Security Analysis of a Decentralized, Revocable and Verifiable Attribute-Based Encryption Scheme

Thomas Prantl, Marco Lauer, Lukas Horn, Simon Engel, David Dingel, Samuel Kounev
Julius-Maximilians-Universität Würzburg
Würzburg, Bavaria, Germany

André Bauer
University of Chicago
Chicago, Illinois , USA

Christian Krupitzer
University of Hohenheim
Stuttgart, Baden-Wuerttemberg
Germany

ABSTRACT

In recent years, digital services have experienced significant growth, exemplified by platforms like Netflix achieving unprecedented revenue levels. Some of these services employ subscription models, with certain content requiring additional payments or offering third-party products. To ensure the widespread availability of diverse digital services anytime and anywhere, providers must have control over content accessibility. To address the multifaceted challenges in this domain, one promising solution is the adoption of attribute-based encryption (ABE). Over the years, various approaches have been proposed in the literature, offering a wide range of features. In a prior study [18], we assessed the security of one of these proposed approaches and identified one that did not meet its promised security standards. In this research we focuses on conducting a security analysis for another ABE scheme to pinpoint its shortcomings and emphasize the critical importance of evaluating the safety and effectiveness of newly proposed schemes. Specifically, we uncover an attack vector within this ABE scheme, which enables malicious users to decrypt content without the required permissions or attributes. Furthermore, we propose a solution to rectify this identified vulnerability.

CCS CONCEPTS

• **Security and privacy** → **Key management**; **Public key encryption**; **Access control**; **Digital rights management**; • **Applied computing**;

KEYWORDS

Attribute-based Encryption, Public key encryption, Access control

ACM Reference Format:
Thomas Prantl, Marco Lauer, Lukas Horn, Simon Engel, David Dingel, Samuel Kounev, André Bauer, and Christian Krupitzer. 2024. Security Analysis of a Decentralized, Revocable and Verifiable Attribute-Based Encryption Scheme. In *The 19th International Conference on Availability, Reliability and Security (ARES 2024), July 30–August 02, 2024, Vienna, Austria.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3664476.3664487

ARES 2024, July 30–August 02, 2024, Vienna, Austria

ACM ISBN 979-8-4007-1718-5/24/07
https://doi.org/10.1145/3664476.3664487

1 INTRODUCTION

In the initial quarter of 2023, Netflix's revenue surged to $8.19 billion, marking an eight-fold growth when compared to figures from ten years ago [25]. This exemplifies the booming growth trend within the digital market. Alongside Netflix, also other notable contributors exist including streaming providers like SKY, HBO, Disney+, gaming platforms such as Steam and Ubisoft Connect, and diverse digital services like digital government or digital services in the automotive sector. In order to facilitate the availability of diverse digital services anytime and anywhere, providers must exert control over content accessibility.

Concurrently, these business models are evolving in complexity, necessitating providers to increasingly refine distinctions regarding content access privileges. For instance, streaming platforms like Disney+ or Amazon Prime operate on a subscription basis; however, certain content may require additional payment. Similarly, Steam hosts an extensive library of tens of thousands of games, but users have access only to those for which they have paid. In essence, the paradigm has shifted from a binary "all or nothing" user access policy to a more nuanced approach wherein users are granted specific subsets of content. In addition to these developments, the market is witnessing an increase in its diversity. This entails that providers are now offering services from other providers as well. For example, companies like T-Mobile or Verizon bundle services such as mobile phone contracts with platforms like Disney+. As a result, content right holders are confronted not only with directly managing access to their own products but also with ensuring that third-party entities, which facilitate access to their contents, implement the access protocols accurately.

In order to address these multifaceted challenges with regard to access to content, one promising solution for managing access among various entities is the application of attribute-based encryption. Over the past years, many different approaches have been presented in the literature, spanning a wide variety of features. Examples include facilitating multiple authorities, revoking access privileges for members, content verification, and the ability to modify access permissions for encrypted files without necessitating decryption. With this flood of new encryption schemes with many new properties, it is important for businesses that these schemes really offer the promised protection and function without errors. That this is not always the case we have already shown in our prior work [Annonymised source], where we were able to prove an error in an encryption scheme during the performance evaluation of a new type of Secure Group Communication scheme. The same happened in this evaluation of the attribute-based scheme proposed by Yu et al. [32], which seemed to us to be a

promising candidate to solve the increasingly complex challenges of rights management in the digital market. The scheme [32] promised capabilities such as accommodating multiple authorities, enabling content verification, and facilitating access rights revocation. However, the rights management of [32] can be partially leveraged. For this reason, this work presents a corresponding security analysis of this attribute-based scheme in order to show the problems in this procedure on the one hand and on the other hand to emphasize herewith that it is of crucial importance to check the flood of novel proposed systems for their actual safety efficiency. Our contributions consist specifically of the following points:

- We present an attack vector on the attribute-based scheme proposed by Yu et al. [32], which allows a malicious data user to decrypt content for which she does not have the appropriate permission or required attributes.
- We present a way to close the attack vector on the attribute-based scheme proposed by Yu et al. [32] which we identified.

The remainder of this paper is structured as follows. In Section 2, we introduce the mathematical foundations essential for understanding the proposed attribute-based scheme by Yu et al. [32], as well as the basic concept of attribute-based schemes. The functionality of the attribute-based scheme proposed by Yu et al. is then presented in Section 3. Then, in Sections 4 and 5, we present our threat model, as well as the attack vector we identified for Yu et al.'s scheme. We then justify the assumptions we made for our attack vector in Section 6, and as part of this we also present our implementation of the attribute-based scheme proposed by Yu et al. and the attack on it. In Section 7, we present a way to close the attack vector we found. We conclude the paper with a review of related work in Section 8, as well as a summary in Section 9.

2 BACKGROUND

In this section we introduce basic information that is essential for further understanding. This includes the mathematical foundations and the concept of attribute-based encryption schemes.

2.1 Mathematical Foundations

First, we introduce the concept of access policies, which is to be realized by attribute-based encryption schemes. Then we define bilinear maps, on which many attribute-based encryption schemes are based, such as the scheme we are analyzing. Then we introduce the concept of linear secret sharing schemes, which is used by the scheme under consideration to realise its access policy.

2.1.1 *Access Structure.*

In attribute-based encryption, access structures are used to describe all possible attribute sets that allow a user to decrypt a given ciphertext.

Definition 2.1 (Access Structure [2])**.** Let $\mathbb{P} = \{P_1, P_2, \ldots, P_n\}$ be a set of parties. A collection $\mathbb{A} \subseteq 2^{\{P_1,P_2,\ldots,P_n\}}$ is monotone if for any B and C: if $B \in \mathbb{A}$ and $B \subseteq C$ then $C \in \mathbb{A}$. An access structure (respectively, monotone access structure) is a collection (respectively, monotone collection) $\mathbb{A}$ of nonempty subsets of $\{P_1, P_2, ..., P_n\}$, i.e., $\mathbb{A} \subseteq 2^{\{P_1,P_2,\ldots,P_n\}} \setminus \{\emptyset\}$. The sets in $\mathbb{A}$ are called the authorized sets, and the sets not in $\mathbb{A}$ are called the unauthorized sets.

For the rest of this paper, the term access structure only refer to monotone access structures. The set of parties $\mathbb{P}$ in the definition corresponds to the universe of attributes in an attribute-based encryption scheme.

2.1.2 *Pairing-based cryptography.*

Bilinear Pairing is a field of cryptography that utilizes a mapping $e : G_1 \times G_2 \rightarrow G_T$ from two groups to a third group. G_1 and G_2 are usually points on an elliptic curve and G_T is a finite field. More specifically, the elliptic curve needs to be a so-called pairingfriendly curve, such as a Barreto-Naehrig (BN) curve or a Barreto-Lynn-Scott (BLS) curve [31].

Before we define the term bilinear map, we first introduce the concepts on which this definition is based. This includes the concept of cyclic groups.

