

Received October 30, 2020, accepted December 6, 2020, date of publication December 11, 2020, date of current version December 23, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3044143

Attacking Pairing-Free Attribute-Based Encryption Schemes

JAVIER HERRANZ 

Department of Mathematics, Universitat Politècnica de Catalunya, 08034 Barcelona, Spain

e-mail: javier.herranz@upc.edu

This work was supported by the Spanish Ministerio de Ciencia e Innovación (MICINN), under Project MTM2016-77213-R and Project PID2019-109379RB-I00.

ABSTRACT Combining several results that have been published in the last years, it is known that it is impossible to design simple and secure attribute-based encryption schemes that work in (classical) settings like the RSA or the pairing-free discrete logarithm ones. The purpose of this article is to broadcast this message through a wide (maybe non-cryptographic) audience, specially now that attribute-based encryption is considered as a useful tool to secure real systems like the Internet of Things. Today, only attribute-based encryption schemes that employ tools like bilinear pairings or lattices can provide some real (and provable) level of security. As an example of the fact that this message is still unknown for many people, we revisit a (maybe non exhaustive) list of articles proposing such insecure attribute-based encryption schemes: we recall which of these schemes have already been attacked and we describe attacks for the other ones.

INDEX TERMS Attacks, attribute-based encryption, cryptographic protocols, data security.


I. INTRODUCTION

Attribute-based encryption (ABE for short) is a cryptographic primitive that allows fine-grained access to encrypted data. Secret keys of users and ciphertexts are associated to sets of attributes and decryption policies, in such a way that a user can decrypt a ciphertext if and only if the subset of attributes satisfies the decryption policy. When secret keys are associated to sets of attributes and ciphertexts are associated to decryption policies, we have ciphertext-policy attribute-based encryption (CP-ABE) [1]. When secret keys are associated to decryption policies and ciphertexts are associated to sets of attributes, we have key-policy attribute-based encryption (KP-ABE) [5].

ABE is receiving a lot of attention in the last years, due to its potential application in several real-life settings, including the Internet of Things (IoT) one. Unfortunately, most of the proposed ABE schemes employ sophisticated mathematical objects, like bilinear pairings or lattices. This leads to schemes where either the keys are very long or the running time of encryption/decryption is a bit high, quite far from the lightweight requirements imposed by settings like IoT. To remedy this situation, several authors have proposed ABE schemes which do not employ bilinear pairings or lattices,

but instead employ classical cryptographic operations, like RSA operations or operations in a cyclic group of prime order where the discrete logarithm is supposed to be hard (for instance, some sets of points of elliptic curves). The list of such articles contains [9], [11], [12], [16], [17], [19] and maybe some others. These articles even come with a formal security proof that relates the difficulty of breaking the corresponding ABE scheme with the hardness of solving some well-studied (and conjectured hard) mathematical problem, like the Decisional Diffie-Hellman (DDH) or the RSA problems.

For people in the cryptographic community, these schemes should look surprising: attribute-based encryption (originally fuzzy identity-based encryption [14]) was proposed as a generalisation of identity-based encryption (IBE), a notion that had been introduced in 1984 [15]. IBE has received a lot of attention (much more than ABE) and right now people are convinced of the real difficulty of designing IBE schemes which work (in a simple way) in classical settings like the RSA one or the pairing-free discrete logarithm one. It has even been proved that it is impossible to (generically, in a black box way) design IBE schemes based on trapdoor permutations or public key encryption or assumptions like DDH [2], [13]. In particular, the only secure IBE scheme in the pairing-free discrete logarithm setting seems to be that in [4]; it is mainly of theoretical interest because it is very inefficient: it uses garbled circuits (and so, non black

The associate editor coordinating the review of this manuscript and approving it for publication was Mamoun Alazab .

box operations) to bypass the aforementioned impossibility results.

A formal proof that constructing ABE schemes is more difficult than constructing IBE schemes was given in [7]. Therein, it was proved that any ABE scheme supporting at least AND policies can be transformed into an IBE scheme. All the ABE schemes in the list [9], [11], [12], [16], [17], [19] support at least AND policies. Therefore, combining any of these schemes with the construction in [7] would lead to an (efficient) IBE scheme in either the RSA or the pairing-free discrete logarithm setting. Since designing such IBE schemes in a secure way looks very hard (or impossible in some situations), the consequence is that the ABE schemes in those articles cannot be secure. A first evidence of this fact was already given in [7], where attacks on the CP-ABE schemes in [9], [12] were described. Later, an attack on the KP-ABE scheme in [19] was described in [17], where the authors propose a possible way to fix the insecurity of the scheme in [19]; the resulting KP-ABE still works in the pairing-free discrete logarithm setting, which once again should raise suspicions on its security. We note here that the KP-ABE scheme in [16] is very similar to that in [19], and that the attack described in [17] easily extends to the scheme in [16].

A. OUR CONTRIBUTIONS

The main goal of this article is to recall, once again, that designing simple and efficient ABE schemes that work in (classical) settings like the RSA or the pairing-free discrete logarithm ones is impossible,¹ given the results in [2], [7], [13]. Therefore, the purpose of this article is to discourage authors (and also journal editors and reviewers) from publishing articles with ABE schemes which will almost certainly be insecure. This is important for communities, like the IoT one, which do not necessarily have deep knowledge of sophisticated primitives like ABE. Actually many of the insecure schemes that we are discussing here are presented (sometimes even including the word IoT in the title) as a nice, secure and very efficient tool to implement IoT systems. So we hope that this manuscript may help specifically to the IoT community.

