Journal of Computer Virology and Hacking Techniques (2020) 16:333–341
https://doi.org/10.1007/s11416-020-00366-8

ORIGINAL PAPER

Advanced attribute-based encryption protocol based on the modified secret sharing scheme

M. A. Kudinov[1,2] · A. A. Chilikov[2,3] · E. O. Kiktenko[1] · A. K. Fedorov[1]

Received: 2 December 2019 / Accepted: 18 August 2020 / Published online: 28 August 2020

Abstract
We construct a new protocol for attribute-based encryption with the use of the modification of the standard secret sharing scheme. In the suggested modification of the secret sharing scheme, only one master key for each user is required that is achieved by linearly enlarging public parameters in access formula. We then use this scheme for designing an attribute-based encryption protocol related to some access structure in terms of attributes. We demonstrate that the universe of possible attributes does not affect the resulting efficiency of the scheme. The security proofs for both constructions are provided.

Keywords Secret sharing · Attribute-based encryption · Monotone access structures

1 Introduction

In the view of the significant increase in the amount of digital communications, the problem of efficient protection of data becomes crucial. An important task is to construct a secured protocol for controlled access to data. In standard protocols for solving this problem, which are mostly based on public-key cryptography, a secret key is required for access to whole encrypted data. A straightforward modifications of such protocols for providing partial access to data lead to a significant increase of the complexity since multiple encryptions of the same data are needed.

This work is supported by Russian Foundation for Basic Research (18-37-20033). A.A.C. is supported by Russian Science Foundation (17-11-01377).

✉ M. A. Kudinov
mishel-kudinov@mail.ru

A. A. Chilikov
chilikov@passware.com

E. O. Kiktenko
e.kiktenko@rqc.ru

A. K. Fedorov
akf@rqc.ru

[1] Russian Quantum Center, Skolkovo, Moscow, Russia 143025

[2] Bauman Moscow State Technical University, Moscow, Russia 105005

[3] Moscow Institute of Physics and Technology, Dolgoprudny, Moscow Region, Russia 141700

Attribute-based encryption (ABE) is a relatively new approach for solving the data access control problem [1–3]. In the ABE schemes, the access to the parts of an encrypted data is determined by a set of *attributes*, which are inherent to various participants. Thus, if attributes of a participant belonging to a particular subset of possible attributes, then he is able to obtain access to a corresponding particular part of the encrypted data. The ABE conception appears to be very promising in a framework of cloud technologies and distributed ledgers. Over the past decade, a number of modifications and improvements have been presented [1,4,5]. However, some of the proposed approaches still suffer from implementation complexity, which increases with the number of attributes.

We note that the concept of ABE has much in common with the secret sharing (SS) problem. However, one of the most common SS schemes [6] has a problem related to a large number of shares per trustee.

In this work, we propose an advanced ABE protocol with a sufficiently low computational complexity. One of the main techniques of our work is a modification of the standard SS scheme, which allows one to use a single key for generating the whole set of required shares. This modification is then used for the construction of the ABE protocol, which is independent to the size of the set of possible attributes.

The paper is organized as follows. In Sect. 2 we provide basic definitions. In Sect. 3 we briefly describe the standard construction of the general SS scheme. In Sect. 4 we present our modification of the SS scheme, which is then used for

constructing the ABE protocol in Sect. 5. In Sect. 6 we estimate the required resources for encryption and decryption for the suggested protocol. We summarize our results and conclude in Sect. 7.

2 Preliminaries

Let us introduce basic definitions and notations.

Let $x \leftarrow \mathcal{X}$, where x is a random value and $\mathcal{X}$ is a probability distribution, denote a sampling of x from the distribution $\mathcal{X}$. Let $y \leftarrow M(x)$, where M is an algorithm, denote the output y of M processed on the input x. Let $x \xleftarrow{\$} X$, where X is a set, denote an element x, which is chosen uniformly at random from the set X. Let $\vee(\phi_1, \ldots, \phi_n)$ and $\wedge(\phi_1, \ldots, \phi_n)$ stand for $\phi_1 \vee \ldots \vee \phi_n$ and $\phi_1 \wedge \ldots \wedge \phi_n$, correspondingly.

Now we define a pseudorandom function (PRF) family. Given the oracle f, we denote $M(f)$ as the execution of the oracle machine M with an access to f.

Definition 1 (*pseudorandom function (PRF) family*) We define $\mathbb{F}_{\mathcal{D}\to\mathcal{E}} = \{f_k : \mathcal{D} \to \mathcal{E}\}_{k\in\mathcal{K}}$, where $|\mathcal{K}| = |\mathcal{D}| = |\mathcal{E}| < \infty$ to be a function family. We define the advantage of an adversary $\mathcal{A}$ against PRF as

$$\mathrm{Adv}^{\mathrm{PRF}}_{\mathbb{F}_{\mathcal{D}\to\mathcal{E}}}(\mathcal{A}) = |\Pr[1 \leftarrow \mathcal{A}(f_k) : k \xleftarrow{\$} \mathcal{K}] - \Pr[1 \leftarrow \mathcal{A}(h) : h \xleftarrow{\$} H_{\mathcal{D}\to\mathcal{E}}]|,$$

where $H_{\mathcal{D}\to\mathcal{E}}$ is a family of all functions from $\mathcal{D} \to \mathcal{E}$ ($|H_{\mathcal{D}\to\mathcal{E}}| = |\mathcal{E}|^{|\mathcal{D}|}$). We define the PRF insecurity of a function family $\mathbb{F}_{\mathcal{D}\to\mathcal{E}}$ against time-ξ adversaries as the maximum advantage of any classical adversary that runs in time ξ :

$$\mathrm{InSec}^{\mathrm{PRF}}(\mathbb{F}_{\mathcal{D}\to\mathcal{E}}, \xi) = \max_{\mathcal{A}}\{\mathrm{Adv}^{\mathrm{PRF}}_{\mathbb{F}_{\mathcal{D}\to\mathcal{E}}}(\mathcal{A})\}$$

Definition 2 (*m-PRF family game*) We say that an oracle ω is initialized with a function $f(\cdot)$ if $\omega(x) = f(x)$, and denote it as $\omega \leftarrow f$. The following procedure is called m-PRF family game

Init: Given $\mathbb{F}_{\mathcal{D}\to\mathcal{E}} = \{f_k : \mathcal{D} \to \mathcal{E}\}_{k\in\mathcal{K}}$, where $|\mathcal{K}| = |\mathcal{D}| = |\mathcal{E}|$, flip a fair coin b. If $b = 1$ then $\Omega = \{\omega_i \leftarrow f_k : k \xleftarrow{\$} \mathcal{K},\ i \in \{1, \ldots, m\}\}$. Otherwise $\Omega = \{\omega_i \leftarrow h : h \xleftarrow{\$} H_{\mathcal{D}\to\mathcal{E}},\ i \in \{1, \ldots, m\}\}$, where $H_{\mathcal{D}\to\mathcal{E}}$ is a family of all functions from $\mathcal{D} \to \mathcal{E}$.

