABA scheme. Here "dynamic" means the attribute trees can be changed without regenerating related parameters, and it is different from the "dynamic" we will discuss in later in this section. A down-to-top attribute tree is built in two steps. First of all, a central attribute tree Γis built, and then it is simplified to obtain the right attribute tree that exactly represents the logical statement. The key idea to build the central attribute key in down-to-top approach is that we add dummy nodes to turn an "OR" threshold gate into an "AND" one. More specifically, if $k_x < l_x$ holds for an interior node x, we add $l_x - k_x$ dummy nodes Dum_x as its children, index them as $j = ind(y)(y \in Dum_x)$ and assign a random value $d_i = q_x(y)$ to each dummy node. In this way, each interior node can be considered as a logical "AND" and then the central attribute tree T^0 can be constructed in the following steps.

1. The system attribute set Ψ is indexed first and all attributes are considered as leaves of Γ.
2. A secret value s_i $(i = ind(x))$ is randomly selected and assigned to each leaf node x.
3. For an interior node x, suppose its children set is $Children(x)$. Since a polynomial $q_i(x)$ passes through points, the coordinates of which are represented as $(y, q_i(y))(y \in \{Children(x)\})$,$q_i(x)$can be computed by Lagrange's theorem [12].
4. Repeat Step 3 until all polynomials are computed until the root.

After the central attribute tree is built, we simplify it according to different attribute subset to gain the attribute tree we need. The simplification is carried out in the following steps. Please read [10] for more details.

1. Delete all leaves that do not belong to the attribute subset, for example, $\Psi - \Psi_i = \{att_2, att_5, att_6\}$ in the example given in top-to-down attribute tree construction in Section 3.

2. Delete an interior node with its descendants if its children are less than the threshold. The remaining part is the required attribute tree.

Compared with the top-to-down approach, the down-to-top approach is more flexible. As long as the attribute subset Ψ_i is a subset of Ψ, the related attribute tree Γ^i can be obtained by simplifying Γ and regenerating parameters can be avoided to some extent. One drawback of this approach is that the central attribute tree Γ and attribute set Ψ are bigger than those used in top-to-down approach. If the attribute tree is comparatively fixed, then a big attribute tree is a waste of resources. Therefore, the down-to-top approach is more beneficial in a dynamic environment.

There is a common open problem for current attribute trees built in both approaches described above. They constraint to represent attributes like "have" and are not designed to express logical relations like "$\leq$" and so on. For example, if a system wants to divide 24 hours into three intervals, 0:01 to 8:00, 8:01 to 16:00 and 16:01 to 24:00, it needs three attribute elements to express it. As a result, the system needs to generate more attribute keys and use more resources. In [5], this problem is compensated by attribute predicates, but it is an approach offered by specification language instead of by the construction way of attribute tree itself.

4.2. GENERAL FRAMEWORK OF ABA SCHEME CONSTRUCTION

In this paper, we divide ABA schemes into two types, i.e., static and dynamic. The concepts of "static" and "dynamic" were first proposed in [13] to describe different group signatures [14, 15, 16]. The main difference is whether the system has the ability to add members into the group at any point. If all members in the group are decided when an ABA system is setup, it is static. Otherwise it is dynamic. In most recent work, however, "dynamic" is used to describe systems where users are involved to generate their secret keys to prevent key escrow, and it is realized by a "join in" protocol. According to this standard, the Khader's Scheme we described in Section 3 is dynamic. Compared with dynamic schemes, all keys are generated by the authority in static ones, and there is no need for an issuer and "join in" protocol. The scheme structure can be described in Fig. 1 as we have used to explain the workflow of ABA schemes.