Therefore, the main contribution of this article is maybe more educational than scientific, and is contained essentially in this Section I. In any case, we give specific examples of the general message, by describing specific attacks on the ABE schemes proposed in [11], [17]. Combining these attacks with the existing attacks [7], [17] that we already mentioned in the previous section, we conclude that all the ABE schemes proposed in the above-mentioned list of articles are indeed insecure, as expected. This list of articles may be incomplete, but if there are other articles out of this list which propose similar ABE schemes, they must be insecure as well. We briefly discuss the case of two schemes in the literature

¹We stress that relaxed versions of IBE and ABE, such as bounded-IBE and bounded-ABE, can be obtained in these settings [6], [8], [18].

which are wrongly presented as ABE schemes, but which are not [3], [10].

Summing up, encryption schemes which are really attribute-based ones and which provide a formal and provable security guarantee require the use of sophisticated (and less efficient) mathematical tools like bilinear pairings or lattices. If some article claims to provide a secure ABE scheme in a classical setting, then there are many chances that either the scheme is not secure or the scheme is not fully attribute-based.

B. ORGANIZATION OF THIS ARTICLE

The rest of this article is organized as follows. In Section II we recall the notions of attribute-based encryption schemes: we give the syntax definition and the required security properties for such schemes. In Section III we present an explicit attack against the ABE scheme proposed in [17], that works in the pairing-free discrete logarithm setting. The attack is valid also for other similar schemes [16], [19]. In Section IV we present an explicit attack against the ABE scheme proposed in [11], that works in the RSA setting. Finally in Section V we briefly discuss a couple of schemes that are wrongly presented as ABE schemes [3], [10].

II. DEFINITIONS FOR ABE: PROTOCOLS AND SECURITY

The two main notions of attribute-based encryption, ciphertext-policy ABE and key-policy ABE, are dual: the roles of attribute sets and decryption policies are swapped. The result in [7] that an ABE scheme supporting AND policies implies an IBE scheme is valid for both notions (in [7] the formal proof is given only for the CP-ABE case, but the proof for the KP-ABE case works almost in the same way).

Since the ABE schemes that we are going to attack in this article belong to the two categories, we describe the general syntax of both CP-ABE and KP-ABE encryption. Regarding security, we describe it for both notions in a unified and quite informal way, which will be enough to understand why the considered schemes [11], [17] are insecure.

A. CP-ABE: SYNTACTIC DEFINITION

A ciphertext-policy attribute-based encryption scheme CP-ABE consists of four probabilistic polynomial-time algorithms:

- **CP-ABE.Setup**($1^\lambda, \mathcal{U}, \mathcal{F}$). The setup algorithm takes as input a security parameter λ , the total universe of attributes $\mathcal{U} = \{\mathbf{at}_1, \dots, \mathbf{at}_n\}$ and the family \mathcal{F} of decryption policies that the scheme supports. It outputs some public parameters \mathbf{pms} and a master secret key \mathbf{msk} .
- **CP-ABE.KeyGen**($A, \mathbf{msk}, \mathbf{pms}$). The key generation algorithm takes as input the master secret key \mathbf{msk} , the public parameters \mathbf{pms} and a set of attributes $A \subset \mathcal{U}$ satisfied by the user. The output is a private key \mathbf{sk}_A .
- **CP-ABE.Encrypt**($m, \mathcal{P}, \Gamma, \mathbf{pms}$). The encryption algorithm takes as input the public parameters \mathbf{pms} ,

a message m and a decryption policy (\mathcal{P}, Γ) where $\mathcal{P} \subset \mathcal{U}$ and $\Gamma \subset 2^{\mathcal{P}}$ satisfies $\Gamma \in \mathcal{F}$. The output is a ciphertext C .

- **CP-ABE.Decryption** $(C, \mathcal{P}, \Gamma, \mathbf{sk}_S, \mathbf{pms})$. The decryption algorithm takes as input a ciphertext C , a decryption policy (\mathcal{P}, Γ) , a secret key \mathbf{sk}_A and the public parameters \mathbf{pms} . The output is a message \tilde{m} .

The property of correctness requires that, if the following four protocols are run:

$(\mathbf{msk}, \mathbf{pms}) \leftarrow \text{CP-ABE.Setup}(1^\lambda, \mathcal{U}, \mathcal{F})$,
 $\mathbf{sk}_A \leftarrow \text{CP-ABE.KeyGen}(A, \mathbf{msk}, \mathbf{pms})$,
 $C \leftarrow \text{CP-ABE.Encrypt}(m, \mathcal{P}, \Gamma, \mathbf{pms})$,
 $\tilde{m} \leftarrow \text{CP-ABE.Decryption}(C, \mathcal{P}, \Gamma, \mathbf{sk}_S, \mathbf{pms})$,

then it holds $\tilde{m} = m$, if $A \cap \mathcal{P} \in \Gamma$ and $\Gamma \in \mathcal{F}$.

Regarding the family \mathcal{F} of admitted decryption policies, \mathcal{F} may for instance contain all the possible policies, $\mathcal{F} = \{\Gamma \subset 2^{\mathcal{U}}\}$, or may contain all the monotone increasing policies, $\mathcal{F} = \{\Gamma \subset 2^{\mathcal{U}}, \Gamma \text{ is monotone increasing}\}$, where Γ is monotone increasing if $A \in \Gamma, A \subset B$ implies $B \in \Gamma$. Some schemes may support only threshold decryption policies, $\mathcal{F} = \{\Gamma_{(t, \mathcal{P})}, \mathcal{P} \subset \mathcal{U}, 1 \leq t \leq |\mathcal{P}|\}$, and $\Gamma_{(t, \mathcal{P})} = \{A \subset \mathcal{P} \text{ s.t. } |A| \geq t\}$. A particular, more restrictive, case of threshold policies corresponds to **AND** policies, of the form $\Gamma_{(\{\mathcal{P}\}, \mathcal{P})} = \{\mathcal{P}\}$, containing only one subset, $\mathcal{P} \subset \mathcal{U}$.

