Applied Attribute-based Encryption Schemes

Sebastian Zickau[‡], Dirk Thatmann[‡], Artjom Butyrtschik[*], Iwailo Denisow[†], and Axel Küpper[‡]
Service-centric Networking | Technische Universität Berlin | Telekom Innovation Laboratories | Berlin, Germany
[‡]{sebastian.zickau|dirk.thatmann|axel.kuepper}@tu-berlin.de
[*]butyeboi@mailbox.tu-berlin.de | [†]iwailodenisow@mail.tu-berlin.de

***Abstract*—The advent of new cryptographic methods in recent years also includes schemes related to functional encryption. Within these schemes Attribute-based Encryption (ABE) became the most popular, including ciphertext-policy and key-policy ABE. ABE and related schemes are widely discussed within the mathematical community. Unfortunately, there are only a few implementations circulating within the computer science and the applied cryptography community. Hence, it is very difficult to include these new cryptographic methods in real-world applications. This article gives an overview of existing implementations and elaborates on their value in specific cloud computing and IoT application scenarios. This also includes a summary of the additions the authors made to current implementations such as the introduction of dynamic attributes.**

***Keywords*—Attribute-based Encryption, Applied Cryptography, Internet of Things, Cloud Computing Security**

I. Introduction

Many cryptographic schemes rely on the idea of a secret key, i.e., a private key within asymmetric cryptography or a symmetric key. In the past decade new cryptographic methods and schemes have arisen, such as *homomorphic encryption* and *functional encryption* [1]. The aim of these methods is to compute functions on encrypted data. The possibilities of outsourcing data and computational complexity to powerful cloud computing servers occurred coincidentally. Unfortunately, the maturity of the newly developed mathematical encryption schemes did not keep up with the rapidly increasing popularity of Internet of Things (IoT) and cloud computing solutions.

Subsets of functional encryption are the Attribute-based Encryption (ABE) schemes. These asymmetric schemes can be used to include access policies within the ciphertext or within the private key. From an applied science point of view, the promise of functional encryption, which should allow computational reasoning on encrypted data, are not fulfilled yet. Nevertheless, there are implementations of ABE subschemes circulating within the computer science community. This article gives an overview on current implementations and their usefulness within application scenarios mainly related to IoT and cloud computing use cases. The authors also made practical changes to current implementations.

The next section motivates the work. An introduction to ABE and its functionality is succeeding. Related work and use cases are described in Section IV. An overview of current implementations and their characteristics with performance comparisons and evaluations, also including an overview of the extensions made by the authors, are given in Section V. The publication is concluded together with an outlook.

II. Motivation

The popularity of services such as Dropbox, Google Drive, and ownCloud makes sharing files with friends and colleagues very convenient. With the introduction of new data distributing systems, questions regarding new security issues, data privacy and legal regulations arise. Securing data on a file level mostly includes symmetric or asymmetric cryptography, which means that the receiving entity must have a secret or a private key to decrypt data that is personally addressed and encrypted. The exchange of cryptographic keys and ensuring the identity of the receiver varies in each scenario. A further limitation is the lack of rights revocation and management of receivers, which often leads to costly data re-encryption. ABE addresses these limitations. Many theoretical ABE schemes exist, each supporting different functionalities and addressing different requirements.

In the context of IoT the communication between devices and services consist of short-lived messages including only of few information. Within the world of IoT the usage of ABE is considered in publish/subscribe (PUB/SUB) scenarios and messaging patterns, such as the Message Oriented Middleware (MOM) and Message Queue Telemetry Transport (MQTT) protocol, also involving more powerful intermediaries, e.g., broker services. In parallel with the advent of IoT the need of data protection increases. Since ABE promises advantages compared to classic pub-key systems, researchers try to take advantages of ABE in IoT scenarios. Thatmann et al. [2] propose a flexible and ABE scheme agnostic architecture supporting forward and backward message secrecy for group communications on top of MQTT. A group controller maintains ABE group attributes. The broker nodes are not able to access the encrypted message payload. The concept of trusted entities, able to take over encryption for computational weak devices, exists.

Within the scope of this article use cases are addressed, in which private cloud data and IoT messages are secured and shared among friends in social and professional contexts or between sensors and requesting services. Additionally the expressiveness of ABE is expanded to also include dynamic attributes such as location information of mobile devices and time constraints [3].

III. ABE Basics

ABE is a family of asymmetric encryption schemes. The first scheme of its kind was devised by Sahai and Waters and published in 2006 [4]. There exist a variety of ABE schemes. The information provided in this section describes the implementation by John Bethencourt [5] called *cpabe*.

In an ABE scheme there are at least three different types of keys. In a first step, a *key authority* generates a *secret master key* and an associated *public key*. The key authority should be a trusted entity since it has access to the secret master keys. These secret master keys can be used to generate *private keys*. A private key is needed to decrypt data. Public keys encrypt data and can be shared with others.