Game: Given a set of oracles Ω, the challenge is to distinguish whether Ω is initialized with functions from $F_{\mathcal{D}\to\mathcal{E}}$ or from $H_{\mathcal{D}\to\mathcal{E}}$

We define the advantage of an adversary $\mathcal{A}$ against m-PRF as

$$\mathrm{Adv}^{m-\mathrm{PRF}}_{\mathbb{F}_{\mathcal{D}\to\mathcal{E}}}(\mathcal{A}) = |\Pr[1 \leftarrow \mathcal{A}(\Omega)|b = 1] - \Pr[1 \leftarrow \mathcal{A}(\Omega)|b = 0]|.$$

We define the m-PRF insecurity of a function family $\mathbb{F}_{\mathcal{D}\to\mathcal{E}}$ against time-ξ adversaries as the maximum advantage of any classical adversary that runs in time ξ :

$$\mathrm{InSec}^{m-\mathrm{PRF}}(\mathbb{F}_{\mathcal{D}\to\mathcal{E}}, \xi) = \max_{\mathcal{A}}\{\mathrm{Adv}^{m-\mathrm{PRF}}_{\mathbb{F}_{\mathcal{D}\to\mathcal{E}}}(\mathcal{A})\}$$

Definition 3 (*Decisional Diffie Hellman (DDH) challenge* [7,8]) Consider a (multiplicative) cyclic group G of the order q with the generator g. We define the advantage of an adversary $\mathcal{A}$ against DDH as

$$\mathrm{Adv}^{\mathrm{DDH}}_{G}(\mathcal{A}) = |\Pr(1 \leftarrow \mathcal{A}(g^a, g^b, g^{ab}) - \Pr(1 \leftarrow \mathcal{A}(g^a, g^b, g^z))| \tag{1}$$

where a, b, z are chosen randomly and independently from $\mathbb{Z}_q$. We define the DDH insecurity of a group G against time-ξ adversaries as the maximum advantage of any classical adversary that runs in time ξ :

$$\mathrm{InSec}^{\mathrm{DDH}}(G, \xi) = \max_{\mathcal{A}}\{\mathrm{Adv}^{\mathrm{DDH}}_{G}(\mathcal{A})\}$$

Definition 4 (*m-DDH challenge*) Consider a (multiplicative) cyclic group G of the order q with the generator g, and following two distibutions:

- $\Omega_{ab} = \{(g^a, g^{b_1}, g^{a\cdot b_1}), (g^a, g^{b_2}, g^{a\cdot b_2}), \ldots, (g^a, g^{b_m}, g^{a\cdot b_m})\}$, where a and b_i are chosen randomly and independently from $\mathbb{Z}_q$ for $i = 1, \ldots, m$,
- $\Omega_z = \{(g^a, g^{b_1}, g^{z_1}), (g^a, g^{b_2}, g^{z_2}), \ldots, (g^a, g^{b_m}, g^{z_m})\}$, where a, b_i, z_i are chosen randomly and independently from $\mathbb{Z}_q$ for $i = 1, \ldots, m$,

We define the advantage of an adversary $\mathcal{A}$ against m-DDH as

$$\mathrm{Adv}^{m-\mathrm{DDH}}_{G}(\mathcal{A}) = |\Pr[1 \leftarrow \mathcal{A}(\Omega_{ab})] - \Pr[1 \leftarrow \mathcal{A}(\Omega_z)]| \tag{2}$$

We define the DDH insecurity of a group G against time-ξ adversaries as the maximum advantage of any classical adversary that runs in time ξ :

$$\mathrm{InSec}^{m-\mathrm{DDH}}(G, \xi) = \max_{\mathcal{A}}\{\mathrm{Adv}^{m-\mathrm{DDH}}_{G}(\mathcal{A})\}$$

Definition 5 Let $\mathcal{P} = \{P_1, \ldots P_n\}$ be a set. An access structure $\mathcal{B}$ is a collection of non-empty subsets of $\mathcal{P}$, i.e., $\mathcal{B} \subseteq 2^{\mathcal{P}}$.

Definition 6 Given a set $\mathcal{P}$, a monotone access structure on $\mathcal{P}$ is a collection of subsets $\mathcal{B} \subseteq 2^{\mathcal{P}}$ such that

$$B \in \mathcal{B}, B \subseteq B' \subseteq \mathcal{P} \Rightarrow B' \in \mathcal{B}.$$

Definition 7 A Boolean function $\Phi : \{0,1\}^n \to \{0,1\}$ is called monotone, if $\Phi(x_1, \ldots, x_n) \leq \Phi(x'_1, \ldots, x'_n)$, whenever for every $i \in \{1, \ldots, n\}$ $x_i \leq x'_i$.

Definition 8 Given an access structure $\mathcal{B}$, define a Boolean function $\Phi_{\mathcal{B}} : \{0,1\}^{|P|} \to \{0,1\}$ on $|\mathcal{P}|$-bit strings, where each bit is indexed by an element from $\mathcal{P}$, such that $\Phi(x) = 1$ iff $\{p : x_p = 1\} \in \mathcal{B}$.

One can look at the Boolean function $\Phi_{\mathcal{B}}$ as an indicator of the set $\mathcal{B}$. It is easy to check that the defined $\Phi_{\mathcal{B}}$ is a monotone Boolean function for a proper monotone access structure $\mathcal{B}$.

Definition 9 For a given set $\mathcal{P}$ and a monotone access structure $\mathcal{B}$ on $\mathcal{P}$, define $\mathcal{F}(\mathcal{B})$ to be the set of all Boolean formulae (expressions consisted of logical operations) on $|P|$ variables, such that for every formula $\phi \in \mathcal{F}(\mathcal{B})$ the output of ϕ is true iff the true variables in ϕ correspond exactly to a set $B \in \mathcal{B}$ (here we assume that each Boolean variable in the formula is indexed with an element form $\mathcal{P}$).