There have already been some researches in ABA scheme construction, but to the best of our knowledge, there is only one paper [3] in which a general framework is defined to construct static ABA schemes. The general framework in [3] can be used to construct attribute based authentication schemes from group signatures, for example, the work done by Boneh and Shacham [17] and BMW group signature scheme [16]. We will not explain the details about the general framework, but the conceptual idea is as follows. To construct an ABA scheme, two necessary parts are needed: an attribute tree and a signature scheme on which ABA schemes are built. To simplify explanations, we call these signature schemes "base schemes". Attribute trees are used to represent attribute requirements and to authenticate whether the user owns required attributes. The base scheme is used to generate signatures for anonymous authentication. As long as the base scheme is fully anonymous and traceable, the output ABA scheme is also fully anonymous and traceable. To only achieve anonymity, base schemes can be group signatures, ring signatures, attribute-based signature (ABS) [18] and attribute-based encryption (ABE) [9] schemes. However, ring signatures usually do not provide traceability and thus cannot be used as a base scheme to construct a traceable ABA scheme.

The way how to construct ABA schemes based on either group signature or ring signature are based on pairings. To our best knowledge, there is only one ABS scheme [35] is constructed without pairings. The scheme in [35] is obtained by applying Fiat-Shamir transform [36] to the attribute-based identification (ABID) proposed by the authors. It achieves advantages in efficiency without pairing but its security proof is based on random oracle model which limits its application to some extent.

Currently, there are three open issues in the area of general framework of ABA scheme construction.

1. In the general framework to construct static ABA schemes, only anonymity and traceability are discussed. How to achieve other security requirements and what requirements a base scheme should satisfy to build an ABA scheme with different security requirements are still not solved.
2. There is no general framework to build a dynamic ABA scheme from base signature schemes.
3. From the summary of recent work in Table 1 presented later, we can see that most schemes are static. As explained before, the "join in" protocol in dynamic schemes can prevent group members and authorities from impersonating honest users and is thus more secure. Therefore, more dynamic ABA schemes rather than static ones are preferable considering security reasons.

Security Models ABA schemes are built on bilinear groups [11] and their security is based on hard problems in groups or bilinear groups, such as Diffie-Hellman problem (DH) [3], DLP [19], q-

SDH [3] and so forth. Moreover, the security requirements of ABA scheme are usually proved under some models, including random oracle model [15], generic model [20] and standard model [21]. The dilemma in cryptographic application is that there is always a gap between theory and practice, and the same for ABA schemes. The realization of their security requirements are based on these hard problems, cryptographic assumption and security models. However, using too many assumptions constrains the usability of these schemes. Some schemes are proved secure in random oracles, but their security is still a question once they are implemented in real system. Some schemes are even unpractical because of too many assumptions and too complicated computations. Security requirements of most ABA schemes are proved under random oracle. However, compared with random model, assumptions in generic and standard model are comparatively less and thus more preferable models to construct more practical ABA schemes.

Considering the cryptographic constructions and assumptions used, ABA schemes, attribute-based authorization in access control, ABS and ABE are quite similar. In the following, we summarize and compare some recent work to gain a feeling of the usage of security models. The results are shown in Table 1. Before that some explanations are needed for better understanding.

Table 1 Security Comparisons

Paper	Anonymity	Unforgeability	Unlinkability	Traceability	Resistance Collusion	D/S	Model
[3].1	Y	Y	Y	Y	Y	S	RM
[3].2	Y	Y	Y	Y	Y	S	RM
[3].3	Y	Y	Y	Y	Y	D	RM
[3].4	Y	Y	Y	Y	Y	D	RM
[23]	Y	-	-	-	Y	S	GM & RM
[21]	Y	Y	-	N	-	S	SM
[20]	Y	Y	Y	-	-	S	GM
[10]	Y	Y	-	Y	Y	D	RM
[8, 32]	Y	-	Y	-	Y	S	-
[22].1	Y	Y	-	N	-	S	RM
[22].2	Y	Y	-	N	-	S	SM
[24]	Y	Y	-	N	-	S	SM
[37]	Y	Y	Y	Y	Y	D	-

1. Four ABA schemes from [3] are analyzed, from chapters 5.3.3, 5.4.3, 5.5.3 and 6.4.1 respectively, and we denote them by [3].1, [3].2, [3].3 and [3].4 accordingly.
2. Two schemes are proposed in [22] and we denote them by [22].1 and [22].2 respectively.
3. "D" is dynamic and "S" is static.
4. "SM", "RM" and "GM" are short for standard model, random oracle model and generic model respectively.
5. "-" means it is not mentioned or analyzed in the referenced paper.