Definition 2.2 (Group [10])**.** Let G be a set and $*$ a binary operation. We call the tuple $(G, *)$ a group if and only if it fulfills the following properties:

- For all $x, y \in G$ it follows, that $x * y \in G$,
- For all $x, y, z \in G$ it follows, that $(x * y) * z = x * (y * z)$,
- It exists the identity element $1 \in G$ so that $x * 1 = x = 1 * x$ applies to all $x \in G$
- For all $x \in G$ it exist $x^{-1} \in G$, so that $x * x^{-1} = 1 = x^{-1} * x$

Definition 2.3 (Cyclic Group [10])**.** Let $C = (G, *)$ be a group. We call C a cyclic group if $a \in G$ exist, so that we can create every $y \in G$ by calculating a^i, where $i \in \mathbb{N}$. The element a with this property is also called generator of G and we write: $C =< a >$.

Definition 2.4 (Bilinear Mapping [27])**.** Let $(G_1, +)$ and $(G_2, +)$ be two additive cyclic groups of (nearly) prime order q with $G_1 =< P >$ and $G_2 =< Q >$, $(G_T, *)$ be a multiplicative cyclic group of order q with $G_T =< g >$. We write as usual 0 for the identity elements of G_1, G_2 and 1 for G_T . A pairing or a bilinear map is a map $e : G_1 \times G_2 \rightarrow G_T$ satisfying the following properties:

- Bilinearity: For all P_1, $P_1' \in G_1$, Q_1, $Q_1' \in G_2$, e is a group homomorphism in each component, i.e. :
  - $e(P_1 + P_1', Q_1) = e(P_1, Q_1)e(P_1', Q_1)$
  - $e(P_1, Q_1 + Q_1') = e(P_1, Q_1)e(P_1, Q_1')$
- Non-degeneracy: e is non-degenerate in each component, i.e. :
  - For all $P_1 \in G_1$, $P_1 \neq 0$, there is an element $Q_1 \in G_2$ such that $e(P_1, Q_1) \neq 1$
  - For all $Q_1 \in G_2$, $Q_1 \neq 0$, there is an element $P_1 \in G_1$ such that $e(P_1, Q_1) \neq 1$
- Computability: There exists an algorithm which computes the bilinear map e efficiently.

If $G_1 = G_2$ it follows from the bilinearity that, $\forall$ ($g \in G_1$ and $l, k \in \mathbb{Z}_q$): $e(g^l, g^k) = e(g, g)^{lk}$ and $\forall$ ($g \in G_1$ and $l, k \in \mathbb{Z}_q$): $e(g^l, g^k) = e(g^k, g^l) = e(g, g)^{lk}$ [12].

2.1.3 *Linear Secret Sharing Schemes (LSSS).*

Secret-sharing is a technique that allows a dealer to split up a secret value into multiple so-called shares, which are then distributed to a group of users. To recover the secret value again, a user must obtain a certain subset of all created shares. For example, in a (t, n)-threshold secret-sharing scheme, a user must obtain at least t of the total n shares to recover the secret. In the following, we first define the basic term secret sharing before introducing the term linear secret sharing.

Definition 2.5 (Secret-Sharing [2])**.** Let S be a finite domain of secrets. A secret-sharing scheme π realizing an access structure $\mathbb{A}$ is a scheme in which the input of the dealer is a secret $s \in S$ such that the following two requirements hold:

- Reconstruction requirement: The secret s can be reconstructed by any authorized set. Or more formally, for any set $G \in \mathbb{A}$ $(G = \{i_1, ..., i_{|G|}\})$, there exists a reconstruction function $h_G : S_{i_1} \times ... \times S_{i_{|G|}} \rightarrow S$ such that for every secret s, and every random input r, it follows: if $\pi(s, r) = < s_1, s_2, \ldots, sn >$ then $h_G(s_{i_1}, \ldots, s_{i_{|G|}}) = s$.
- Security requirement: Every unauthorized set of parties cannot reveal any partial information about the secret. We state this condition explicitly: for any set $B \notin A$, for every two secrets $a_1, a_2 \in S$, and for every vector of possible shares $\{s_i\}_{i \in B}$:

$$Pr[\bigwedge_{P_i \in B} \pi_i(a_1, r) = s_i] = Pr[\bigwedge_{P_i \in B} \pi_i(a_2, r) = s_i]$$

Where the probabilities are taken over the random input of the dealer. $pi_i(x, r)$ denotes the i-th share of $\pi(x, r)$.

Definition 2.6 (Linear Secret-Sharing [29])**.** We call a secret-sharing scheme π over a set of parties P linear (over $\mathbb{Z}_p$) if it fulfils the following requirements [29]:

- For each party the shares form a vector over $\mathbb{Z}_p$.
- A Matrix M with l rows and n columns exists, which we call the share-generating matrix for π. The function p defines the party labeling row i as $p(i)$, for all $i = 1, \ldots, l$. Let $s \in \mathbb{Z}_p$ be the secret to be exchanged and $r_2, \ldots, r_n \in \mathbb{Z}_p$ have been chosen at random, then Mv is the vector of l parts of the secret s according to π, where $v = (s, r_2, \ldots, r_n)^T$. Thereby $(Mv)_i$ represents the share of the party $p(i)$

2.2 Concept of Attribute-based Encryption Schemes

The main purpose of attribute-based encryption schemes is to define access policies that can be used to determine who can decrypt certain data [23]. As an example of how such an access policy can look in practice, an example of the following academic situation is given in Figure 1. The proposal for the exam in the course computer science (CS) should only be accessible to the professor of this course or the teaching assistants of this course. The access policy thus corresponds to the Boolean formula (Role: *Professor* AND Course: CS) OR (Role: *Teaching-Assistant* AND Course: CS)), which is shown as a tree in Figure 1. In addition to the tree policies just presented, threshold policies, AND policies and the linear secret sharing scheme matrix (LSSS) are also used as access policies in existing attribute-based schemes. The scheme we are analyzing implements its access policy by means of LSSS. The access policy of an attribute-based scheme can either be encoded in the secret keys for decryption or specified by the ciphertext. The former variant is also called Key-Policy Attribute-Based Encryption (KP-ABE) scheme and the latter Cipertext-Policy Attribute-Based Encryption (CP-ABE) [8, 21].

Figure 2 summarizes all essential information about the basic concept of attribute-based schemes. Since the scheme we want to analyze belongs to the category of CP-ABE schemes, the concept of CP-ABE schemes is shown accordingly in this figure. CP-ABE

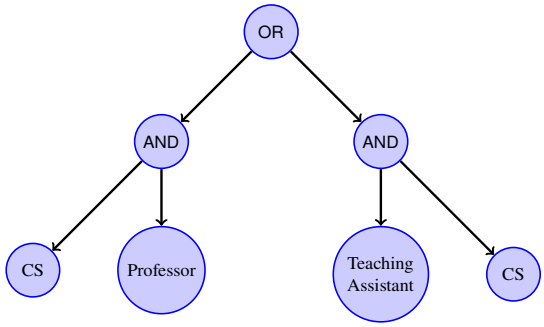

Figure 1: Example representation of an access policy as a tree, according to Priya et al. [23].

schemes want to allow a data owner to encrypt its data by means of a chosen access policy, so that only certain data users can decrypt this data. For this purpose, corresponding keys for encryption and decryption must be provided and distributed. This task is taken over by the authority, which provides the data users with corresponding secret keys for decryption and the data owners with a public key for encryption. Thereby the public key was created by the authority on the basis of the access policy of the data owner, which means that this access policy is part of the ciphertext after decryption. The access policy is expressed in this example by a tree policy. The cloud serves as a cache in which the data owner can store her encrypted data so that the data users can download it later.