We note that $\phi, \phi' \in \mathcal{F}(\mathcal{B})$ implies that ϕ and ϕ' correspond to the same function $\Phi_{\mathcal{B}}$. They may, however, represent entirely different formula to express this function.

Definition 10 (*Random oracle* [9]) Random oracle is an oracle (a theoretical black box) that responds to every unique query with a value chosen uniformly at random from its output domain. If a query is repeated, it responds the same way every time that query is submitted. We refer a set of independent Random Oracles, $\{\mathrm{RO}_1, \ldots, \mathrm{RO}_t\}$, as a family of Random Oracles.

3 Standard construction of the general SS scheme

We begin our consideration with a SS scheme, which is proposed by Benaloh and Leichter [6], that we refer to as a *standard SS scheme*. For this purpose we first introduce a definition of the *secure generalized SS scheme*. It is known that for certain access structures every secure generalized SS scheme must be able to assign multiple shares to each trustee (see Theorem 2 below). In this case, we use $s_{p,j}$ to denote the jth share given to trustee p.

We define the scheme with the use of the following roles. We call the *dealer*, a user who shares a secret according to some access structure. The *trustees* are users among which the secret is shared. A *party* is a group of trustees. We denote the set of all trustees as $\mathcal{P}$.

Definition 11 (*Secure generalized SS scheme*) Given a monotone access structure $\mathcal{B}$ on a set of trusties $\mathcal{P}$ and a set of possible secrets $\mathcal{S}$, a secure generalized SS scheme for $\mathcal{B}$ is a method of dividing a secret $s \in \mathcal{S}$ into shares $\{s_{p,j}\}_{p \in \mathcal{P}, j \in \mathbb{N}}$ such that

- for every $B \in \mathcal{B}$, there is an algorithm for reconstructing the secret s from the subset of shares $\bigcup_{p \in B} \bigcup_{j} s_{p,j}$;
- for every $B \notin \mathcal{B}$ the subset of shares $\bigcup_{p \in B} \bigcup_{j} s_{p,j}$ provides no information (in an information theoretic sense) about the value of s.

In what is presented below, we define the secret domain $\mathcal{S} = \mathbb{Z}_q$, for some positive integer q. We then are able to construct the secure generalized SS scheme.

It is known that every monotone function Φ can represented with a formula ϕ consisted only of $\wedge$ and $\vee$ operations (without NOT operation). It is then sufficient to demonstrate how to divide a secret "across" these two operators. We use $X_{p,j}$ to denote the j^{th} appearance of variable $X_p : p \in P$ in a formula ϕ. We refer it as j-notation. For example, a formula $(X_1 \wedge X_2) \vee (X_1 \wedge X_3)$ transforms to $(X_{1,1} \wedge X_{2,1}) \vee (X_{1,2} \wedge X_{3,1})$

Let $\$(s, \phi)$ be a random function, which declares shares for each trustee $p \in P$ for $s \in \mathcal{S}$ and a monotone formula ϕ, that is defined as follows (we assume that ϕ is represented in j-notation):

- $\$(s', X_{p,j})$ assigns the share s' to trustee $p \in P$, such that $s_{p,j} := s'$;
- $\$(s', \vee(\phi_1, \ldots, \phi_n)) = \bigcup_{1 \leq i \leq n} \$(s, \phi_i)$;
- $\$(s, \wedge(\phi_1, \ldots, \phi_n)) = \bigcup_{1 \leq i \leq n} \$(s_i, \phi_i)$, where the s_i are chosen uniformly from $\mathcal{S}$, such that $s = (\sum_{i=1}^{n} s_i) \pmod q$.

It is then possible to show that for every monotone access structure $\mathcal{B}$, the SS scheme defined by $\$(s, \phi)$ satisfies the definition of a secure generalized SS scheme.

Theorem 1 *Let $\mathcal{B}$ be a monotone access structure on a set $\mathcal{P}$, $\phi \in \mathcal{F}(\mathcal{B})$ such that it is represented in j-notaition and contains only operators $\wedge$ and $\vee$, and let s be a secret from $\mathbb{Z}_q$. The SS scheme determined by $\$(s, \phi)$ is a secure generalized SS scheme for $\mathcal{B}$.*

Finally, we note that it is shown in [6] that there are access structures, which cannot be realized without giving multiple (or extra large) shares to some trustee.

Theorem 2 *There exist access structures for which any generalized SS scheme must give some trustee shares which are from a domain larger than that of the secret.*

See [6] for the proofs of Theorems 1 and 2.

4 Advanced SS scheme

4.1 General idea

As it is noted in [6], that we are unable to realize most monotone access structures with a standard SS scheme. However, one can modify the structures that can be realized efficiently, such that each trustee holds only one secret value, which we refer as a master key. With the use of the master key, a trustee is able to calculate all required shares.

We define the scheme with the use of the roles as defined above.

Let us begin with an illustrative example. Assume that a dealer wants to share a secret $s \in \mathbb{Z}_q$ between trustees Alice (A), Bob (B), Charlie (C), and David (D) according to the following access formula:

$$((X_{A,1} \wedge X_{B,1}) \vee (X_{B,2} \wedge X_{C,1}) \vee (X_{C,2} \wedge X_{D,1})), \quad (3)$$

where $X_{p,j}$ is a Boolean variable that represents a trustee p and appeared jth time in the formula. Let us introduce an address for each variable as its position in the formula as follows:

$$((\overset{0}{X}_{A,1} \wedge \overset{1}{X}_{B,1}) \vee (\overset{2}{X}_{B,2} \wedge \overset{3}{X}_{C,1}) \vee (\overset{4}{X}_{C,2} \wedge \overset{5}{X}_{D,1})) \quad (4)$$

Thus, $X_{A,1}.\text{address} = 0$, $X_{B,1}.\text{address} = 1$, $X_{B,2}.\text{address} = 2$, and so on.

To share a secret, the dealer first gives each trustee $p \in \{A, B, C, D\}$ a value mk_p, which is chosen uniformly at random from some domain $\mathcal{K}$. Next we refer to mk_p as a master key belonging to a trustee p.