B. KP-ABE: SYNTACTIC DEFINITION

A key-policy attribute-based encryption scheme **KP-ABE** consists of four probabilistic polynomial-time algorithms:

- **KP-ABE.Setup** $(1^\lambda, \mathcal{U}, \mathcal{F})$. The setup algorithm takes as input a security parameter λ , the total universe of attributes $\mathcal{U} = \{\mathbf{at}_1, \dots, \mathbf{at}_n\}$ and the family \mathcal{F} of decryption policies that the scheme supports. It outputs some public parameters \mathbf{pms} and a master secret key \mathbf{msk} .
- **KP-ABE.KeyGen** $(\mathcal{P}, \Gamma, \mathbf{msk}, \mathbf{pms})$. The key generation algorithm takes as input the master secret key \mathbf{msk} , the public parameters \mathbf{pms} and a decryption policy (\mathcal{P}, Γ) where $\mathcal{P} \subset \mathcal{U}$ and $\Gamma \subset 2^{\mathcal{P}}$ satisfies $\Gamma \in \mathcal{F}$. The output is a private key $\mathbf{sk}_{(\mathcal{P}, \Gamma)}$.
- **KP-ABE.Encrypt** (m, A, \mathbf{pms}) . The encryption algorithm takes as input the public parameters \mathbf{pms} , a message m and a set of attributes $A \subset \mathcal{U}$. The output is a ciphertext C .
- **KP-ABE.Decryption** $(C, A, \mathbf{sk}_{(\mathcal{P}, \Gamma)}, \mathbf{pms})$. The decryption algorithm takes as input a ciphertext C , the corresponding subset of attributes $A \subset \mathcal{U}$, a secret key $\mathbf{sk}_{(\mathcal{P}, \Gamma)}$ and the public parameters \mathbf{pms} . The output is a message \tilde{m} .

The property of correctness requires that, if the following four protocols are run:

$(\mathbf{msk}, \mathbf{pms}) \leftarrow \text{KP-ABE.Setup}(1^\lambda, \mathcal{U}, \mathcal{F})$,
 $\mathbf{sk}_{(\mathcal{P}, \Gamma)} \leftarrow \text{KP-ABE.KeyGen}(\mathcal{P}, \Gamma, \mathbf{msk}, \mathbf{pms})$,
 $C \leftarrow \text{KP-ABE.Encrypt}(m, A, \mathbf{pms})$,
 $\tilde{m} \leftarrow \text{KP-ABE.Decryption}(C, A, \mathbf{sk}_{(\mathcal{P}, \Gamma)}, \mathbf{pms})$,

then it holds $\tilde{m} = m$, if $A \cap \mathcal{P} \in \Gamma$ and $\Gamma \in \mathcal{F}$.

C. SECURITY OF ABE SCHEMES

An ABE scheme is secure if an adversary who knows many secret keys is not able to obtain any information on the plaintext m encrypted in a ciphertext C , of course assuming that none of the secret keys alone is enough to decrypt C .

That is, in the CP-ABE setting, the adversary is able to query for secret keys of subsets of attributes A_1, A_2, \dots, A_k and then he requests for an encryption C^* of either m_0 or m_1 for a decryption policy (\mathcal{P}, Γ) such that $A_i \cap \mathcal{P} \notin \Gamma$, for all $i = 1, \dots, k$. The CP-ABE scheme is secure if such an adversary cannot distinguish if the received ciphertext C^* decrypts to m_0 or to m_1 .

In the KP-ABE setting, the adversary is able to query for secret keys of decryption policies $(\mathcal{P}_1, \Gamma_1), \dots, (\mathcal{P}_k, \Gamma_k)$ and then he requests for an encryption C^* of either m_0 or m_1 for a subset of attributes $A \subset \mathcal{U}$ such that $A \cap \mathcal{P}_i \notin \Gamma_i$, for all $i = 1, \dots, k$. The KP-ABE scheme is secure if such an adversary cannot distinguish if the received ciphertext C^* decrypts to m_0 or to m_1 .

This is the basic idea, that can be formalised with an experiment involving a challenger and an adversary. The experiment can have slight modifications, for instance by requiring the adversary to choose the challenge decryption policy (in the CP-ABE case) at the beginning of the experiment or not, or by allowing the adversary to make also decryption queries to an oracle. These variations lead to different security notions like adaptive versus selective security, or security against chosen-ciphertext attacks versus security against chosen-plaintext attacks. This is not really relevant in this work, because the attacks that we are going to describe are simple selective and chosen-plaintext attacks; they work even for the most limited type of adversaries.

III. ATTACK AGAINST THE KP-ABE SCHEME [17], IN THE PAIRING-FREE DISCRETE LOGARITHM SETTING

The classical Discrete Logarithm framework consists of a cyclic group \mathbb{G} of prime order p . Examples of such groups are some groups of points in elliptic curves or subgroups of \mathbb{Z}_q , when $p|q - 1$. We will use additive notation in \mathbb{G} , that is, $\mathbb{G} = \{aP, a \in \{0, 1, \dots, p - 1\}\}$, where P is a generator of \mathbb{G} . The prime number p and the generator P are public information, available to everybody.