4.3. HIERARCHY

Hierarchy has not been discussed much in ABA schemes and the only work partly related we are aware of is the hierarchical structures of attribute-based encryption (ABE) proposed in [23, 25]. In both papers, the hierarchy is induced by a hierarchy of signers and Fig. 4 is one example to show its general structure. Instead of one central authority, authorities are divided into several layers according to trust levels. Usually, the higher level the authority is, the more trustworthy it is, where level 0 is the highest level of trust. For instance, the authorities of level 0 and level 3 in Fig. 4 have the most and the least trust respectively. There is a top authority (Level 0) which is considered as "trusted". It generates system parameters and is responsible for key generation for the next level authorities and so forth. All authorities except for the top authority can generate signers' attribute keys. In this way, signers are organized in a hierarchical structure.

This idea is similar to the hierarchy of group signatures in [26]. However, the main goal of [26] is to provide a hierarchical tracking ability structure. Since some ABA schemes are built based on group signatures and ring signatures, they have some similarities in the organization of hierarchies. First of all, there should be a group of signers in ABA, group or ring signature schemes, no matter the group is real or conceptual. Signers in a group or ring signature scheme generate signatures by a group-based private key. However, signers in ABA schemes sign by their own attribute private keys, but the verification is based on the group based public

key. This is how they can hide their identities in a signature in ABA, group or ring signature schemes. If tracking is required, there should be a group manager or an authority that has the identity information of all the signers in the group. Besides, identity information should be contained in the signature, so that it can be used by the opener to reveal the signer's identity. However, the identity information cannot be extracted by a verifier.

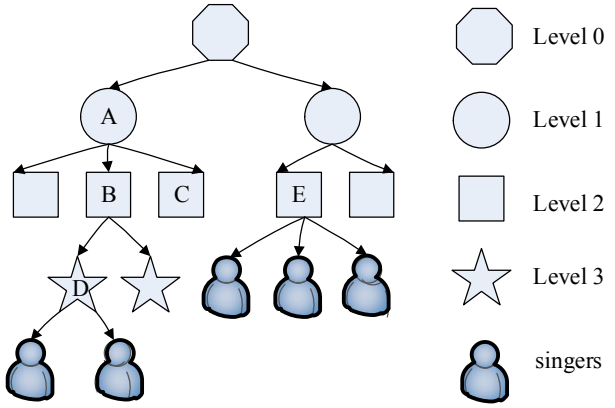

Fig. 4 – Signer-based Hierarchical Structure.

There are mainly three advantages for using hierarchical structures [23, 25]. First of all, authorities are structured in more fine-grained trust levels, which are closer to applications in real life. Secondly, the decentralized structure distributes the workload of the system so that a specific authority will not become the system bottleneck. Finally, the compromise of one non-top authority will not lead to the compromise of the whole system.

The hierarchy in [26] is user- or signer-based (Fig. 4). Except for signer-based hierarchical structures, hierarchy can also be attribute-based (Fig. 5), for which there is no published work to our best knowledge. In signer- and attribute-based hierarchical structures, a signer can belong to any authority except for the central one. However, in attribute-based hierarchical structures, attributes should be arranged strictly according to layers, so the chance that attributes are defined multiple times can be kept as low as possible.

From Fig. 3 and Fig. 5, we can see that there are mainly three disadvantages for signer-based hierarchy.