3 FUNCTIONALITY OF THE ANALYSED ATTRIBUTE-BASED SCHEME

The attribute-based scheme presented by Yu et al. [32], whose security we would like to analyse, belongs to the category of CP-ABE schemes. Compared to the basic concept presented in Figure 2, Yu et al. schemes differ in that multiple authorities exist, called Attribute Authority (AA), which are managed by a Central Authority (CA). Before we analyse the attribute-based scheme presented by Yu et al. in terms of its security, we first present how it works in detail. For this purpose, we start with the set up of the CA as well as the AAs. We then show how a new data user is registered, how a data owner can encrypt its data and how a data user can decrypt it. Finally, we show how an attribute can be withdrawn from a data user

3.1 Set Up of the Central Authority (CA)

Setting up the CA involves the following two tasks: (1) generating the global public parameters (GPP) and (2) generating the master secret key (MSK). To generate the GPP, the CA must first choose the global security parameter λ, and define two cyclic groups G_1 and G_T, both of which have order p, where p is a prime number. On these two groups, the CA must specify the bilinear map $e : G_1 \times G_1 \rightarrow G_T$, where g is a generator of G_1 and $e(g, g)^{\alpha}$ is a generator of G_T. Here α is chosen randomly from $\mathbb{Z}_p$. Furthermore, the CA must choose a function $H(*)$ that maps a single attribute to an element of the group G_1, and a random $c \in \mathbb{Z}_p$ to be able to compute the value g^c. Afterwards, the CA is able to publish the global public parameters GPP as follows:

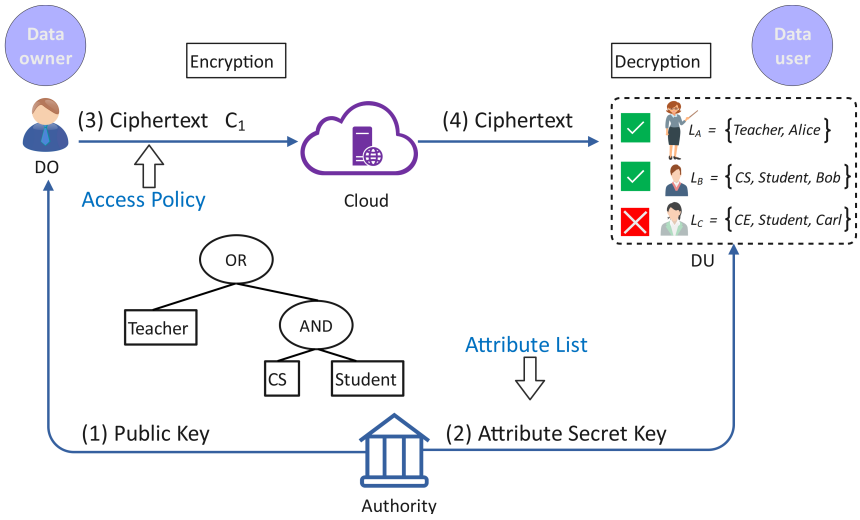

Figure 2: Concept of CP-ABE in the style of [33].

$$GPP = (g, g^c, H(*)) \tag{1}$$

To create the master secret key (MSK) for the CA, we first assume that the AAs $AA_{aid_1}, \ldots, AA_{aid_m}$ exist. For each of these AA_{aid_i}, the CA chooses a random $n_{aid_i} \in \mathbb{Z}_p$. Thereby it must hold, that $\sum_{i=0}^{m} n_{aid_i} = 0$. Then, for each AA_{aid_i}, the CA calculates the value $g^{n_{aid_i}}$ and transmits this via a secure channel to the corresponding AA_{aid_i}. The master secret key of the CA is then composed as follows:

$$MSK = (m, n_{aid_1}, \ldots, n_{aid_m}) \tag{2}$$

3.2 Attribute Authority (AA) Set Up

Setting up an AA consists of generating its private and public key. In the following, we assume that we want to set up AA_{aid_i} and that the parameters $g^{n_{aid_i}}$ and GPP have already been transmitted by the CA. The AA_{aid_i} then randomly chooses $\alpha_{aid_i}, \beta_{aid_i}, \gamma_{aid_i}, \delta_{aid_i} \in \mathbb{Z}_p$. The secret key SK_{aid_i} is then composed as follows:

$$SK_{aid_i} = (g^{n_{aid_i}}, \alpha_{aid_i}, \beta_{aid_i}, \gamma_{aid_i}, \delta_{aid_i}) \tag{3}$$

The generation of the public key requires that the attribute set S_{aid_i} of AA_{aid_i} is specified. It is assumed that the attributes of the individual AAs do not overlap. Otherwise, the AAs whose attribute sets overlap would have to coordinate with each other and select the same parameters for the affected attributes. Also, the list of data users who have the affected attributes must always be synchronised among the affected AAs. For each attribute $j \in S_{aid_i}$, two default data users u_0 and u_1 are created and added to the data user set U_j of the attribute j. Then, for all attributes $j \in S_{aid_i}$, for each data user i of U_j, the values $v_{i,j}$ and $t_{i,j} \in \mathbb{Z}_p$ are randomly chosen. In the next step, AA_{aid_i} calculates for each attribute $j \in S_{aid_i}$ the two parameters $PK_{1,j}$ and $PK_{2,j}$ using the Equations 4 and 5.

$$PK_{1,j} = g^{\frac{\prod_{i \in U_j}(v_{i,j})}{\beta_{aid_i}}} \tag{4}$$

$$PK_{2,j} = H(j)^{\gamma_{aid_i} \cdot \delta_{aid_i} \cdot \prod_{i \in U_j}(v_{i,j})} \tag{5}$$

In total, the public key of AA_{aid_i} is composed according to Equation 6. The respective values $w_{i,j}$ are calculated by means of Equation 7. In these two Equations, the function $e(x, y)$ is the bilinear mapping function introduced in Section 2.

$$PK_{aid_i} = (e(g,g)^{\alpha_{aid_i}}, g^{c \cdot \delta_{aid_i}}, \{PK_{1,j}, PK_{2,j}, \{w_{i,j}\}_{i \in U_j}\}_{j \in S_{aid_i}}) \tag{6}$$

$$w_{i,j} = t_{i,j} \cdot \prod_{k \in U_j, k \neq i} v_{k,j}^{-1} + v_{i,j} \tag{7}$$

3.3 Data User Registration

If a new data user u should be included in the system, she applies to the CA for her global data user identifier $ID_u \in \mathbb{Z}_p$ and the corresponding certificate with which she can identify herself to the AAs. The new data user then receives the secret key required for decryption from the individual AAs. In the following, we show which parameters the new data user receives from a specific AA, in this case AA_{aid_i}. First, AA_{aid_i} uses ID_u to calculate the parameters K_{u,aid_i} and K'_{u,aid_i} according to the Equations 8 and 9.

$$K_{u,aid_i} = g^{\alpha_{aid_i}} \cdot g^{c \cdot ID_u} \cdot g^{n_{aid_i} \cdot ID_u} \tag{8}$$

$$K'_{u,aid_i} = g^{\frac{ID_u}{\delta_{aid_i}}} \tag{9}$$

Then AA_{aid_i} calculates for each attribute j of the attribute set S_u of the new data user the parameters $K_{1,j}$, $K_{2,j}$, $K_{3,j}$ and $K_{4,j}$ using the Equations 10 - 13.

$$K_{1,j} = H(j)^{\beta_{aid_i} \cdot (ID_u + \gamma_{aid_i})} \tag{10}$$

$$K_{2,j} = H(j)^{\beta_{aid_i} \cdot ID_u (\gamma_{aid_i} - v_{u,j})} \tag{11}$$

$$K_{3,j} = H(j)^{-t_{u,j} \cdot v_{u,j} \cdot (ID_u + \gamma_{aid_i})} \tag{12}$$

$$K_{4,j} = H(j)^{\beta_{aid_i} \cdot \gamma_{aid_i} \cdot v_{u,j}} \tag{13}$$

The complete secret key of the new data user is then composed as follows:

$$SK = \{\{K_{u,aid_i}, K'_{u,aid_i}\}_{i \in \{1,\ldots,m\}}, \{K_{1,j}, K_{2,j}, K_{3,j}, K_{4,j}\}_{j \in S_u}\} \tag{14}$$

Since the values $\{PK_{1,j}, PK_{2,j}, \{w_{i,j}\}_{i \in U_j}\}_{j \in S_{aid_i}}$ depend on the existing data users, these values must be updated after adding a data user u. For each attribute j of the new data user u, the AA managing this attribute must extend the data user set U_j for attribute j by u. Then, all affected AAs must update their PKs. This is done for the AA_{aid_k} as follows: First, Equation 15 is used to calculate the new required values $w_{u,t}$ for each affected attribute t:

$$w_{u,t} = t_{u,t} \cdot \prod_{k \in U_j, k \neq u} v_{k,t}^{-1} + v_{u,t} \tag{15}$$