A. DESCRIPTION OF THE KP-ABE SCHEME IN [17]

Article [17] has two contributions- The first one is an attack against the KP-ABE scheme in [19], which is correct. The second one is a modification of the (insecure) scheme in [19] along with a security proof for the new KP-ABE scheme. This second contribution is incorrect; maybe the new scheme cannot be attacked with the same strategy as the authors of [17] used to attack the original scheme [19], but there are other attacks as we show here.

For simplicity, we only describe the Setup and Key Generation protocols of their KP-ABE scheme, and for the particular

case of policies $\Gamma_{(t, \mathcal{P})}$ consisting of a single (t, \mathcal{P}) -threshold gate on a subset $\mathcal{P} = \{\mathbf{at}_1, \dots, \mathbf{at}_k\} \subset \mathcal{U}$ of k attributes inside the global universe $\mathcal{U} = \{\mathbf{at}_1, \mathbf{at}_2, \dots, \mathbf{at}_n\}$ of n attributes. This will be enough to describe our attack and to understand that the scheme in [17] is not secure (in particular, the security analysis provided in [17] must be wrong).

Setup($1^\lambda, \mathcal{U}, \mathcal{F}_{\Gamma_{(t, \mathcal{P})}}$). The setup algorithm starts by choosing a cyclic group \mathbb{G} of prime order p , such that p is λ bits long, and a generator P of \mathbb{G} . A total of $n + 1$ random and independent elements $s, s_1, s_2, \dots, s_n \in \mathbb{Z}_p$ are chosen. For each $i \in \{1, \dots, n\}$, the value $P_i = s_i P$ is computed; the value $Y = sP$ is also computed. A suitable pseudo-random function **PRF** is chosen, whose outputs are elements of \mathbb{Z}_p .

The master secret key is $\mathbf{msk} = (s, s_1, s_2, \dots, s_n)$.

The public parameters of the system are $\mathbf{pms} = (p, \mathbb{G}, P, Y, P_1, \dots, P_n, \mathbf{PRF})$.

KeyGen($t, \mathcal{P}, \mathbf{msk}, \mathbf{pms}$). The key generation algorithm takes as input a subset of attributes $\mathcal{P} \subset \mathcal{U}$ and a threshold t satisfying $1 \leq t \leq |\mathcal{P}|$, which define a threshold decryption policy $\Gamma_{(t, \mathcal{P})}$, the master secret key \mathbf{msk} and the public parameters \mathbf{pms} .

Let us denote the subset of attributes as $\mathcal{P} = \{\mathbf{at}_1, \dots, \mathbf{at}_k\}$. Choose at random a seed r , and compute for each $j \in \{1, \dots, k\}$ the value $\alpha_j = \mathbf{PRF}(r, i_j)$.

Choose at random a polynomial $f(x) \in \mathbb{Z}_p[X]$ with degree at most $t - 1$ such that $f(0) = s$. For each $\mathbf{at}_{i_j} \in \mathcal{P}$ define the value $D_{i_j} = \frac{f(\alpha_j)}{s_{i_j}} \bmod p$ (here division means multiplication with the inverse modulo p).

The secret key that is sent to the user is $\mathbf{sk}_{(\mathcal{P}, t)} = (r, D_{i_1}, \dots, D_{i_k})$. Note that r must be included in the secret key of the user; otherwise, the user could not compute the values α_j which are needed to run polynomial interpolation, in the decryption phase.

B. THE ATTACK

If the scheme was secure, an adversary who could make (many) queries for the $(2, 2)$ -threshold policy defined on the set of two attributes $\{\mathbf{at}_1, \mathbf{at}_2\}$ should not be able to decrypt ciphertexts that have been computed for the set $\{\mathbf{at}_1\}$ consisting of a single attribute. However, we show that an adversary needs to make just two secret key queries for the $(2, 2)$ -threshold policy in order to get the value $\frac{s}{s_1} \bmod p$, which is the value that a (honest) user with $(1, 1)$ -threshold decryption policy on $\{\mathbf{at}_1\}$ would get. Therefore, our adversary would be able to decrypt the same ciphertexts as that honest user, which clearly breaks the (wrongly claimed) security of the KP-ABE scheme in [17]. In other words, what we present is a key-recovery attack, which is stronger than an attack against the IND-CPA property; in any case, the conclusion is that the KP-ABE scheme in [17] is not secure.

In the first query by the adversary, for the $(2, 2)$ -threshold decryption policy defined on the set $\{\mathbf{at}_1, \mathbf{at}_2\}$ of attributes, he obtains $\mathbf{sk}^{(1)} = (r^{(1)}, D_1^{(1)}, D_2^{(1)})$. Analogously, in the second query for the same decryption policy, the adversary

obtains $\mathbf{sk}^{(2)} = (r^{(2)}, D_1^{(2)}, D_2^{(2)})$. Let us define the four values $\alpha_i^{(j)} = \mathbf{PRF}(r^{(j)}, i)$, for $i, j \in \{1, 2\}$, which are known to the adversary.