Let us then assume that the dealer and trustees have access to independent random oracles family $\{\text{RO}_i : i \in \mathbb{Z}_q\}$ with an output domain in $\mathbb{Z}_q$. In order to generate a share that corresponds to a variable $X_{p,j}$, a trustee p has to query the random oracle RO_{mk_p} with $X_{p,j}.\text{address}$. For example, the shares for the defined access formula are computed in this way:

$$\begin{aligned}
s_{A,1} &= \text{RO}_{\text{mk}_A}(X_{A,1}.\text{address}) = \text{RO}_{\text{mk}_A}(0),\\
s_{B,1} &= \text{RO}_{\text{mk}_B}(X_{B,1}.\text{address}) = \text{RO}_{\text{mk}_B}(1),\\
s_{B,2} &= \text{RO}_{\text{mk}_B}(X_{B,2}.\text{address}) = \text{RO}_{\text{mk}_B}(2),\\
s_{C,1} &= \text{RO}_{\text{mk}_C}(X_{C,1}.\text{address}) = \text{RO}_{\text{mk}_C}(3),\\
s_{C,2} &= \text{RO}_{\text{mk}_C}(X_{C,2}.\text{address}) = \text{RO}_{\text{mk}_C}(4),\\
s_{D,1} &= \text{RO}_{\text{mk}_D}(X_{D,1}.\text{address}) = \text{RO}_{\text{mk}_D}(5). \quad (5)
\end{aligned}$$

Since each random oracle is independent, every share is a random value from $\mathbb{Z}_q$. As a result, a sum of shares, e.g. $s' := (s_{A,1} + s_{B,2}) (\text{mod } q)$, is also a uniformly random variable from $\mathbb{Z}_q$. To make it possible to reconstruct a secret s by trustees A and B, we add a publicly known value y_1 to this bracket, such that $(y_1 + s') (\text{mod } q) = s$.

Consequently, we modify our access formula into the following form:

$$((\overset{0}{X}_{A,1} \wedge \overset{1}{X}_{B,1} \wedge Y_1) \vee (\overset{2}{X}_{B,2} \wedge \overset{3}{X}_{C,3} \wedge Y_2) \vee (\overset{4}{X}_{C,2} \wedge \overset{5}{X}_{D,1} \wedge Y_3)), \quad (6)$$

where Y_i are Boolean variables that correspond to fictitious trustees, whose shares y_i are considered to be publicly known to every actual trustee. The value of y_i is computed in such a way that a reconstruction of secret becomes possible. We note that y_i is computed by the dealer, since he knows all the master keys.

Below we present our scheme in a more formal and efficient way.

4.2 Formal construction

Let n be a security parameter, $\mathbb{F}_q = \{f_k : k \in \mathcal{K}\}$ be a PRF family, where $q \geq 2^n$ and $f_k : \mathbb{Z}_q \to \mathbb{Z}_q$ with $|\mathcal{K}| = q$. Here we chose $f_k : \mathcal{D} \to \mathcal{E}$ with $\mathcal{D} = \mathbb{Z}_q$, but one can choose another domain. Note that $\mathcal{E} = \mathbb{Z}_q$, so we are able to sum the shares in $\mathbb{Z}_q$. Let H_q be a family of all functions $\mathbb{Z}_q \to \mathbb{Z}_q$. Let $l = \text{poly}(n)$ be the maximum size of monotone formula that we can use efficiently and let $l' := l/2$. Hereby the size of the monotone formula is the number of times that variables occur in the formula.

The roles for the scheme (*dealer*, *trustees*, and *party*) are defined in the previous subsection.

First, we define a modifying function $g_s(\phi)$, where ϕ is an access formula, whose size is less than $l' + 1$ and it is written in j-notation, and $s \in \mathbb{Z}_q$. Let $X_{p,j}$ be a variable, which represents a trustee p and it is appeared jth time in the formula. Let $X_{p,j}.\text{address}$ represents the position of the variable in ϕ. Let $\text{mk}_p \in \mathcal{K}$ be the value of p's master key. We denote Y_i as a variable that corresponds to a fictitious trustee and y_i as the value of his share. We use ϕ_i as subformula. Since every formula can be written in the following form:

$$\circ(\phi_1, \phi_2, \ldots, \phi_j, X_{p_1,k_1}, X_{p_2,k_2}, \ldots X_{p_t,k_t}), \quad (7)$$

where $\circ$ stands for either $\wedge$ or $\vee$.

Let is introduce a global counter α, which is initialized with 1. There are three separate cases to look at:

- $g_s(X_{p_1,k_1} \wedge \cdots \wedge X_{p_t,k_t}) = (X_{p_1,k_1} \wedge \cdots \wedge X_{p_t,k_t} \wedge Y_\alpha)$, where $t \geq 1$ and $y_i = s - f_{\text{mk}_{p_1}}(X_{p_1,k_1}.\text{address}) - \cdots - f_{\text{mk}_{p_t}}(X_{p_t,k_t}.\text{address}) (\text{mod } q)$, and the counter α is incremented $\alpha := \alpha + 1$.
- $g_s(X_{p_1,k_1} \wedge \cdots \wedge X_{p_t,k_t} \wedge \phi_1 \wedge \cdots \wedge \phi_j) = (X_{p_1,k_1} \wedge \cdots \wedge X_{p_t,k_t} \wedge g_{s_1}(\phi_1) \wedge \cdots \wedge g_{s_j}(\phi_j))$, where $j \geq 1$,

$\phi_i = \vee(\cdot)$ with at least one operator, $s_i \xleftarrow{\$} \mathbb{Z}_q$ for $i \in \{1, \ldots, j-1\}$ and $s_j := s - f_{\mathrm{mk}_{p_1}}(X_{p_1,k_1}.\mathrm{address}) - \ldots - f_{\mathrm{mk}_{p_t}}(X_{p_t,k_t}.\mathrm{address}) - s_1 - \cdots - s_{j-1} \pmod q$.

– $g_s(X_{p_1,k_1} \vee \cdots \vee X_{p_t,k_t} \vee \phi_1 \vee \cdots \vee \phi_j) = (g_s(X_{p_1,k_1}) \vee \cdots \vee g_s(X_{p_t,k_t}) \vee g_s(\phi_1) \vee g_s(\phi_2) \vee \cdots \vee g_s(\phi_j))$.

Let us clarify that the address of a variable is the number of the position of that variable in the formula ϕ (conventionally, we count from left to right).

Now we can describe our *advanced SS scheme*. To share a secret the dealer should follow these steps:

1. Choose a secret $s \xleftarrow{\$} \mathbb{Z}_q$.
2. Choose a master key for each trustee in the union $\mathcal{P}$ uniformly at random from $\mathcal{K}$ (for each $p \in \mathcal{P} : \mathrm{mk}_p \xleftarrow{\$} \mathcal{K}$).
3. Choose a monotone formula ϕ of size less or equal to l', which represents an access structure.
4. Evaluate $\phi' = g_s(\phi)$.
5. Publish ϕ', so that the values y_i are available for everyone.