Thus, the new values $\tilde{w}_{i,j}$ for each affected attribute t can now be calculated using Equation 16.

$$\tilde{w}_{i,t} = (w_{i,t} - v_{i,t}) \cdot v_{u,t}^{-1} + v_{i,t} \tag{16}$$

Finally, for each affected attribute t, $PK_{1,t}$ and $PK_{2,t}$ can be updated using the Equations 17 and 18.

$$\tilde{PK}_{1,t} = PK_{1,t}^{v_{u,t}} \tag{17}$$

$$\tilde{PK}_{2,t} = PK_{2,t}^{v_{u,t}} \tag{18}$$

3.4 Encryption

To encrypt a message M, the data owner first selects a random $s \in \mathbb{Z}_p$ and computes the two ciphertext components C and C' according to the Equations 19 and 20. Thereby AA_c stands for the set of authorities whose attributes are to be used for encryption.

$$C = M \cdot \left(\prod_{aid_k \in AA_c} e(g,g)^{\alpha_{aid_k}} \right)^s \tag{19}$$

$$C' = g^s \tag{20}$$

The data owner can specify the access policy by means of an LSSS access structure, which consists of an L × N matrix, which is abbreviated as M_A in the following. The function p maps the rows of M_A to the corresponding attributes. After specifying M_A, the data owner chooses $y_2, \ldots, y_N$ and $r_1, \ldots, r_L$ randomly from $\mathbb{Z}_p$. Using these values, the data owner is now able to determine the vector $v = (s, y_2, \ldots, y_N)$ and calculate $\lambda_i = v \cdot M_{A_i}$ for each $i \in 1, \ldots, L$. Here M_{A_i} stands for the i-th row of M_A. In the final step of encrypting M, the data owner must compute the ciphertext components C_i, C'_i, D_i, and AK_i for each attribute j of the access policy according to the Equations 21, 22, 23 and 24.

$$C_i = g^{c \cdot \delta_{aid_k} \cdot \lambda_i} \cdot H(j)^{-r_j \cdot \gamma_{aid_k} \cdot \delta_{aid_k} \cdot \prod_{i \in U_j} v_{i,j}} \tag{21}$$

$$C'_i = PK_{1,j}^{r_j} = g^{\frac{r_i \cdot \prod_{i \in U_j} v_{i,j}}{\beta_{aid_k}}} \tag{22}$$

$$D_i = g^{r_i} \tag{23}$$

$$AK_i = g^{c \cdot \delta_{aid_k} \cdot \lambda_i} \tag{24}$$

The ciphertext CT of the message M is thus composed altogether as follows:

$$CT = \{(C, C', (C_1, C'_1, D_1, AK_1) \ldots, (C_L, C'_L, D_L, AK_L)\} \tag{25}$$

3.5 Decryption

To decrypt a ciphertext CT whose access policy consists of the attribute set I_M to the original message M, a data user must first determine which of its attributes S_u are contained in the access policy. Or more formally, the intersection $I = I_M \cap S_u$ must be determined. To do this, the data user can apply the function p to the associated access policy M_A to determine the attributes I_M contained therein. Knowing which attributes are relevant, the data user can determine the set AA_C of all AAs whose PKs were used for encryption. Then the data user can decrypt CT to M using the Equation 26. However, this first requires calculating ③ using Equation 27, which in turn requires calculating ① and ② using Equations 28 and 29. In Equation 29, the values of ω_i are chosen in such a way that they reconstruct the secret s.

$$M = \frac{C}{③} \tag{26}$$

$$③ = \frac{①}{②} \tag{27}$$

$$① = \prod_{aid_k \in AA_C} e(C', K_{u,aid_k}) \tag{28}$$

$$② = \prod_{j \in S_u} \left(\frac{e(C_i, K'_{u,aid_k}) e(C'_i, K_{1,j}^{w_{u,j}}) e(C'_i, K_{2,j}) e(D_i, K_{3,j})}{e(C'_i, K_{4,j})} \right)^{\omega_i m} \tag{29}$$

3.6 Data User Revocation

If a data user u is to be stripped of attribute j, this requires updating the public keys of the AA that manages attribute j. In the following, we assume that AA_{aid_k} manages attribute j and needs to update the following components of its public key: $PK_{1,j}$, $PK_{2,j}$, $\{\{w_{i,j}\}_{i \in U_j}\}_{j \in S_{aid_k}}$. The first two components are updated by the Equations 30 and 31. The required parameter UK can be determined by the Equation 32.

$$\tilde{PK}_{1,j} = PK_{1,j}^{UK} \tag{30}$$

$$\tilde{PK}_{2,j} = PK_{2,j}^{UK} \tag{31}$$

$$UK = v_{u,j}^{-1} \tag{32}$$

Then the data user u is removed from the data user set U_j of the attribute j, or more formally: $\tilde{U}_j = U_j \setminus \{u\}$. In the last step, for all

other data users $i \in U_j$ the value $w_{i,j}$ must be updated using Equation 33.

$$\{\tilde{w}_{i,j} = (w_{i,j} - v_{i,j}) \cdot v_{u,j} + v_{i,j}\}_{i \in U_j} \tag{33}$$

Existing ciphertexts whose access policy requires the attribute j like, e.g., $CT = \{(C, C', (C_1, C_1', D_1, AK_1), \ldots (C, C', (C_L, C_L', D_L, AK_L)\}$, can also change their access policy, so that the data user u, from whom the attribute j was withdrawn after CT was generated, can no longer decrypt it. To do this, either CT could be recalculated using the updated keys, or alternatively, CT could be updated using Equations 34 and 35 accordingly. Using these two Equations, for all $i \in \{1, \ldots L\}$, the values C_i and C_i' are adjusted.

$$\tilde{C}_i = C_i^{UK} \tag{34}$$

$$\tilde{C_i'} = {C'}_i^{UK} \tag{35}$$

4 THREAT MODEL

After presenting the attribute-based encryption scheme whose security we want to analyze, we define our threat model in the following. The authors of the scheme have already evaluated the security of their method against an outstanding attacker, who can collaborate with data users for attacks in [32]. For this purpose, it was assumed that the attacker herself is not a data user but can gain access to secret keys of data users.

For our security analysis, however, we assume that the attacker is a legitimate data user and has certain attributes or more precisely, a corresponding secret key. Our scenario can therefore be seen as a subset of the original thread model, since the attacker can only have access to one secret key, while in the original she can even have access to the secret keys of multiple users. The goal of this malicious data user is to access data for which she does not have the right attributes. This should not be possible in an attribute-based encryption scheme, as this is precisely the main selling point of such schemes, that a finely granular definition of access rights is possible based on the attributes. Thus, with attribute-based encryption schemes, it should not be possible for entities who are not legitimate data users and who do not have corresponding secret keys to decrypt data, as well as for legitimate data users to decrypt any data for which their attributes do not comply with the respective access policy.

5 ATTACK VECTOR

As attack, we assume that a malicious data user u was able to successfully infiltrate the system and got as far as the AAs generating a secret private key for her. In this process, the attribute set S_u was assigned to her. In addition, a data owner d, with $d \neq u$, encrypted her data and generated the ciphertext CT as a result. The attributes used for the access policy for the ciphertext CT form the set S_{CT} and $S_{CT} \not\subseteq S_u$ applies. An excerpt of all the information that the malicious data user has in this scenario is listed in Table 1. (Note: In this table, we assume that each data user knows its own identifier ID_u. Additionally, we assume that the number m of existing AAs is public knowledge. We deal with these two assumptions in more detail in our proof of concept implementation of the attack).

Table 1: Extract of the information accessible to the malicious data user.