The first key $\mathbf{sk}^{(1)}$ corresponds to a random degree-1 polynomial $f^{(1)}(x) = s + a^{(1)}x$, so we have

$$\begin{cases} D_1^{(1)} = \frac{s + a^{(1)}\alpha_1^{(1)}}{s_1} \bmod p & (1) \\ D_2^{(1)} = \frac{s + a^{(1)}\alpha_2^{(1)}}{s_2} \bmod p & (2) \end{cases}$$

Analogously, the second key $\mathbf{sk}^{(2)}$ corresponds to a random degree-1 polynomial $f^{(2)}(x) = s + a^{(2)}x$, so we have

$$\begin{cases} D_1^{(2)} = \frac{s + a^{(2)}\alpha_1^{(2)}}{s_1} \bmod p & (3) \\ D_2^{(2)} = \frac{s + a^{(2)}\alpha_2^{(2)}}{s_2} \bmod p & (4) \end{cases}$$

Isolating $a^{(1)}$ in (2) and substituting it in (1), we can write

$$\frac{s}{s_1} \left(1 - \frac{\alpha_1^{(1)}}{\alpha_2^{(1)}} \right) = D_1^{(1)} - D_2^{(1)} \cdot \frac{s_2}{s_1} \cdot \frac{\alpha_1^{(1)}}{\alpha_2^{(1)}} \bmod p \quad (5)$$

Defining $b^{(1)} = 1 - \frac{\alpha_1^{(1)}}{\alpha_2^{(1)}}$ and $c^{(1)} = D_2^{(1)} \cdot \frac{\alpha_1^{(1)}}{\alpha_2^{(1)}}$, we can rewrite (5) as

$$b^{(1)} \frac{s}{s_1} = D_1^{(1)} - c^{(1)} \frac{s_2}{s_1} \bmod p \quad (5^*)$$

Doing the same in (3) and (4), we end up in

$$b^{(2)} \frac{s}{s_1} = D_1^{(2)} - c^{(2)} \frac{s_2}{s_1} \bmod p, \quad (6^*)$$

where $b^{(2)} = 1 - \frac{\alpha_1^{(2)}}{\alpha_2^{(2)}}$ and $c^{(2)} = D_2^{(2)} \cdot \frac{\alpha_1^{(2)}}{\alpha_2^{(2)}}$.

Now we can multiply (5*) with $c^{(2)}$ and subtract the result of multiplying (6*) with $c^{(1)}$, to remove the element $\frac{s_2}{s_1}$, obtaining the equality

$$\left(b^{(1)}c^{(2)} - b^{(2)}c^{(1)} \right) \frac{s}{s_1} = c^{(2)}D_1^{(1)} - c^{(1)}D_1^{(2)} \bmod p \quad (7)$$

Due to the random properties of the pseudo-random function **PRF**, we have that $b^{(1)} \neq 0$, $b^{(2)} \neq 0$ and $b^{(1)}c^{(2)} - b^{(2)}c^{(1)} \neq 0$ with all but negligible probability. In particular, the value $\frac{s}{s_1} \bmod p$ can be computed by the adversary from equation (7), which finishes the description of the attack.

C. APPLICABILITY TO THE SCHEMES IN [16], [19]

The scheme in [19] can be seen as a simplified version of the scheme in [17], where the pseudo-random function **PRF** is not used at all and computation $\alpha_j = \mathbf{PRF}(r, i_j)$ is (essentially) replaced with $\alpha_j = i_j$. This simplified version was already attacked in [17], and we remark here that our attack of the previous section is obviously valid also for the scheme in [19].

The KP-ABE scheme in [16] is actually a generalization of the scheme in [19] where the life of the system is divided in periods and the keys for users are updated in each period i ,

according to some values t_i chosen by the master authority. There is also an additional factor $H(U_{ID})$ in the computation of the values D_i in the secret keys, where $H : \{0, 1\} \rightarrow \mathbb{Z}_p$ is a hash function. It is very easy to see that both the attack in [17] and our attack in the previous section are also valid for the scheme in [16].

IV. ATTACK AGAINST THE CP-ABE SCHEME [11], IN THE RSA SETTING

Up to our knowledge, the first published ABE scheme working in the RSA setting was that in [9], which was shown to be insecure in [7]. Recently, authors of the article [11] propose a modification of the (insecure) scheme in [9] and claim that it is secure. The security analysis, however, is just a discussion on why the attack proposed in [7] is not directly applicable to the new scheme. There is no formal security proof at all, and in fact we will show that the modified scheme is clearly insecure: essentially the same attack described in [7] against [9] also works for the modified scheme of [11].

A. DESCRIPTION OF THE CP-ABE SCHEME IN [11]

The RSA framework consists of an integer $N = pq$, product of two big prime numbers. Any integer g satisfying $\gcd(g, N) = 1$ enjoys the property $g^{\phi(N)} = 1 \pmod N$, where $\phi(N) = (p - 1)(q - 1)$. Factoring N is a very hard problem; this implies that computing $\phi(N)$ from N is also very hard.

The CP-ABE schemes in [9], [11] support AND decryption policies $\Gamma_{(\mathcal{P}_1, \mathcal{P})}$, defined on the total universe $\mathcal{U} = \{\mathbf{at}_1, \dots, \mathbf{at}_n\}$ of attributes. They use the following notation: any subset $A \subset \mathcal{U}$ will be represented by an n -bit string $a_1 a_2 \dots a_n$, where $a_i = 1$ if $\mathbf{at}_i \in A$, and $a_i = 0$ if $\mathbf{at}_i \notin A$. For example, if $n = 4$ and $A = \{\mathbf{at}_1, \mathbf{at}_4\}$, then the bit string corresponding to A is 1001.