To reconstruct a secret a party should follow these steps:

1. Each trustee p in the party has to evaluate their shares: $s_{p,j} = f_{\mathrm{mk}_p}(X_{p,j}.\mathrm{address})$.
2. Using the corresponding shares and public values y_i, a verified party can calculate the secret s according to the way it is shared.

Definition 12 Given a set $\mathcal{P}$ and a monotone access structure $\mathcal{B}$ on $\mathcal{P}$, an *advanced SS scheme* for $\mathcal{B}$ is a method of dividing a secret s into shares $s_{p,j}$ such that the following statements hold true:

– When $B \in \mathcal{B}$, the secret s can be reconstructed from the shares $\bigcup_{p \in B} \bigcup_j s_{p,j}$ and public values $y_1, \ldots, y_t$.
– When $B \notin \mathcal{B}$, the secret s can be reconstructed only with a negligible probability from the shares $\bigcup_{p \in B} \bigcup_j s_{p,j}$ and public values $y_1, \ldots, y_t$.

4.3 Security proof

Here we introduce a notion of the security model that is used for our scheme, which is similar to the Selective-Id model [10–12].

Definition 13 (*Selective-Id model for advanced SS scheme*) The following procedure is called Selective-Id model for advanced SS scheme.

Init: The adversary chooses an access structure $\mathcal{B}$ with a corresponding formula ϕ and gives it to the challenger.
Phase 1: The adversary declares the set of trustees γ, which does not satisfy the formula ϕ and obtains master keys of trustees from γ from the challenger.
Challenge: The adversary submits two secrets s_0 and s_1. The challenger flips a fair coin b and shares the secret s_b.
Phase 2: The challenger gives to the adversary $\phi' = g_{s_b}(\phi)$ and corresponding values $y_1, \ldots, y_j$.
Guess: The adversary outputs a guess b' of b.

The advantage of an adversary in this game is defined as $|\Pr[b' = b] - \frac{1}{2}|$.

Theorem 3 *Consider the advanced SS scheme for a set of parties $\mathcal{P}$ based on PRF family $\mathbb{F}_q = \{f_k : \mathbb{Z}_q \to \mathbb{Z}_q\}_{k \in \mathcal{K}}$ with $|\mathcal{K}| = q$. The advantage ε' in the Selective-Id model of any classical adversary $\mathcal{A}$ that runs in time ξ' satisfies the inequality $\varepsilon' \leq \mathrm{InSec}^{\mathrm{PRF}}(\mathbb{F}_q, \xi) \cdot |\mathcal{P}|$, where $\xi' \approx \xi$ assuming that time needed for sampling no more than $|\mathcal{P}| + l'$ random variables is negligible, where l' is the maximum size of the formula which can be efficiently processed by the advanced SS scheme.*

Proof First of all, one can easily notice that the reconstruction of the secret happens the same way as in the standard SS scheme. We also note that if $B \notin \mathcal{B}$ (i.e. B does not satisfy the formula ϕ), then $B \cup (\bigcup_i Y_i)$ does not satisfy the access structure defined by $\phi' = g_s(\phi)$ as $X_{p,j} \wedge 1 = X_{p,j}$.

Consider, a modification of the advanced SS scheme (modified advanced SS scheme), where PRF family $\mathbb{F}_q$ is replaced with a set of random oracles. One can see that this scheme is exactly the standard SS scheme based on formula $\phi' = g_s(\phi)$. So there is no chance for an adversary to compute the secret, which possesses the shares from $B \notin \mathcal{B}$.

Now suppose that there exists a probabilistic polynomial time adversary $\mathcal{A}$, which has an advantage ε' in Selective-Id model for advanced SS scheme. Without loss of generality, we assume that it's probability of guessing a correct value is $\Pr[b' = b] = 1/2 + \varepsilon'$. Then we show that it is possible to distinguish the PRF family $\mathbb{F}_q$ from truly random function family with a probability at least $\varepsilon'/|\mathcal{P}|$. To show this we construct an oracle machine $\mathcal{M}^{\mathcal{A}}$ that has an advantage ε' in $|\mathcal{P}|$-pseudorandom function family game (see Algorithm 1). Let us calculate the probabilities to obtain $v' = 0$ and $v' = 1$ (v' is defined in Algorithm 1).

Suppose that the challenge Ω is initialised with functions from the family $\mathbb{F}_q$. In this case, the situation for the adversary is completely the same as in the case of the (non-modified) advanced SS scheme. Therefore, the adversary

Algorithm 1: $\mathcal{M}^{\mathcal{A}}$

Input : Security parameter n, function family $\mathbb{F}_q$, $|\mathcal{P}|$-pseudorandom function family challenge $\Omega = \{\omega_{p_1}, \ldots, \omega_{p_N}\}$, where $\{p_1, \ldots, p_N\} = \mathcal{P}$.
Output: A guess v'.
The adversary $\mathcal{A}$ declares an access structure, a corresponding formula ϕ, and a set of trustees γ, which does not satisfy the formula ϕ.
$\mathcal{A}$ queries the master keys of trustees from γ.
Generate a master key uniformly at random for each trustee in γ and response to the adversary with those keys.
The adversary submits two secrets s_0 and s_1.
Flip a fair coin b and share the secret s_b according to the *advanced SS scheme*, but instead of generating master keys for trustees in $\mathcal{P} \setminus \gamma$ and calculating the shares with $f_k \in \mathbb{F}_q$, use an oracle ω_p from Ω for trustee $p \in \mathcal{P} \setminus \gamma$ and calculate the shares as $s_{p.j} = \omega_p(X_{p.j}.\text{address})$. We call this modification $g'_s(\phi)$.
Give to the adversary $\phi' = g'_{s_b}(\phi)$ and corresponding values $y_1, \ldots, y_j$.
The adversary outputs a guess b' of b.
if $b' = b$ **then**
  return $v' = 1$
else
  return $v' = 0$

correctly guesses the value b' with an advantage ε' or what is the same with probability $\frac{1}{2} + \varepsilon'$.

If the challenge Ω is initialized with functions from the family H_q, then the shares of the trustees from $\mathcal{P} \setminus \gamma$ are chosen uniformly at random. And the situation is the same as in the *standard SS scheme*. Since γ does not satisfy the formula, it is required to obtain at least one more share to get the secret, but all the remaining shares are chosen uniformly at random. Therefore, according to the Theorem 1 the adversary has no information about the secret in this situation. Thus, in this case the adversary can only randomly guess the value b, so b' is correctly guessed with a probability $\frac{1}{2}$.