$SK = \{\{K_{u,aid_i}, K'_{u,aid_i}\}_{i \in \{1,\ldots,m\}}, \{K_{1,j}, K_{2,j}, K_{3,j}, K_{4,j}\}_{j \in S_u}\}$
$CT = \{(C, C', (C_1, C_1', D_1, AK_1) \ldots, (C_L, C_L', D_L, AK_L)\}$
g
g^c
g^s
$u\ (= ID_u)$
m

To decrypt CT, the malicious data user would only need to compute Equation 27. We show below how this is possible for her without having the right attributes. The calculation of Equation 27 is traced back to the calculation of Equations 28 and 29. Since the Equation 28 depends only on information that the malicious data user already has, she can calculate it directly. To calculate the Equation 29, the malicious data user should actually have the necessary attributes. Therefore, the malicious data user should not be able to calculate this Equation. However, the malicious data user is still able to do so by using the authors' proof in [32], which is intended to prove that the decryption process of a ciphertext returns the original message. Equation 36 shows this proof. Note that Equation 36 is a transformation of Equation 29.

$$\begin{aligned} ② &= \prod_{j \in S_u} \left(\frac{e(C_i, K'_{u,aid_k}) e(C_i', K_{1,j}^{w_{u,j}}) e(C_i', K_{2,j}) e(D_i, K_{3,j})}{e(C_i', K_{4,j})} \right)^{\omega_i m} \\ &= \prod_{j \in S_u} e(g,g)^{c \cdot \lambda_i \cdot u \cdot w_i \cdot m} \\ &= e(g,g)^{c \cdot s \cdot u \cdot m} \end{aligned} \tag{36}$$

With the information available to the malicious data user, she can calculate the Equation 36 as follows: although the values of s and c might not be known to the data user, g^s and g^c are. Considering that bilinear maps must fulfill $e(g^l, g^k) = e(g,g)^{lk}$ for all $g \in G_1$ and $l, k \in \mathbb{Z}_p$ (see Section 2), it follows directly that $e(g,g)^{s \cdot c \cdot u \cdot m} = e(g^s, g^c)^{u \cdot m}$. Thus, the malicious data user can compute Equation 36 and, hence, also Equation 29. This allows the malicious data user to compute Equation 27, which enables her to decrypt the message M from CT using Equation 26.

Since a malicious data user can thus decrypt all ciphertexts without using any attributes, the revocation process of the attribute-based scheme, which is supposed to revoke certain attributes from data users who already have a secret key SK, is also without effect.

6 ATTACK - PROOF OF CONCEPT

In this section, we present how we have implemented the attribute-based scheme and the attack of a malicious data user in practice. Before that, however, we go into more detail about the assumptions made for the attack and show why they are justified.

6.1 Assumption Justification

Our attack by a malicious data user is based on the assumption that the attacker knows all the information from Table 1. In the following, we would like to show that this is the case. The first parameter in Table 1 is the attacker's secret key, which is obviously known to her because she is one of the legitimate data users (who, however, does not have all the attributes). The second parameter is the ciphertext CT, which is stored in the cloud and is accessible to the malicious data user, as the considered attribute-based scheme does not provide a control mechanism that determines who is allowed to download which data from the cloud.

The parameters g and g^c are part of the global public parameter GPP and thus also known to the attacker. Since the parameter g^s corresponds to the second value of the ciphertext CT, the attacker also knows this value. This leaves only the parameters m and u. According to the authors of [32], m is part of the master secret key of CA and should therefore not be known globally. However, m is known to all legitimate data users, since m can be read from the decryption algorithm. The authors of [32] assume in the decryption process that the ciphertext has been encrypted with the public keys of the AAs from the set AA_C.

However, the set AA_C must necessarily contain every existing AA, otherwise the proof of the correctness of the decryption process does not work. This proof can be seen in Equation 37. Essential for the proof is that the sum over the n_{aid_i} of all participants AA_i is equal to 0 (see red marking in the Equation). However, this is only certainly the case if n_{aid_i} of all AAs are summed up, since the respective values of n_{aid_i} have been chosen exactly so that they add up to 0 when summed over all existing AAs. Hence, each data user's secret key contains the two values K_{u,aid_i} and K'_{u,aid_i} from each existing AA. Thus, each data user can determine the value m via the number of these values.

$$
\begin{aligned}
① &= \prod_{aid_k \in AA_C} e(C', K_{u,aid_k}) \\
&= \prod_{aid_k \in AA_C} e(g^s, g^{\alpha_{aid_k}} g^{c \cdot ID_u} g^{n_{aid_k} \cdot ID_u}) \\
&= e(g,g)^{s \cdot c \cdot ID_u \cdot m} \cdot \Big(\prod_{aid_k \in AA_C} e(g^s, g^{\alpha_{aid_k}}) \Big) \cdot e(g,g)^{s \cdot ID_u \cdot \sum n_{aid_k}} \\
&= e(g,g)^{s \cdot c \cdot ID_u \cdot m} \cdot \prod_{aid_k \in AA_C} e(g^s, g^{\alpha_{aid_k}})
\end{aligned}
\tag{37}
$$

This leaves only the parameter u, which a malicious data user still needs to know for her attack. In the derivation of the correctness of the decryption process, the divisor ② is given as $e(g,g)^{c \cdot s \cdot ID_u \cdot m}$ in the calculation of ③, see Equation 27. However, in Equation 36, ② can also be written as $e(g,g)^{c \cdot s \cdot u \cdot m}$. Thus $e(g,g)^{c \cdot s \cdot ID_u \cdot m} = e(g,g)^{c \cdot s \cdot u \cdot m}$, which in turn has the consequence that $u = ID_u$ holds and, hence, the required parameter u can also be derived by the attacker.

Thus u corresponds to the global identifier ID_u of the malicious data user, which she has received from CA, so that the attacker also knows the value u. The attacker also knows the value u, since the authors of [32] do not specify that the global identifier of the data users should be hidden from them.

6.2 Attack Implementation

For our proof-of-concept implementation of the attack on the attribute-based scheme presented by Yu et al. [32], we first need a corresponding implementation of this attribute-based Scheme. For this purpose, we first show how we have implemented (1) the access policy or, more precisely, how we have generated the LSSS matrices and (2) the rest of the scheme. Then we present a concrete scenario with a simple example attack.

6.2.1 Implementation of the attribute-based Scheme proposed by Yu et al. [32]. In the following, we will first discuss the implementation of the access policy and then the implementation of the rest of the scheme. To generate an LSSS matrix [1] for an access policy, we followed the procedure from [9]. For this purpose, we assume that the access policy consists of a Boolean formula, which (1) in turn consists only of ANDs and ORs and (2) is represented by two-valued functions. For example, such a Boolean formula could look as follows: $AND(OR(A,B), OR(B,C))$. According to [9], the first step in creating an LSSS matrix for such a Boolean formula is to represent the Boolean formula as a tree (see Figure 3). To do this, we create a node with the corresponding function for the outermost function. Then we consider the first argument of the outermost function. If this is directly a Boolean variable, we create a leaf node for this variable and insert it as a child node accordingly. Should the first argument be a function, we create a node for this function as an interior node and proceed for this node analogously to the outermost function. For the second argument of the outermost function, we proceed analogously to the first argument.

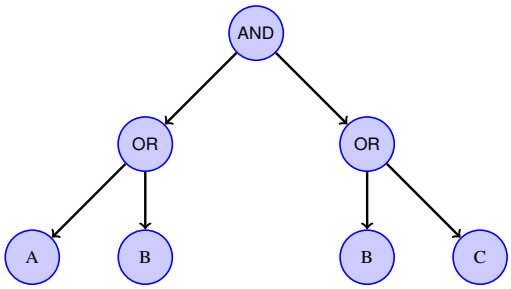

Figure 3: Converting a Boolean formula into a tree

To create the corresponding LSSS matrix for this tree, we first label the root with the vector (1) and set our global count variable c to 1. Then we go through the individual levels of the tree up to the leaves and label the nodes with vectors. For each level, we consider whether the node is an AND or an OR node. In the case of an OR node, we label the child nodes with the vector of this OR node. If the node is an AND node with the label vector v, we first create the vector v' by appending as many zeros to the vector v until the length of v' corresponds to the value c. Then we label one of the child nodes of the AND node with the vector $v'|1$, which corresponds to the vector

[1] Note: The formula symbol for the LSSS matrix in the description of the considered scheme is M_A

v' to which a 1 was appended. We label the other child node with the vector $(0)^c| - 1$, which corresponds to the vector $(0)^c$ to which the value -1 was attached. The vector $(0)^c$ in turn corresponds to a vector with c entries, which all have the value zero. After labelling the two child nodes of the AND node, we increase the counter c by one. When the whole tree is labelled, the labels of the leaves form the LSSS matrix. To do this, first bring all the labels of the leaves to the same length by filling them with zeros if necessary. Then each vector label of a leaf forms a row of the LSSS matrix. For our example formula, this would lead to the following LSSS matrix:

$$\begin{pmatrix} 1 & 1 \\ 1 & 1 \\ 0 & -1 \\ 0 & -1 \end{pmatrix}$$

The corresponding function p, which maps the rows 1 to 4 to the attributes A, B and C, would be defined as follows:

$$p(x) = \begin{cases} A & x = 1 \\ C & x = 4 \\ B & else \end{cases}$$

For the rest of the implementation of the scheme proposed by Yu et al. we followed the procedure given by the authors. For this, we also had to perform the required mathematical operations, especially the operations of bilinear maps. For this purpose, we used the PBC library [11], which implements bilinear maps.