With this notation, an AND decryption policy $\Gamma_{(\mathcal{P}_1, \mathcal{P})}$ may be represented by the bit string $b_1 b_2 \dots b_n$ corresponding to subset \mathcal{P} . If A is a subset of attributes (held by a user), with bit string $a_1 a_2 \dots a_n$, the condition that must be satisfied in order for that user to decrypt, $A \cap \mathcal{P} \in \Gamma_{(\mathcal{P}_1, \mathcal{P})}$, which is equivalent to $\mathcal{P} \subset A$, becomes $a_i \geq b_i, \forall i = 1, \dots, n$.

We describe now the relevant parts of some protocols of the scheme in [11], i.e., those necessary to understand our attack.

Setup($1^\lambda, \mathcal{U}, \mathcal{F}_{\text{AND}}$). The setup algorithm starts by choosing two prime numbers p, q such that $N = pq$ is λ bits long, along with a random element g satisfying $\gcd(g, N) = 1$. Then, if $\mathcal{U} = \{\mathbf{at}_1, \dots, \mathbf{at}_n\}$ contains n attributes, one considers an additional attribute \mathbf{at}_{n+1} which is 1 for every user and is 0 for every policy. The algorithm chooses $n + 1$ (prime) numbers p_1, \dots, p_n, p_{n+1} with $\gcd(p_i, \phi(N)) = 1$, and computes their inverses modulo $\phi(N)$, that is $q_i = p_i^{-1} \pmod{\phi(N)}$, for $i = 1, \dots, n, n + 1$.

Then, two random integers x, k are chosen, satisfying $\gcd(k, \phi(N)) = 1$ and $\gcd(x, q_i) = \gcd(k, q_i) = 1$, for all $i = 1, \dots, n, n + 1$. The following values are then computed: $d_{\mathcal{U}} = \prod_{\mathbf{at}_i \in \mathcal{U}} q_i, D_{\mathcal{U}} = g^{d_{\mathcal{U}}}, Y = g^x$ and $R = g^k$.

The master secret key is $\text{msk} = (x, k, p, q, q_1, \dots, q_n)$.

The public parameters of the system are $\text{pms} = (N, g, D_{\mathcal{U}}, Y, R, p_1, \dots, p_n, p_{n+1})$.

KeyGen($A, \text{msk}, \text{pms}$). The key generation algorithm takes as input a subset of attributes $A \subset \mathcal{U}$, the master secret key msk and the public parameters pms .

The value $d_A = \prod_{\mathbf{at}_i \in A} q_i$ is computed; remember that $\mathbf{at}_{n+1} \in A$ for all subsets of attributes corresponding to users of the system. The secret value for this subset A of attributes is a random pair $\text{sk}_A = (k_1, k_2)$ satisfying the condition $k \cdot k_1 + x \cdot k_2 = d_A \pmod{\phi(N)}$ (a possible way to generate such a pair of integers is described in [9], [11]).

Encrypt($\mathcal{P}, m, \text{pms}$). The encryption algorithm takes as input an AND policy, defined by a subset of attributes $\mathcal{P} \subset \mathcal{U}$, a plaintext m and the public parameters pms . The encryption consists of a one-time pad of m using the session key derived

from the value $K_m = g^{r_m \prod_{\mathbf{at}_i \in \mathcal{P}} q_i}$, where r_m is a random value chosen by the sender. This one-time pad is combined with the standard techniques (using hash functions) to achieve chosen-ciphertext security. The ciphertext contains additional elements $Y_m = Y^{r_m}$ and $R_m = R^{r_m}$, but the security of the encryption, in principle, is due to the fact that only users with a secret key sk_A satisfying $\mathcal{P} \subset A$ will be able to compute the value K_m from pms, Y_m, R_m and sk_A .

Indeed, if $\text{sk}_A = (k_1, k_2)$ satisfies $k \cdot k_1 + x \cdot k_2 = d_A \pmod{\phi(N)}$, then the user can compute the integer $\alpha = \prod_{\mathbf{at}_i \in A - \mathcal{P}} p_i$ from pms (here is where we need p_{n+1} to be public, because this factor p_{n+1} always appears in α) and then compute

$$(Y_m^{k_2} \cdot R_m^{k_1})^\alpha = \dots = K_m.$$

The bad news are that this is not the only way to compute K_m , as we will show in the next section: an adversary can combine secret keys for some subsets of attributes that do not contain \mathcal{P} and still compute K_m . Therefore, the scheme is insecure because an adversary controlling users who, individually, do not satisfy policy \mathcal{P} is able to decrypt a ciphertext addressed to policy \mathcal{P} .

B. THE ATTACK

Consider the case with $n = 2$ attributes in total, $\mathcal{U} = \{\mathbf{at}_1, \mathbf{at}_2\}$. The public parameters contain, in particular, values p_1, p_2, p_3 . We will consider a ciphertext computed for the policy $\mathcal{P} = \mathcal{U} = \{\mathbf{at}_1, \mathbf{at}_2\}$, and will show that an adversary who requests a secret key for subsets $B_1 = \{\mathbf{at}_1\}$ and $B_2 = \{\mathbf{at}_2\}$ is able to decrypt the ciphertext.

The ciphertexts contains elements $Y_m = Y^{r_m} = g^{x r_m}$ and $R_m = R^{r_m} = g^{k r_m}$, and the inherent one-time secret key for one-time pad is $K_m = g^{r_m q_1 q_2}$.

As a result of the secret key query for subset B_1 , the adversary gets $\text{sk}_{B_1} = (k_1^{(1)}, k_2^{(1)})$ such that $k \cdot k_1^{(1)} + x \cdot k_2^{(1)} = q_1 q_3 \pmod{\phi(N)}$.