1. Signer-based hierarchy is not suitable for a system without a hierarchical structure of decentralized authorities.
2. In signer-based hierarchical ABA schemes, the definition and structure of attributes are not well organized, and there can be overlaps between different attribute structures. For example, authority D inherits from B, and both B and C inherit from A. It can happen that C and D define the same attributes differently. For a signer that belongs to these two different authorities, it has to describe the same object in two different ways.
3. If every lower-authority uses attribute structures defined by itself, it may be difficult to understand attribute requirements between each other. For instance, a signer belonging to authority D might have difficulty communicating with a signer belonging to authority E.

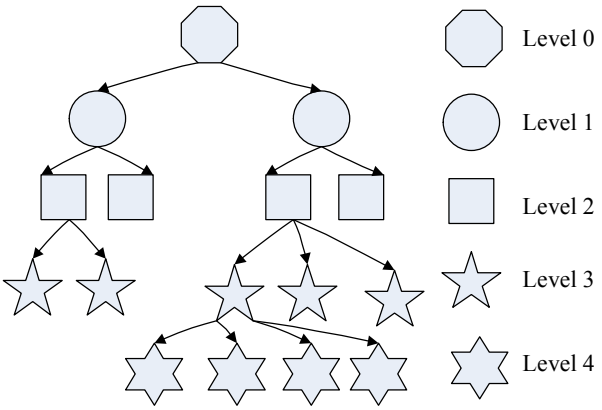

Fig. 5 – Attribute-based Hierarchical Structure.

Given the above reasons, if an ABA scheme without a hierarchical structure of signers wants to use hierarchy, it can be built based on the hierarchy of attributes. Compared with the signer-based hierarchical structure, the attribute-based hierarchical structure has the following two advantages.

1. The system has a unified definition of attributes, so it is easier for signers belonging to different domains to communicate with each other.
2. Attributes are well organized without overlaps, so the system can be less resource demanding.

So a system can use either scheme depending on different needs. However, researches on either topic are very few and there is no systematic method or instructions about how to build a hierarchical ABA scheme.

4.4. TRACEABILITY

The main purpose of ABA schemes is to achieve anonymity, which promises that the verifier cannot get any identity information about the signer during authentication. However, for some systems, tracking is of great importance. When disputes happen and the signer's identity is treated as legitimate evidence, tracking is useful. Because the property of

anonymity in ABA schemes, the signer itself cannot track the signer's identity, so it has to ask an authority (the "opener") to reveal the signer's identity. Whether the system has tracking ability or not greatly depends on the way how signatures are generated and what is included in the signature. For a traceable signature, it should have at least two parts, one for proving that the signer has the required attributes and one for identity tracking.

Based on whether the ABA schemes are traceable or not, we can divide most of the recent work into two parts. In general, ABA schemes based on group signatures are usually traceable. A typical example is the schemes proposed in [3], where some schemes are based on results from [17, 27]. If ABA schemes are based on ring signature [28], it is quite possible that they are not traceable, such as the scheme proposed in [21]. However, an ABA scheme with tracking ability usually has longer signature size compared with those untraceable systems. The method about how to achieve shorter or even constant size signature for a traceable ABA scheme is still under study.

4.5. REVOCATION

Revocation is an important feature of ABA systems. There has not been much work focused on ABA revocation, but it is well studied in group signatures, identity-based encryption (IBE) [29] and attribute-based encryption (ABE) [30, 31, 33].We will consider revocation methods that can be used in ABA schemes and discuss some of them.

Generally, there are two types of methods for revocation, a revocation list with revoked users or expiration time related attribute keys. The scheme in [17] uses a revocation list, which contains a token representing each revoked user. During system setup, each user should be registered in an authority's database and be known only to the authority. Once they are revoked, they can be found in the database and then added into the revocation list. For this method, the revocation list should be public and available all the time. Thus the biggest disadvantage for this method is that verifiers should have access to the Internet or at least to the server where the revocation list is stored.