Let $v = 0$ corresponds to the challenge Ω initialized with functions from the family H_q and $v = 1$ to the challenge Ω initialized with functions from the pseudorandom function family. Then the overall advantage in the $|\mathcal{P}|$-pseudorandom game is $|\Pr[v' = 1 | v = 0] - \Pr[v' = 1 | v = 1]| = |\frac{1}{2} - (\frac{1}{2} + \varepsilon')| = \varepsilon'$.

By the hybrid argument [13] we can distinguish a pseudorandom function family with probability $\varepsilon'/(|\mathcal{P}|)$. In order to apply the hybrid argument consider two distributions,

$$D_1 = \{D_{1.i} = f_k : k \xleftarrow{\$} \mathcal{K},\ 1 \le i \le |\mathcal{P}|,\ f_k \in \mathbb{F}_q\}, \tag{8}$$

and

$$D_2 = \{D_{2.i} = h \xleftarrow{\$} H_q : 1 \le i \le |\mathcal{P}|\}. \tag{9}$$

Define a sequence of hybrid distributions $D_1 = T_0,\ T_1, \ldots, T_{|\mathcal{P}|} = D_2$, where $T_i = \{T_{i.j} = h \xleftarrow{\$} H_q : 1 \le j \le i\} \cup \{T_{i.j} = f_k : k \xleftarrow{\$} \mathcal{K}, i < j \le |\mathcal{P}|, f_k \in \mathbb{F}_q\}$. So we have $\text{Adv}_{D_1, D_2}(\mathcal{M}^{\mathcal{A}}) = \varepsilon'$. Let us remind that

$$\text{Adv}_{T_i, T_{i+1}}(\mathcal{M}^{\mathcal{A}}) = |\Pr[x \xleftarrow{\$} T_i : \mathcal{M}^{\mathcal{A}}(x) = 1] - \\ - \Pr[x \xleftarrow{\$} T_{i+1} : \mathcal{M}^{\mathcal{A}}(x) = 1]| \tag{10}$$

By the triangle inequality, it is clear that

$$\text{Adv}_{D_1, D_2}(\mathcal{M}) \le \sum_{i=0}^{|\mathcal{P}|-1} \text{Adv}_{T_i, T_{i+1}}(\mathcal{M})$$

Thus, there exists some η, such that $0 \le \eta < |\mathcal{P}|$ and

$$\text{Adv}_{T_\eta, T_{\eta+1}}(\mathcal{M}) \ge \text{Adv}_{D_1, D_2}(\mathcal{M})/|\mathcal{P}| = \varepsilon'/|\mathcal{P}|. \tag{11}$$

Suppose that we have a sample $\omega \xleftarrow{\$} \mathbb{F}_q$ or $\omega \xleftarrow{\$} H_q$. Let us construct a distribution $T' = \{T_i \xleftarrow{\$} H_q : 1 \le i \le \eta\} \cup \{T_{\eta+1} = \omega\} \cup \{T_i \leftarrow f_k : k \xleftarrow{\$} \mathcal{K},\ \eta + 1 < i \le |\mathcal{P}|,\ f_k \in \mathbb{F}_q\}$. If $\omega \xleftarrow{\$} \mathbb{F}_q$ then T' is distributed the same as T_η, otherwise it is distributed as $T_{\eta+1}$. Thus, we can distinguish samples from $\mathbb{F}_q$ and H_q with probability $\varepsilon'/|\mathcal{P}|$.

Finally, we obtain: $\varepsilon' \le \text{InSec}^{\text{PRF}}(\mathbb{F}_q, \xi) \cdot |\mathcal{P}|$, where ξ is a total time of running $\mathcal{M}^{\mathcal{A}}$ plus initialization of an appropriate hybrid. Neglegting the time needed for preparing data for $\mathcal{A}$ and the hybrid T' we obtain $\xi \approx \xi'$. □

5 Advanced ABE scheme

5.1 Formal construction

Consider a group of users, where each user posses a list of attributes. Let $\mathcal{P}$ be a set of all existing attributes. Let us call a community a subgroup of users, who possess a particular attribute $p \in \mathcal{P}$. In what follows, we refer to the community p as a subgroup of users that possess an attribute p. We note that a user can belong to several communities if he has more than one attribute.

Let $n \in \mathbb{N}$ be a security parameter, G be a multiplicative group of a prime order q, where $2^n < q < 2^{n+1}$ in which DDH assumption is considered to be true, g is a generator of that group, H_q is a family of all functions $\mathbb{Z}_q \to \mathbb{Z}_q$, and $\mathbb{F}_q = \{f_k : \mathbb{Z}_q \to \mathbb{Z}_q\}_{k \in G}$ is a PRF family. We construct the advanced ABE scheme based on the advanced SS scheme in the following form.

Setup: Each community p in the universe $\mathcal{P}$ generates their secret key $\text{sk}_p \xleftarrow{\$} \mathbb{Z}_q$ and a correspong public key pk =

g^{sk_p}. Then the public key is shared among the whole group of users. So that the set of public keys $PK = \{\mathrm{pk}_p = g^{\mathrm{sk}_p} : p \in \mathcal{P}\}$ is assumed to be known to every user in the group.

Encryption $(M, PK, \phi, \mathbb{F}_q)$: To encrypt a message $M \in \mathbb{Z}_q$ under public keys PK and formula ϕ, which represents some monotone access structure, one generates $s \xleftarrow{\$} \mathbb{Z}_q, e \xleftarrow{\$} \mathbb{Z}_q$ and computes the ciphertext in the following form $E = \{E' = M + s(\mathrm{mod}\, q), g^e, \phi' = g_s(\phi), y_1, \ldots, y_t\}$, where $g_s(\cdot), y_1, \ldots, y_t$ come from the advanced SS scheme based on PRF family $\mathbb{F}_q$ and the corresponding master keys are calculated as $\mathrm{mk}_p = g^{\mathrm{sk}_p \cdot e}$.