As a result of the secret key query for subset B_2 , the adversary gets $\text{sk}_{B_2} = (k_1^{(2)}, k_2^{(2)})$ such that $k \cdot k_1^{(2)} + x \cdot k_2^{(2)} = q_2 q_3 \pmod{\phi(N)}$.

Now the attacker can compute the two values

$$T_1 = Y_m^{k_2^{(1)}} \cdot R_m^{k_1^{(1)}} = g^{r_m q_1 q_3} \quad T_2 = Y_m^{k_2^{(2)}} \cdot R_m^{k_1^{(2)}} = g^{r_m q_2 q_3}$$

The attacker proceeds by raising these two values to the (public) element p_3 , which is the inverse of q_3 modulo $\phi(N)$. Therefore, the attacker obtains the two values

$$\tilde{T}_1 = (T_1)^{p_3} = g^{r_m q_1} \quad \tilde{T}_2 = (T_2)^{p_3} = g^{r_m q_2}$$

Note that these values satisfy the equality $\tilde{T}_1^{p_1} = \tilde{T}_2^{p_2} = g^{r_m}$. Since (p_1, p_2) are prime numbers, they are co-prime, and by Bezout's identity, one can compute integer values a_1, a_2 such that $a_1 p_1 + a_2 p_2 = 1$. Now we can write

$$\begin{aligned} \tilde{T}_1 &= \tilde{T}_1^{a_1 p_1 + a_2 p_2} = \tilde{T}_1^{a_1 p_1} \cdot \tilde{T}_1^{a_2 p_2} = \\ &= \tilde{T}_2^{a_1 p_2} \cdot \tilde{T}_1^{a_2 p_2} = \left(\tilde{T}_2^{a_1} \cdot \tilde{T}_1^{a_2} \right)^{p_2} \end{aligned}$$

Therefore, the value $K := \tilde{T}_2^{a_1} \cdot \tilde{T}_1^{a_2}$ satisfies $K^{p_2} = \tilde{T}_1$. Raising this last equality to q_2 , we get $K = \tilde{T}_1^{q_2} = g^{r_m q_1 q_2} = K_m$.

Summing up, the adversary can compute the one-time key $K_m = K = \tilde{T}_2^{a_1} \cdot \tilde{T}_1^{a_2}$ and so decrypt the ciphertext.

V. BRIEF DISCUSSION ON THE SCHEMES IN [3], [10]

The scheme in [10] is presented as being a threshold-based ABE scheme, but it is actually a (strange) realization of the concept of fuzzy-identity based encryption [14]. In fuzzy IBE, there is a fixed threshold t , secret keys are associated to a subset of attributes W , ciphertexts are associated to a subset of attributes W' , and decryption is allowed whenever $W \cap W' \geq t$. But the scheme in [10] is not exactly a fuzzy-IBE scheme either, because the scheme requires $W' \cap W \geq t$ in order to give a secret key to a user with subset of attributes W . In other words, the scheme provides keys only to users who are authorized to decrypt with respect to a fixed subset W' . Note that, with this restriction, phases 1 and 2 in the security game described in Section 2.2 of [10] make no sense, because all such queries would be answered with the \perp symbol, according to the structure of their Key Generation protocol (in Section 2.1). Summing up, the description of the scheme in [10] is very strange and may have no application at all in real systems. In any case, if the scheme was modified to be a real fuzzy-IBE scheme, in the classical pairing-free discrete logarithm setting, our claim is that the scheme would be insecure, because fuzzy-IBE is a generalization of IBE (which is obtained when $t = 1$) and because secure generic constructions of IBE from the Diffie-Hellman assumption are impossible [13].

Regarding the scheme in [3], it is again a non-standard CP-ABE, because the participation of the master authority is required to decrypt any single ciphertext. Their idea is that the master entity, instead of giving the secret keys to the users, stores all these secret keys, and later in the decryption phase uses the stored keys to do part of the decryption. Here we also claim that the applicability of such a scheme in real-life

systems seems really poor. If one modifies their scheme so that the secret keys $SK_{i,GID} = k_i + H(GID)x$ are sent to the users, then the resulting (and now standard) CP-ABE scheme is clearly insecure: an adversary can make two queries for different users GID_1 and GID_2 , both of them with a single attribute $\{\mathbf{at}_i\}$. With a simple system of two equations and two unknowns, the adversary can recover the secret values k_1 and x from the received values SK_{1,GID_1} and SK_{1,GID_2} . After that, the adversary can make queries for arbitrary users GID with a single attribute $\{\mathbf{at}_i\}$, and will obtain k_i from the received value $SK_{i,GID}$, for $i = 2, 3, \dots, n$. That is, the adversary can obtain all the secret values of the master entity, namely x, k_1, \dots, k_n , with $n + 1$ secret key queries.

VI. CONCLUSION

Attribute-based cryptography is a very attractive concept, due to its potential applications in real-life systems (access control, the Internet of Things...). Thus, researchers try to find efficient, secure and (if possible) simple constructions of attribute-based constructions. In particular, some researchers have tried to design (simple and efficient) attribute-based encryption schemes that work in the (classical) discrete logarithm or RSA based settings. Unfortunately, simple and secure constructions in these settings are not possible even for the easier task of identity-based encryption [2], [13]. Therefore, the attribute-based encryption schemes proposed in these classical settings which are simple or efficient must be insecure, because otherwise they could be used to construct secure identity-based schemes [7].