The second method is to use expiration time. When signers' attribute keys are generated, they are usually combined with a time expiration date [20]. Compared with the first approach, verifiers do not have to have access to the revocation list, and what they have to do is to check whether the signature is generated by expired keys or not. There are two main drawbacks for this approach. The first one is that when the keys are expired, they need to be updated even though the users are still legitimate, which consumes extra system resources. The other drawback is that it is impossible to revoke either a certain user or attribute key before they become expired, unless the system has an extra revocation mechanism.

What has been discussed above is categorized by the methods of revocation. If we consider the coarse- or fine-grained revocations, then they can be divided into user-leveled [34] and attribute-leveled [33] revocation as stated in [20]. If the revocation is user-leveled, then once a user is revoked, all his or her attribute keys are revoked. However, if only an attribute of a user is revoked, the other attributes of the user are still usable. In an ABA system, the users are supposed to be dynamic, new members joining and old members leaving or being revoked. We have already discussed two methods based on revocation lists and expiration time, which can be either user- or attribute-leveled. Revocation list based mechanism cannot work in an off-line environment while expiration time mechanism is not flexible enough to revoke either a signer or an attribute at arbitrary point. There is still a need for methods that combine user-leveled with attribute-leveled revocation together to provide more efficiency and flexibility.

5. CONCLUSIONS

The main purpose of this paper is to review some work in ABA schemes, discuss open problems and inspire more researches in ABA scheme construction and designing. First of all, we introduce the main structure and workflow of ABA schemes to gain a big picture how ABA schemes work. Then we give an example how an ABA schemes can be cryptographically constructed, what security requirements it satisfies under what cryptographic assumptions and models, so that readers can get a clear view and more details about ABA schemes. Section 4 is the main part of this paper. We review and analyze recent work and approaches in some fields of ABA schemes first, including attribute trees building, cryptographic construction, security models, hierarchy, traceability and revocation. Based on these analyses, we discuss open problems in each field and hope that more and better researches can be inspired in ABA schemes.

6. REFERENCES

[1] V. C. Hu, D. Ferraiolo, R. Kuhn, A. Schnitzer, K. Sandlin, R. Miller, K. Scarfone, Guide to attribute based access control (ABAC) definition and considerations, available online on http://nvlpubs.nist.gov/nistpubs/special publications/NIST.sp.800-162.pdf, accessed June 2015.

[2] H. Li, Y. Dai, L. Tian, and H. Yang, Identity-based authentication for cloud computing, *Cloud Computing, Lecture Notes in Computer Science*, (5931) (2009), pp. 157-166.

[3] D. D. Khader, *Attribute-based Authentication Scheme*, PhD thesis, University of Bath, 2009.

[4] C. Schlger, M. Sojer, B. Muschall, and G. Pernul, Attribute-based authentication and authorisation infrastructuresfor e-commerce providers, *E-Commerce and Web Technologies*, *Lecture Notes in Computer Science*, (4082) (2006), pp. 132-141.

[5] J. Camenisch, M. Dubovitskaya, A. Lehmann, G. Neven, C. Paquin, and F.-S. Preiss, Concepts and languages for privacy-preserving attribute-based authentication, Policies and Research in Identity Management, *IFIP Advances in Information and Communication Technology*, (396) (2013), pp. 34-52.

[6] M. Covington, M. Sastry, and D. Manohar, Attribute-based authentication model for dynamic mobile environments, *Security in Pervasive Computing, Lecture Notes in Computer Science*, (3934) (2006), pp. 227-242.

[7] T. Priebe, W. Dobmeier, C. Schlger, and N. Kamprath, Supporting attribute-based access control in authorization and authentication infrastructures with ontologies, *Journal of Software*, (2) 1 (2007), pp. 27-38.

[8] L. Guo, C. Zhang, J. Sun, and Y. Fang, Paas: Aprivacy-preserving attribute-based authentication system for ehealth networks, in *Proceedings of the IEEE 32nd International Conference on Distributed Computing Systems (ICDCS)*, 2012, pp. 224-233.