6.2.2 Example Attack Scenario. As an example situation to demo the attack vector we found, we created the following situation using our implementation of the attribute-based scheme proposed by Yu et al. [32]. We created a central authority CA, which again manages the two authorities AA_1 and AA_2. Here AA_1 has the attribute set $\{1, 2\}$ and AA_2 has the attribute set $\{3\}$. Next, we generate the normal data user u_1, which has the attributes $\{2, 3\}$, and the malicious data user u_2, which is a legitimate data user registered in the system, but has no attributes at all. For this initial situation, we generate a random message m and encrypt it using the following access policy to the ciphertext c: $1\ OR\ (2\ AND\ 3)$. The tree representation of this access policy is illustrated in Figure 4.

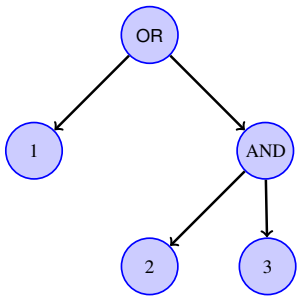

Figure 4: Acces policy tree used in our attack scenario.

From the access policy tree in Figure 4, we derive the following LSSS matrix and corresponding map function p for our attack scenario:

$$\begin{pmatrix} 1 & 0 \\ 1 & 1 \\ 0 & -1 \end{pmatrix}$$

$$p(x) = \begin{cases} 1 & x = 1 \\ 2 & x = 2 \\ 3 & else \end{cases}$$

In the above described attack scenario only u_1 should be able to decrypt c. However, u_2 is also able to decrypt c back to m, as our implementation shows. In the Figure 5 one can see the output that will be printed during the execution of our attack script. In this figure, besides the fact that both the attacker and the authorized recipient can decrypt the encrypted message, we have omitted the encoder and decoder for the implementation of the attack. Instead, we assume that the messages to be encrypted directly correspond to elements of the curve. However, a corresponding encoder and decoder can easily be complemented by the two functions $element\_from\_bytes$ and $element\_to\_bytes$ of the PBC library. Thus, one could implement the encoder by first converting a string to bytes and in turn converting it to a curve element using the $element\_to\_bytes$ function. For the decoder one could simply convert a curve element to bytes using $element\_from\_bytes$ and then convert the bytes back to a string.

Our implementation of the attack can be downloaded via https://github.com/WueSePrantl/YuAttack and instructions on how to get our implementation of the attack vector to work are given in the following bulleted list.

- Set up and start Ubuntu 22.04
- Install GMP library [2]
- Install M4 [3]
- Install PBC library [4]
- Install flex [5]
- Install bison [6]
- Install a version of OpenSSL [7]
- Copy params directory from the pbc directory to the folder of the attack

7 ADJUSTMENTS OF THE PROPOSED ATTRIBUTE-BASED SCHEME

In this section, we show how the attack we presented can be defended against. To do this, we first present our idea for a possible solution and then go into detail about how our solution approach changes the original procedure. Finally, we analyze the performance overhead caused by our modification.

[2]The official website of the GMP library: https://gmplib.org/#DOWNLOAD. Installation instructions can be found on the following website: http://rstudio-pubs-static.s3.amazonaws.com/493124_a46782f9253a4b8193595b6b2a037d58.html

[3]The official website of M4: https://www.gnu.org/software/m4/. Installation instructions can be found on the following website: https://howtoinstall.co/package/m4

[4]The official website of the PBC library https://crypto.stanford.edu/pbc. The PBC library can be downloaded from https://crypto.stanford.edu/pbc/download.html and installation instructions can be found on https://crypto.stanford.edu/pbc/manual/ch01.html

[5]The official website of flex: https://github.com/westes/flex. Installation instructions can be found on the following website: https://howtoinstall.co/package/flex

[6]The official website of bison: https://www.gnu.org/software/bison/. Installation instructions can be found on the following website: https://howtoinstall.co/package/bison

[7]We have chossen libssl-dev, installation instructions can be found on the following website: https://howtoinstall.co/package/libssl-dev

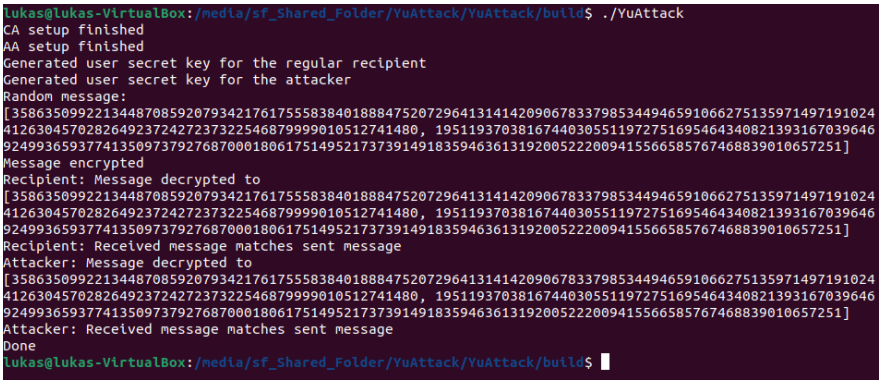

Figure 5: Output of the implementation of our attack vector. Both the authorized data user and the malicious data user without authorization can decrypt the message.

7.1 Attack Mitigation

In the previous section, we presented an attack vector and proof of concept for attacking the attribute-based scheme [32] proposed by Yu et al. In this section, we discuss possible countermeasures against this attack. For this purpose, we focus on the information that an attacker has in the current variant of the scheme (see from Table 1). Our goal is to find a solution that is as simple as possible and does not require any changes to the underlying mathematical relationships. Looking at Section 6.1 again, one can see that if it would be possible to hide one of the parameters m and ID_u from the data users, the attack would already not work in the form we described. This makes these two parameters our first option to consider for countermeasures. Therefore, we first consider whether the parameter m could be hidden from the data users. A malicious data user could infer this parameter by the number of components in its secret key. Thus, for a possible simple solution, one might evaluate whether one could also provide data users with only a subset of the components of their current SK so that the method would still work. However, since we have shown in Section 6.1 that for the decryption process $\sum n_{aid_k} = 0$ must hold, we cannot simply deprive data users of certain components of their current SK without further ado.

Therefore, we next consider whether the ID_u parameter, and hence u, can be hidden from the data user. Looking more closely at the concept of the attribute-based scheme proposed by Yu et al. in Section 3, we see that a data user does not need the ID_u parameter for the decryption process. The only purpose for which the data user needs ID_u is to communicate it to the AAs so that they can use it to generate her secret key. Therefore, in our opinion, a data user does not need to know what concrete value her identifier ID_u has. Thus, a simple countermeasure against the attack we have presented would be for CA to provide data users only with an encrypted version of ID_u, and to provide AAs with the corresponding decryption keys. In this way, a data user can communicate the specific value of her parameter ID_u to the AAs without knowing it herself.