This message is (or should be) well known by the cryptographic literature, at least from 2017. But articles proposing new candidates are still being submitted and published in prestigious journals, maybe circumventing a rigorous review process by cryptographic experts. The purpose of this article is to clarify this message to the widest possible audience; to illustrate and emphasize the message, we show explicit attacks against some attribute-based encryption schemes proposed in the last years.

REFERENCES

- [1] J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-policy attribute-based encryption," in *Proc. IEEE Symp. Secur. Privacy*, Apr. 2007, pp. 321–334.
- [2] D. Boneh, P. A. Papakonstantinou, C. Rackoff, and Y. Vahlis and B. Waters, "On the impossibility of basing identity based encryption on trapdoor permutations," in *Proc. FOCS*, 2008, pp. 283–292.
- [3] S. Ding, C. Li, and H. Li, "A novel efficient pairing-free CP-ABE based on elliptic curve cryptography for IoT," *IEEE Access*, vol. 6, pp. 27336–27345, 2018.
- [4] N. Döttling and S. Garg, "Identity-based encryption from the Diffie-Hellman Assumption," *Proc. Crypto*. Berlin, Germany: Springer-Verlag, 2017, pp. 537–569.
- [5] V. Goyal, O. Pandey, A. Sahai, and B. Waters, "Attribute-based encryption for fine-grained access control of encrypted data," *Proc. Comput. Commun. Secur.*, 2006, pp. 89–98.
- [6] J. Herranz, "Attribute-based versions of Schnorr and ElGamal," *Appl. Algebra Eng., Commun. Comput.*, vol. 27, no. 1, pp. 17–57, 2016.
- [7] J. Herranz, "Attribute-based encryption implies identity-based encryption," *IET Inf. Secur.*, vol. 11, no. 6, pp. 332–337, 2017.
- [8] G. Itkis, E. Shen, M. Varia, D. A. Wilson, and A. Yerukhimovich, "Bounded-collusion attribute-based encryption from minimal assumptions," in *Proc. PKC*. Berlin, Germany: Springer-Verlag, 2017, pp. 67–87.

- [9] V. Odelu, A. K. Das, M. Khurram Khan, K.-K.-R. Choo, and M. Jo, "Expressive CP-ABE scheme for mobile devices in IoT satisfying constant-size keys and ciphertexts," *IEEE Access*, vol. 5, pp. 3273–3283, 2017.
- [10] A. Karati, R. Amin, and G. P. Biswas, "Provably secure threshold-based ABE scheme without bilinear map," *Arabian J. Sci. Eng.*, vol. 41, no. 8, pp. 3201–3213, Aug. 2016.
- [11] D. Khandla, H. Shahy, M. Kumar Bz, A. R. Pais, and N. Raj. (2019). *Expressive CP-ABE Scheme Satisfying Constant-Size Keys and ciphertexts*. [Online]. Available: <https://eprint.iacr.org/2019/1257>
- [12] V. Odelu and A. K. Das, "Design of a new CP-ABE with constant-size secret keys for lightweight devices using elliptic curve cryptography," *Secur. Commun. Netw.*, vol. 9, no. 17, pp. 4048–4059, Nov. 2016.
- [13] P. A. Papakonstantinou, C. Rackoff, and Y. Vahlis. (2012). *How Powerful are the DDH Hard Groups*. [Online]. Available: <https://eprint.iacr.org/2012/653>
- [14] A. Sahai and B. Waters, "Fuzzy identity-based encryption," in *Proc. Eurocrypt*. Berlin, Germany: Springer-Verlag, 2005, pp. 457–473.
- [15] A. Shamir, "Identity-based cryptosystems and signature schemes," in *Proc. Crypto*. Berlin, Germany: Springer-Verlag, 1984, pp. 47–53.
- [16] K. Sowjanya, M. Dasgupta, S. Ray, and M. S. Obaidat, "An efficient elliptic curve cryptography-based without pairing KPABE for Internet of Things," *IEEE Syst. J.*, vol. 14, no. 2, pp. 2154–2163, Jun. 2020, doi: [10.1109/JSYST.2019.2944240](https://doi.org/10.1109/JSYST.2019.2944240).
- [17] S.-Y. Tan, K.-W. Yeow, and S. O. Hwang, "Enhancement of a lightweight attribute-based encryption scheme for the Internet of Things," *IEEE Internet Things J.*, vol. 6, no. 4, pp. 6384–6395, Aug. 2019.
- [18] S. Tessaro and D. A. Wilson, "Bounded-collusion identity-based encryption from semantically-secure public-key encryption: Generic constructions with short ciphertexts," *Proc. PKC*. Berlin, Germany: Springer-Verlag, 2014, pp. 257–274.
- [19] X. Yao, Z. Chen, and Y. Tian, "A lightweight attribute-based encryption scheme for the Internet of Things," *Future Gener. Comput. Syst.*, vol. 49, pp. 104–112, Aug. 2015.



JAVIER HERRANZ received the Ph.D. degree in applied mathematics from the Polytechnic University of Catalonia (UPC), Barcelona, Spain, in 2005. He was as a Postdoctoral Researcher with the Ecole Polytechnique, France, for nine months, the Centrum voor Wiskunde en Informatica, The Netherlands, for nine months, and IIIA-CSIC, Bellaterra, Spain, for two years. Since 2009, he has been with the Research Group MAK, Department of Applied Mathematics, UPC, first as a Postdoctoral Researcher, and currently as a Senior Lecturer. His research interests are related to cryptography and privacy of databases.

• • •