[9] V. Goyal, O. Pandey, A. Sahai, and B. Waters, Attribute-based encryption for fine-grained access control of encrypted data, in *Proceedings of the 13th ACM Conference on Computer and Communications Security (CCS'06)*, 2006, pp. 89-98.

[10] K. Emura, A. Miyaji, and K. Omote, A dynamic attribute-based group signature scheme and its applicationin an anonymous survey for the collection of attribute statistics, in *Proceedings of International Conference on Availability, Reliability and Security (ARES'09)*, 2009, pp. 487-492.

[11] H. Cohen, G. Frey, R. Avanzi, C. Doche, T. Lange, K. Nguyen, and F. Vercauteren, *Handbook of Elliptic and Hyperelliptic Curve Cryptography*, 2nd Edition, Chapman & Hall/CRC, 2012.

[12] M. Armstrong, *Lagranges theorem in Groups and Symmetry, Undergraduate Texts in Mathematics*, Springer, New York. 1988, pp. 57-60.

[13] J. Camenisch and M. Stadler, Efficient group signature schemes for large groups, *Advances in Cryptology CRYPTO'97, Lecture Notes in Computer Science*, (1294) (1997), pp. 410-424.

[14] D. Chaum and E. van Heyst, Group signatures, Advances in Cryptology EUROCRYPT91, *Lecture Notes in Computer Science*, (547) (1991), pp. 257-265.

[15] G. Ateniese, J. Camenisch, S. Hohenberger, and B. de Medeiros, Practical group signatures without random oracles, *Cryptology ePrint Archive, Report 2005/385* (2005).

[16] M. Bellare, D. Micciancio, and B. Warinschi, Foundations of group signatures: Formal definitions, simplified requirements, and a construction based on general assumptions, Advances in Cryptology EUROCRYPT'2003, *Lecture Notes in Computer Science*, (2656) (2013), pp. 614-629.

[17] D. Boneh and H. Shacham, Group signatures with verifier-local revocation, in Proceedings of the 11th ACM Conference on Computer and Communications Security, CCS'04, 2004, pp. 168-177.

[18] H. Maji, M. Prabhakaran, and M. Rosulek, Attribute-based signatures, *Topics in Cryptology CT-RSA 2011, Lecture Notes in Computer Science*, (6558) (2011), pp. 376-392.

[19] N. Smart, *Cryptography: An Introduction*, Mcgraw-Hill College, USA, 2004.

[20] Y. Lian, L. Xu, and X. Huang, Attribute-based signatures with efficient revocation, in *Proceedings of 2013 5th International Conference on Intelligent Networking and Collaborative Systems (INCoS)*, 2013, pp. 573-577.

[21] W. Wenqiang, C. Shaozhen, Attribute-based ring signature scheme with constant-size signature, Information Security, IET, (4) 2 (2010), pp. 104-110.

[22] J. Li, M. H. Au, W. Susilo, D. Xie, and K. Ren, Attribute-based signature and its applications, in *Proceedings of the 5th ACM Symposium on Information, Computer and Communications Security, (ASIACCS'10)*, 2010, pp. 60-69.

[23] X. Liu, Y. Xia, S. Jiang, F. Xia, and Y. Wang, Hierarchical attribute-based access control with authentication for outsourced data in cloud computing, in *Proceedings of the 12th IEEE International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*, 2013, pp. 477-484.

[24] A.-J. Ge, C.-G. Ma, and Z.-F. Zhang, Attribute-based signature scheme with constant size signature in the standard model, *Information Security, IET*, (6) 2 (2012), pp. 47-54.

[25] Z. Wan, J. Liu, and R.-H. Deng, Hasbe: A hierarchical attribute-based solution for flexible and scalable access control in cloud computing, *IEEE Transactions on Information Forensics and Security*, (7) 2 (2012), pp. 743-754.

[26] M. Trolin and D. Wikstrm, Hierarchical group signatures, *Automata, Languages and Programming, Lecture Notes in Computer Science*, (3580) (2005), pp. 446-458.