Decryption $(E, SK, Attr, \mathbb{F}_q)$:, where SK is a set of all secret keys known to a concrete user and $Attr$ is a set of attributes he posseses. For each $\mathrm{sk}_p \in SK$, a user calculates the master key $\mathrm{mk}_p = (g^e)^{\mathrm{sk}_p}$. Then if $Attr$ satisfies the access structure, then the secret s can be reconstructed using $MK = \{\mathrm{mk}_p\}$, ϕ' and $y_1, \ldots, y_t$. The message is obtained from E' as $M = E' - s(\mathrm{mod}\, q)$.

5.2 Security proof

In order to provide a formal security analysis of the advanced ABE scheme, we introduce the following definition.

Definition 14 (*Attribute-based Selective-Set model*) The following procedure is called attribute-based Selective-Set model:

Init: The adversary chooses an access structure and a corresponding formula ϕ and sends ϕ to the challenger.

Phase 1: The adversary declares the set of communities γ, which does not satisfy the formula ϕ and obtains secret keys of communities from γ from the challenger.

Challenge: The adversary submits two secrets s_0 and s_1. The challenger flips a fair coin b and encrypts $m \xleftarrow{\$} \mathbb{Z}_q$: $E' = m + s_b(\mathrm{mod}\, q)$.

Phase 2: The challenger gives to the adversary public keys of all communities and E, which is a ciphertext of m generated according to the advanced ABE scheme.

Guess: The adversary outputs a guess b' of b.

The advantage of an adversary in this game is defined as $|\Pr[b' = b] - \frac{1}{2}|$.

Below we prove that the security of our scheme in the attribute-based Selective-Set model reduces to the hardness of the DDH challenge and pseudorandomness of the function family.

Theorem 4 *Consider the advanced ABE scheme based on an PRF family $\mathbb{F}_q$ and set of communities $\mathcal{P}$. The advantage ε' in the the Attribute-based Selective-Set model game of any classical adversary $\mathcal{A}$ that runs in time ξ' satisfies the following inequality: $\varepsilon' \leq \mathrm{InSec}^{\mathrm{DDH}}(G, \xi) \cdot |\mathcal{P}| + \mathrm{InSec}^{\mathrm{PRF}}(\mathbb{F}_q, Q_s) \cdot |\mathcal{P}|$. With $\xi \approx \xi' \approx \tilde{\xi}$ assuming that time required for sampling no more than $3|\mathcal{P}| + l'$ random variables is negligible, where l' is the maximum size of the formula which can be efficiently processed by the advanced ABE scheme.*

Proof First, suppose that the master keys are replaced with uniformly random keys. In this case, let us denote the advantage in breaking the modified advanced ABE protocol in the attribute-based Selective-Set model as $\widetilde{\varepsilon}$. If $(\varepsilon' - \widetilde{\varepsilon})$ is not negligible, then we can construct a machine that breaks $|\mathcal{P}|$-DDH challenge with an advantage of at least $(\varepsilon' - \widetilde{\varepsilon})$.

We assume that $\widetilde{\varepsilon} < \varepsilon'$, since we limit the value of $\widetilde{\varepsilon}$ by the pseudoradnomnes property and if ε' is less than $\widetilde{\varepsilon}$ then we can limit them both.

Let us denote a $|\mathcal{P}|$-DDH challenge $\Omega = \{w_{p_1}, \ldots, w_{p_N}\}$, where $N = |\mathcal{P}|$, $\{p_1, \ldots, p_N\} = \mathcal{P}$ and w_{p_i} is a tuple either $(g^a, g^{b_i}, g^{a \cdot b_i})$ or (g^a, g^{b_i}, g^{z_i}). We use $w_{i.j}$ to denote the jth element of the tuple. To prove the theorem, consider the following algorithm.

If $\omega_{p.3}$ is sampled uniformly at random ($v = 0$), then the master keys are chosen uniformly at random. Hence the adversary has no information about the master keys he did not query. Remind that we denote the advantage of the adversary in this situation as $\widetilde{\varepsilon}$. Otherwise ($v = 1$) the situation is the same as in the original ABE protocol. Thus, we have the overall advantage in the $|\mathcal{P}|$-DDH game as follows:

$$\begin{aligned} \mathrm{InSec}^{|\mathcal{P}|-\mathrm{DDH}}(G, \xi_{\mathcal{P}}) &\geq |\Pr[v' = 1 | v = 0] - \Pr[v' = 1 | v = 1]| \\ &= |(\frac{1}{2} + \widetilde{\varepsilon}) - (\frac{1}{2} + \varepsilon')| = \varepsilon' - \widetilde{\varepsilon}, \end{aligned} \tag{12}$$

where $\xi_{\mathcal{P}}$ is a running time of Algorithm 2. Neglegting the time for preparing data for $\mathcal{A}$ we obtain $\xi_{\mathcal{P}} \approx \xi'$.

Algorithm 2: $\mathcal{M}^{\mathcal{A}}$

Input : Security parameter n, $|\mathcal{P}|$-DDH challenge Ω.
Output: A guess v'.
The adversary $\mathcal{A}$ chooses an access structure and a corresponding formula ϕ and sends it to the challenger.
$\mathcal{A}$ declares the set of communities γ, which does not satisfy the formula ϕ, whose secret keys he wishes to get and queries them.
Generate a secret key for each community in γ and response to the adversary with those keys.
The adversary submits two secrets s_0 and s_1.

Flip a fair coin b and encrypt a message $m \xleftarrow{\$} \mathbb{Z}_q$ according to the advanced ABE scheme with $s = s_b$, but instead of secret keys for communities in $\mathcal{P} \setminus \gamma$ use sample ω_p from Ω for community $p \in \mathcal{P} \setminus \gamma$. Take $\omega_{p.2}$ as his public key and $\omega_{p.3}$ as his master key. We call this modification $g'_s(F)$.
Give to the adversary
$E = \{E' = m + s(\text{mod } q), \omega_{1,1}, \phi' = g'_s(\phi), y_1, \ldots, y_j\}$.
The adversary outputs a guess b' of b.
if $b' = b$ **then**
  return $v' = 1$
else
  return $v' = 0$

In analogy to the proof of Theorem 3, one can see that due to the hybrid argument

$$\text{InSec}^{\text{DDH}}(G, \xi) \geq (\varepsilon' - \widetilde{\varepsilon})/|\mathcal{P}|,$$

where $\xi \approx \xi_{\mathcal{P}}$ neglegting the time, needed to prepare an appropriate hybrid.