7.2 Modifications of the Original Scheme

In the following, we describe the modifications to the original attribute-based scheme presented for the integration of our solution approach. The first modification is that during the set up phase of the CA, (1) a symmetric encryption scheme SE is specified for the master secret key and (2) a random key k_{SE} is generated for SE. The modified master secret key MSK^{mod} thus has the following form:

$$MSK^{mod} = (m, n_{aid_1}, \ldots, n_{aid_m}, SE, k_{SE}) \tag{38}$$

In order for the attribute authorities to later be able to decrypt the ID_us encrypted with the scheme SE and the symmetric key k_{SE}, they must accordingly also know SE and k_{SE}. Therefore, we adjust the set up phase of the AAs such that CA not only communicates the value $g^{n_{aid_i}}$ and the global public parameter GPP to the respective AAs, but also communicates SE and k_{SE} over a secure channel. These two additional parameters are also stored in the AAs' secret keys SK_{aid_i}. The modified AA secret keys $SK^{mod}_{aid_i}$ thus have the following form:

$$SK^{mod}_{aid_i} = (g^{n_{aid_i}}, \alpha_{aid_i}, \beta_{aid_i}, \gamma_{aid_i}, \delta_{aid_i}, SE, k_{SE}) \tag{39}$$

Finally, only the data user registration has to be adapted. This consists of the CA not transmitting the value ID_u directly to the data user, but encrypting it beforehand using the SE scheme and the k_{SE} key. The CA's certificate for the data user is issued accordingly for the encrypted variant of ID_u. If a data user subsequently wishes to have her secret key generated by the individual AAs, she sends them the certificate together with the encrypted value of ID_u.Thanks to their knowledge, the respective AAs are able to obtain the value of ID_u by decrypting it accordingly. The remaining steps of the attribute-based scheme proposed by Yu et al. do not need to be adjusted.

Through these adaptations, data users only receive the parameter u in encrypted form and can thus no longer carry out the attack we have described, as the parameter u must be available in unencrypted form for the described attack. With our adaptation, we eliminate the incorrect assumption of the security proof from the original publication that the product $u * m$ cannot be calculated by attackers. Since our adaptations limits the access to parameters, specifically u, and makes them only accessible to the authorities, the rest of the original security proof remains unaffected.

7.3 Modification Performance Overhead

Next, we analyze the additional effort caused by our modification in terms of computing times. Our modification does not generate

any additional performance overhead for the data users. For the *CA*, our modification means that (1) a random, secret symmetric key must be generated once and (2) an additional encryption must now be performed once for each data user. In our opinion, the one-time generation of a symmetric key is negligible from a performance perspective. To estimate the overhead caused by the additional encryption, we carried out corresponding measurements on an HPE ProLiant DL360 Gen9 server, with 8 CPU cores with 2.6 GHz each and 32 GB RAM. We used Ubuntu 22.04 as the operating system. On this server, we can perform symmetric AES encryption with 128-bit keys in under 100ms. For 1,000,000 data users, this would mean an additional effort of less than 2 days. As these are one-off additional costs, which can also be reduced by appropriate parallelization, we believe that they are still acceptable.

Our modification also results in a performance overhead for the individual *AA*. This consists of the fact that an additional decryption is now required for each data user. Analogous to the additional encryption of the *CA*, we can estimate the calculation time required for this for 1,000,000 data users at less than 2 days, which in our opinion is also still acceptable, as these costs only occur once.

8 RELATED WORK

Regarding the analysis of encryption schemes, there are different approaches in the literature. Surveys like [4, 7, 20, 24] attempt to classify encryption schemes and to provide overviews of their features and theoretical performance. Some works also address the question of how to determine the most appropriate encryption scheme for a specific use case based on theoretical analysis, e.g. [13]. In addition to these theoretical analyses, there exists also literature that measures performance of encryption schemes in practice on real hardware [14–19, 22]. We differ from these works in that we do not analyse performance or address the question of which encryption scheme would be best in a specific situation. Instead, our analysis of the attribute-based encryption scheme under consideration focuses on whether it really delivers the promised security.

In addition to analyzing the performance and features of encryption schemes, there is also work in the literature on analyzing encryption schemes in terms of their security. A new branch of research tries to test their security by means of neural networks. For example, Gohr et al. [5] implemented a key-recovery attack on 11-round SPECK-32/64, a block cipher, using deep learning. In doing so, Gohr et al.'s deep learning model was able to undercut the previous best time complexity of such an attack from 2^{46} to 2^{38}. Analogous results were obtained by the authors of [6], who were able to undercut the current state-of-the-art subkey recovery attack on the block cipher Simon32 in both time complexity and data complexity using a deep learning approach. In addition to cryptanalysis of the block cipher SPECK-32/64 and Simon32, neural networks have also been used to attack lattice-based cryptosystems based on the Learning With Errors (LWE) problem. For example, in [30], the authors were able to fully recover the corresponding secrets for small-to-mid size LWE instances with sparse binary secrets.

Among the most similar to our work are "classical" cryptanalyses that analyze the underlying mathematics of the respective encryption scheme and try to find flaws or shortcuts in the proofs that allow the respective scheme to be leveraged. For example, the work [1] analyzed the attribute-based scheme proposed by Boyen et al. [3] in terms of its security and was able to identify possible attacks on the LSSS matrix used, including mitigation strategies. Problems in the underlying mathematics could also be found and fixed by the authors of [26] in Wang et al.'s proposed attribute-based scheme [28].

This non exhaustive selection of related work shows that encryption schemes are analyzed with respect to a wide variety of criteria, such as performance, features, or their security. With the increasing number of encryption schemes being proposed, in our opinion, the thorough analysis of these schemes becomes increasingly important, ensuring that these schemes really provide the security they promise. But beside the security aspect, the proposed schemes' performance must be taken into consideration too, since use cases grow in complexity, the number of users steadily increases, and the associated rights management is continually required to become of finer granularity.

9 CONCLUSION

In recent years, the digital service industry has seen remarkable growth. Simultaneously, these services are becoming more intricate, requiring providers to fine-tune distinctions related to content access privileges. Even for services that operate on a subscription basis, some specific content may necessitate extra payments or incorporate third-party elements. To ensure diverse digital services are accessible anytime and anywhere, providers must maintain control over content accessibility. To tackle the diverse challenges, a promising approach is the implementation of attribute-based encryption. Over the past years, many different approaches have been presented in the literature, spanning a wide variety of features. To this end, it is imperative for businesses to verify that these schemes truly deliver the promised protection and function seamlessly. In our previous work [Annonymised source] , we identified errors in an encryption scheme while evaluating a new SGC system, a similar situation arose in this assessment of the ABE scheme proposed by Yu et al. [32]. Building upon our previous research, this study conducts a thorough security analysis of the ABE scheme, which seemed to us to be a promising candidate to solve the increasingly complex challenges of rights management in the digital market. Our goal was to shed light on the issues in this procedure and underscore the critical need to assess the safety and efficiency of newly proposed systems. Specifically, we identify an attack vector within this ABE scheme that enables a malicious user to decrypt content without the necessary permissions or attributes. Furthermore, we propose a solution to rectify this identified vulnerability.

REFERENCES

[1] Shweta Agrawal, Rajarshi Biswas, Ryo Nishimaki, Keita Xagawa, Xiang Xie, and Shota Yamada. 2022. Cryptanalysis of Boyen's attribute-based encryption scheme in TCC 2013. *Designs, Codes and Cryptography* 90, 10 (2022), 2301–2318.
[2] Amos Beimel et al. 1996. Secure schemes for secret sharing and key distribution. (1996).
[3] Xavier Boyen. 2013. Attribute-based functional encryption on lattices. In *Theory of cryptography conference*. Springer, 122–142.
[4] Omar Cheikhrouhou. 2016. Secure group communication in wireless sensor networks: a survey. *Journal of Network and Computer Applications* 61 (2016), 115–132.
[5] Aron Gohr. 2019. Improving attacks on round-reduced speck32/64 using deep learning. In *Advances in Cryptology–CRYPTO 2019: 39th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 18–22, 2019, Proceedings, Part II 39*. Springer, 150–179.
[6] Zezhou Hou, Jiongjiong Ren, and Shaozhen Chen. 2021. Cryptanalysis of Round-Reduced SIMON32 Based on Deep Learning. Cryptology ePrint Archive, Paper