[27] D. Boneh, X. Boyen, and H. Shacham, Shortgroup signatures, *Advances in Cryptology CRYPTO'2004, Lecture Notes in Computer Science*, (3152) (2004), pp. 41-55.

[28] R. Rivest, A. Shamir, and Y. Tauman, How to leaka secret: Theory and applications of ring signatures, *Theoretical Computer Science, Lecture Notes in Computer Science*, (3895) (2006), pp. 164-186.

[29] A. Boldyreva, V. Goyal, and V. Kumar, Identity-based encryption with efficient revocation, in *Proceedings of the 15th ACM Conference on Computer and Communications Security, (CCS'08)*, 2008, pp. 417-426.

[30] S. Yu, C. Wang, K. Ren, and W. Lou, Attribute based data sharing with attribute revocation, in *Proceedings of the 5th ACM Symposium on Information, Computer and Communications Security, (ASIACCS'10)*, 2010, pp. 261-270.

[31] J. Hur and D. K. Noh, Attribute-based access control with efficient revocation in data outsourcing systems, *IEEE Transactions on Parallel and Distributed Systems*, (22) 7 (2011), pp. 1214-1221.

[32] L. Guo, C. Zhang, J. Sun, andY. Fang, A privacy-preserving attribute-based authentication system for mobile health networks, *IEEE Transactions on Mobile Computing*, (13) (2014), pp. 1927-1941.

[33] N. Takeru, M. Masami, and S. Yoshiaki, Attribute-based encryption with attribute revocation and grant function using proxy re-encryption and attribute key for updating, *Future Information Technology, Lecture Notes in Computer Science*, (276) (2014), pp. 119-125.

[34] J. Ye, W. J. Zhang, S. L. Wu, Y. Y. Gao, and J. T. Qiu, Attribute-based fine-grained access control with user revocation, *Information and Communication Technology, Lecture Notes in Computer Science*, (8407) (2014), pp. 586-595.

[35] A. Hiroaki, A. Seiko, and S. Kouichi, Attribute-based signatures without pairings via the fiat-shamir paradigm, in *Proceedings of the 2nd ACM Workshop on ASIA Public-Key Cryptography (ASIAPKC'14)*, 2014, pp. 49-58.

[36] M. Abdalla, J. H. An, M. Bellare, and C. Namprempre, From identification to signatures via the Fiat-Shamir transform: minimizing assumptions for security and forward-security, in *EUROCRYPT2002, Lecture Notes in Computer Science*, (2332) (2002), pp. 418-433.

[37] H. H. Yang, and V. Oleshchuk, A dynamic attribute-based authentication scheme, *Codes, Cryptology, and Information Security, Lecture Notes in Computer Science*, (9084) (2015), pp. 106-118.

Huihui Yang *received the B.S degree in computer science from Wuhan University, Wuhan, China, in 2007 and M.S in information security from the Chinese Academy of Science, Beijing, China in 2011. From 2011 to 2012, she worked as a software design engineer in Ericsson (China) Communications Co. Ltd in Beijing. Since 2012, she has been a PhD student in information security in University of Agder, Grimstad, Norway. Her research interests include security and privacy, applied cryptography, protocol design and verification.*

Vladimir Oleshchuk *is Professor of Computer Science and Head of the Communication and System Security Group at the University of Agder, Norway. He received his MSc in Applied Mathematics (1981) and PhD in Computer Science (1988) from the Taras Shevchenko Kiev National University, Kiev, Ukraine, and his MSc in Innovations and Entrepreneurship from the Norwegian University of Science and Technology (NTNU). He has been working at the University of Agder since 1992. He has served as an editor and a member of program committees. He is a senior member of the IEEE and a senior member of the ACM. His current research interests include security, privacy and trust for wireless systems and their applications to e-health, wireless sensor networks, P2P systems, and mobile systems, application of formal methods to enforce security and privacy, security-related text analysis and privacy-preserving data analysis.*