Finally, we limit the value of $\widetilde{\varepsilon}$. Due to the fact the master keys are chosen uniformly at random, the security of such a scheme reduces to the security of the advanced SS scheme straightforwardly. Therefore, according to Theorem 3: $\widetilde{\varepsilon} \leq \text{InSec}^{\text{PRF}}(\mathbb{F}_{\text{q}}, \tilde{\xi}) \cdot |\mathcal{P}|$, with $\tilde{\xi} \approx \xi_{\mathcal{P}}$.

Thus, we arrive to the final result:

$$\varepsilon' \leq \text{InSec}^{\text{DDH}}(G, \xi) \cdot |\mathcal{P}| + \text{InSec}^{\text{PRF}}(\mathbb{F}_{\text{q}}, \tilde{\xi}) \cdot |\mathcal{P}|,$$

with $\xi' \approx \xi \approx \tilde{\xi}$. □

6 Efficiency estimation for advanced ABE scheme

Here we analyze the efficiency of the proposed advanced ABE scheme in terms of sizes of ciphertext, public parameters, and private key, and the computation time for decryption and encryption.

Consider a ciphertext $E = \{E' = m + s(\text{mod } q), g^e, \phi' = g_s(\phi), y_1, \ldots, y_j\}$ and a plaintext $PT = \{m, \phi\}$. We note that it is required to publish the rules of the access structure, hence we assume that the plaintext is accomplished by the formula ϕ. One can see that ϕ' is no more than twice bigger than ϕ. This is due to the fact that the number of additional Boolean variables corresponded to fictitious trustees (communities) can not exceed the number of Boolean variables corresponded to the actual trustees (communities). We then note that ϕ and ϕ' make a major contribution into the size of PT and E. Hence, the overhead of the ciphertext compared to plaintext is of the size linear in the size of the formula ϕ.

The public parameters of the system are of size linear in the number of existing attributes. The private key of the community consists of a single value from $\mathbb{Z}_q$.

The encryption procedure generates two random values, performs one addition in $\mathbb{Z}_q$ and one exponentiation in the group G, l calls to functions from $\mathbb{F}_q$, where l denotes the size of the formula ϕ. The modification of the formula ϕ into ϕ' is performed in the linear time with the usage of syntax tree.

Thus, the amount of the communities in the scheme is $|\mathcal{P}|$. The decryption procedure needs to perform at most $|\mathcal{P}|$ exponentiations, l' sums and pseudorandom function calls, where l' is the size of formula ϕ'. Finally, one subtraction is required.

7 Conclusion

Here we summarize the main results of our work. First, we have presented the modification of the SS scheme, which allows a user to store only one value to calculate the corresponding shares. Based on this modification, we have proposed the advanced ABE protocol. We have provided the security and efficiency analysis of the proposed scheme.

One of the most significant impacts of this paper is rejection of bilinear mappings, which evidently increases the efficiency of the proposed scheme and allows to dimamicaly add new attributes.

One can see that the proposed ABE scheme is not collusion resistant as well as some other ABE schemes (e.g. see [14]). We note, that all known collusion resistant schemes are based on using of trusted centers which are absent in our scheme.

There are several ways to improve the proposed scheme. The first one is based on adding new logical elements, e.g. threshold, so that the formula ϕ can be constructed more efficiently. The second question is related to modification of this protocol with respect to the use of other key exchange schemes.

References

1. Goyal, V., Pandey, O., Sahai, A., Waters B.R.: Attribute-based encryption for fine-grained access control of encrypted data. In: Proceedings of the ACM Conference on Computer and Communications Security, pp. 89–98 (2006)

2. Boneh, D., Franklin, M.K.: Identity-based encryption from the Weil pairing. In: Proceedings of the 21st Annual International Cryptology Conference on Advances in Cryptology, pp. 213–229 (2001)
3. Boneh, D., Gentry, C., Waters, B.: Collusion resistant broadcast encryption with short ciphertexts and private keys. In: Advances in Cryptology—CRYPTO, Lecture Notes in Computer Science, vol. 3621, pp. 258–275 (2005)
4. Sahai, A., Waters, B.: Fuzzy identity-based encryption. In: Advances in Cryptology EUROCRYPT, vol. 2005, pp. 457–473 (2005)
5. Abdalla, M., Catalano, D., Dent, A.W., Malone-Lee, J., Neven, G., Smart, N.P.: Identity-based encryption gone wild. In: Bugliesi, M., Preneel, B., Sassone, V., Wegener, I. (eds.) ICALP (2). Lecture Notes in Computer Science, vol. 4052, pp. 300–311. Springer, Berlin, Heidelberg (2006)
6. Benaloh, J., Leichter, J.: Generalized secret sharing and monotone functions. In: Goldwasser, S. (ed.) Advances in Cryptology CRYPTO 88. Lecture Notes in Computer Science, vol. 403, pp. 27–35. Springer, New York, NY (1990)
7. Boneh, D.: The decision Diffie–Hellman problem. In: Buhler, J.P. (ed.) Proceedings of the Third Algorithmic Number Theory Symposium, Lecture Notes in Computer Science, vol. 1423, pp. 48–63. Springer, Berlin (1998)
8. Cramer, R., Damgrd, I., Kiltz, E., Zakarias, S., Zottarel, A.: DDH-like assumptions based on extension rings. In: Public Key Cryptography-PKC 2012, Lecture Notes in Computer Science, vol. 7293, pp. 644–661 (2012)
9. Bellare, M., Rogaway, P.: Random oracles are practical: A paradigm for designing efficient protocols. In: ACM Conference on Computer and Communications Security (ACM CCS), pp. 62–73 (1993)
10. Canetti, R., Halevi, S., Katz, J.: A forward-secure public-key encryption scheme. In: Advances in Cryptology—Eurocrypt, Lecture Notes in Computer Science, vol. 2656, pp. 255–271 (2003)
11. Canetti, R., Halevi, S., Katz, J.: Chosen ciphertext security from identity based encryption. In: Advances in Cryptology-Eurocrypt, Lecture Notes in Computer Science, vol. 3027, pp. 207–222 (2004)
12. Boneh, D., Boyen, X.: Efficient selective-ID secure identity based encryption without random oracles. In: Cachin, C., Camenisch, J.L. (eds.) Advances in Cryptology—EUROCRYPT 2004. EUROCRYPT 2004. Lecture Notes in Computer Science, vol. 3027, pp. 223–238. Springer, Berlin (2004)
13. Goldreich, O.: Foundations of Cryptography: Volume 1 Basic Tools. Cambridge University Press, Cambridge (2001)
14. Kapadia, A., Tsang, P., Smith, S.: Attribute-Based Publishing with Hidden Credentials and Hidden Policies (2007)