2021/362. https://eprint.iacr.org/2021/362 https://eprint.iacr.org/2021/362.
[7] Bibo Jiang and Xiulin Hu. 2008. A survey of group key management. In *2008 international conference on computer science and software engineering*, Vol. 3. IEEE, 994–1002.
[8] C. Lee, Pei-Shan Chung, and M. Hwang. 2013. A Survey on Attribute-based Encryption Schemes of Access Control in Cloud Environments. *Int. J. Netw. Secur.* 15 (2013), 231–240.
[9] Allison Lewko and Brent Waters. 2011. Decentralizing attribute-based encryption. In *Annual international conference on the theory and applications of cryptographic techniques*. Springer, 568–588.
[10] Ben Lynn. [n. d.]. *Groups*. https://crypto.stanford.edu/pbc/notes/group/group.html Accessed: Dezember 15, 2023.
[11] Ben Lynn et al. 2006. PBC library manual 0.5. 11.
[12] Koti Nishat and BR Purushothama. 2016. Group-oriented encryption for dynamic groups with constant rekeying cost. *Security and communication networks* 9, 17 (2016), 4120–4137.
[13] Thomas Prantl, André Bauer, Lukas Iffländer, Christian Krupitzer, and Samuel Kounev. 2023. Recommendation of secure group communication schemes using multi-objective optimization. *International Journal of Information Security* (2023), 1–42.
[14] Thomas Prantl, Simon Engel, André Bauer, Ala Eddine Ben Yahya, Stefan Herrnleben, Lukas Iffländer, Alexandra Dmitrienko, and Samuel Kounev. 2022. An Experience Report on the Suitability of a Distributed Group Encryption Scheme for an IoT Use Case. In *2022 IEEE 95th Vehicular Technology Conference: (VTC2022-Spring)*. 1–7. https://doi.org/10.1109/VTC2022-Spring54318.2022.9860762
[15] Thomas Prantl, Lukas Iffländer, Stefan Herrnleben, Simon Engel, Samuel Kounev, and Christian Krupitzer. 2021. Performance Impact Analysis of Securing MQTT Using TLS. In *Proceedings of the ACM/SPEC International Conference on Performance Engineering* (Virtual Event, France) *(ICPE '21)*. Association for Computing Machinery, New York, NY, USA, 241–248. https://doi.org/10.1145/3427921.3450253
[16] Thomas Prantl, Dominik Prantl, Lukas Beierlieb, Lukas Iffländer, Alexandra Dmitrienko, Samuel Kounev, and Christian Krupitzer. 2021. Benchmarking of Pre- and Post-Quantum Group Encryption Schemes with Focus on IoT. In *2021 IEEE International Performance, Computing, and Communications Conference (IPCCC)*. 1–10. https://doi.org/10.1109/IPCCC51483.2021.9679365
[17] Thomas Prantl, Dominik Prantl, Lukas Beierlieb, Lukas Iffländer, Alexandra Dmitrienko, Samuel Kounev, and Christian Krupitzer. 2021. Performance Evaluation for a Post-Quantum Public-Key Cryptosystem. In *2021 IEEE International Performance, Computing, and Communications Conference (IPCCC)*. 1–7. https://doi.org/10.1109/IPCCC51483.2021.9679412
[18] Thomas Prantl, Peter Ten, Lukas Iffländer, Alexandra Dmitrenko, Samuel Kounev, and Christian Krupitzer. 2020. Evaluating the Performance of a State-of-the-Art Group-oriented Encryption Scheme for Dynamic Groups in an IoT Scenario. In *2020 28th International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS)*. 1–8. https://doi.org/10.1109/MASCOTS50786.2020.9285948
[19] Thomas Prantl, Peter Ten, Lukas Iffländer, Stefan Herrnleben, Alexandra Dmitrenko, Samuel Kounev, and Christian Krupitzer. 2021. Towards a Group Encryption Scheme Benchmark: A View on Centralized Schemes with Focus on IoT. In *Proceedings of the ACM/SPEC International Conference on Performance Engineering* (Virtual Event, France) *(ICPE '21)*. Association for Computing Machinery, New York, NY, USA, 233–240. https://doi.org/10.1145/3427921.3450252
[20] Thomas Prantl, Timo Zeck, Andre Bauer, Peter Ten, Dominik Prantl, Ala Eddine Ben Yahya, Lukas Ifflaender, Alexandra Dmitrienko, Christian Krupitzer, and Samuel Kounev. 2022. A Survey on Secure Group Communication Schemes With Focus on IoT Communication. *IEEE Access* 10 (2022), 99944–99962. https://doi.org/10.1109/ACCESS.2022.3206451
[21] Thomas Prantl, Timo Zeck, Lukas Horn, Lukas Iffländer, André Bauer, Alexandra Dmitrienko, Christian Krupitzer, and Samuel Kounev. 2023. Towards a Cryptography Encyclopedia: A Survey on Attribute-Based Encryption. *Journal of Surveillance, Security and Safety* (2023).
[22] Thomas Prantl, Timo Zeck, Lukas Iffländer, Lukas Beierlieb, Alexandra Dmitrenko, Christian Krupitzer, and Samuel Kounev. 2022. Towards a Cryptography Benchmark: A View on Attribute Based Encryption Schemes. In *2022 5th Conference on Cloud and Internet of Things (CIoT)*. 158–165. https://doi.org/10.1109/CIoT53061.2022.9766494
[23] Aayushi Priya. 2018. A Survey: Attribute Based Encryption for Secure Cloud. *IJOSTHE* 5 (06 2018), 12. https://doi.org/10.24113/ojssports.v5i3.70
[24] Li Shu-Quan. [n. d.]. A Survey on Key Management for Multicast. In *Proceedings of Second International Conference on Information Technology and Computer Science(ITCS)*. 309–312.
[25] statista. [n. d.]. *Netflix's Quarterly Revenue*. https://www.statista.com/statistics/273883/netflixs-quarterly-revenue/ Accessed: August 2, 2023.
[26] Yi-Fan Tseng, Chun-I Fan, Si-Jing Wu, Hsin-Nan Kuo, and Jheng-Jia Huang. 2019. Cryptanalysis and improvement on Wang et al.'s attribute-based searchable encryption scheme. *International Journal of Machine Learning and Computing* 9, 5 (2019), 636–643. https://doi.org/10.18178/ijmlc.2019.9.5.851
[27] Osmanbey Uzunkol and Mehmet Sabır Kiraz. 2018. Still wrong use of pairings in cryptography. *Appl. Math. Comput.* 333 (2018), 467–479.
[28] Haijiang Wang, Xiaolei Dong, and Zhenfu Cao. 2017. Multi-value-independent ciphertext-policy attribute based encryption with fast keyword search. *IEEE Transactions on Services Computing* 13, 6 (2017), 1142–1151.
[29] Brent Waters. 2011. Ciphertext-policy attribute-based encryption: An expressive, efficient, and provably secure realization. In *International workshop on public key cryptography*. Springer, 53–70.
[30] Emily Wenger, Mingjie Chen, Francois Charton, and Kristin E. Lauter. 2022. SALSA: Attacking Lattice Cryptography with Transformers. In *Advances in Neural Information Processing Systems*, S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (Eds.), Vol. 35. Curran Associates, Inc., 34981–34994.
[31] Ed. Y. Sakemi, T. Kobayashi, T. Saito, and R. Wahby. [n. d.]. *Pairing-Friendly Curves*. https://www.ietf.org/archive/id/draft-irtf-cfrg-pairing-friendly-curves-08.html Accessed: Dezember 15, 2023.
[32] Ping Yu, Qiaoyan Wen, Wei Ni, Wenmin Li, Caijun Sun, Hua Zhang, and Zhengping Jin. 2019. Decentralized, revocable and verifiable attribute-based encryption in hybrid cloud system. *Wireless Personal Communications* 106 (2019), 719–738.
[33] Yinghui Zhang, Robert H Deng, Shengmin Xu, Jianfei Sun, Qi Li, and Dong Zheng. 2020. Attribute-Based Encryption for Cloud Computing Access Control: A Survey. *ACM Comput. Surv.* 53, 4, Article 83 (Aug. 2020), 41 pages. https://doi.org/10.1145/3398036