With ABE, it is possible to encrypt data with a fine-grained access control policy. This is achieved through the usage of *attribute sets*. Private keys, which can only be created using a secret master key, contain such attribute sets. During file encryption, the user needs to specify a *policy*, a Boolean formula describing what *attributes* are required for decryption, e.g., *((student and enrollment* $<$ *2012) or professor)*. This policy describes that a file is only accessible either by professors or students that have enrolled before 2012. When trying to decrypt the file, a private key is needed. If the attribute set of the private key satisfies this Boolean formula, then access to the plaintext is granted.

An advantage of ABE is that the generation of private keys and the decryption are decoupled. Private keys generated with certain attributes will always be able to decrypt data, for which they fulfill the Boolean formulas, without ever communicating to the key authority again. This is not always a desirable property, since the revocation of keys would not be possible without re-encrypting the data or redistributing every private key. Newer ABE approaches can account for this problem with revocation through a proxy [6] or through policy delegation [7], which allows updating the policy of a file to a more restrictive policy requiring only the public key. However, because of the decoupled nature of ABE, the attributes are static. They can only be changed by distributing a *new* private key.

An important characteristic of ABE is the prevention of collusion attacks. As an example, a file is encrypted with the policy *(X and Y)* and sent to two users. The first user has a private key with the attribute X while the second user has a private key with the attribute Y. Each of them is not able to decrypt the file, since both attributes taken separately do not satisfy the formula. It is then a desirable property of ABE that both users cannot combine their keys to form a single key, which will be able to decrypt the file. In fact, ABE does fulfill this property, called collusion resistance, as described in [5].

There are two main variants of ABE. In Ciphertext-Policy ABE (CP-ABE), the Boolean formula is saved in the ciphertext, i.e., the encrypted file. The attributes that are needed to satisfy policies are saved in a private key. In Key-Policy ABE (KP-ABE), these two roles are directly reversed: the private key holds the formula and the ciphertext possesses attributes. The latter variant could be applied to a digital rights management (DRM) use case, for instance.

There are four basic functions in CP-ABE:

- **Setup** - Generates a *public key* and an associated *secret master key*.
- **Encrypt** - Encrypts a file with a *public key* and an annotated *policy*.
- **Keygen** - Creates a *private key* with *attributes*. To do this, a *secret master key* is needed. This *private key* is associated with the *secret master key* and its *public key*.
- **Decrypt** - Needs a *ciphertext* and a *private key*. Decrypts the *ciphertext* only if the *attributes* in the *private key* satisfy the *policy* and the *private key* is associated with the *public key* that has been used initially for encryption.

There are two types of atomic formulas: *normal* and *numerical attributes*. A normal attribute is just a string of letters and digits that begins with a letter, e.g., *student*, *professor*, *workgroup303*, *male*, or *female*. Numerical attributes have a name and an associated value. As an attribute in a private key it would look like this: *hiringdate = 2014*, *yearofbirth = 1990*. In Bethencourt's cpabe implementation [5] integers in the range from 0 to $2^{64} - 1$ are supported. In access policies these numerical attributes can be compared to numbers with the following operators: $=, <, >, \geq, \leq$, e.g., *hiringdate* $\leq$ *2014*, *yearofbirth = 2000*, *accesslevel* $\geq$ *8*.

It is not possible to compare two different numerical attributes of a user, so the following is not possible: *count_students* $>$ *count_examinations*. The second argument of a numerical comparison always needs to be a number that is known at the time of encryption. For the access policy itself, there are three possible operators: *or*, *and*, and *of*. These operators can be combined to form complex formulas. A file might be encrypted with the following access policy: *((student and enrollment* $<$ *2014 and termsstudied* $>$ *3) or (student and termsstudied* $>$ *5) or professor)*.

As in other asymmetric encryption algorithms, ABE is intended to be deployed in a key encapsulation scheme, i.e., hybrid encryption scheme [8]. This is a typical way to construct systems using asymmetric encryption since the asymmetric part is computationally expensive. This is especially necessary for ABE, because of the complexity and the time needed to generate keys and encrypt or decrypt ciphertexts.

IV. Related Work and Use Cases

In this section an overview of current ABE scenarios is given and the added value of using these schemes, with regard to the requirements of such use cases, is highlighted.

Within the IoT world, there are certain characteristics that can be seen as fundamental. When involving a large amount of sensors and devices, which are attached to the Internet, there exist one-to-many and many-to-many communications, in the flavors of human-to-machine or machine-to-machine use cases. Securing the transferred data is sometimes difficult, as there are devices with low computational power and a variety of protocols, technologies, frameworks and standards beyond TCP/IP involved, such as MQTT, AMQP, CoAP, STOMP, XMPP, Zigbee, Z-Wave, MTConnect, OPC UA, ANT+, NFC, RFID, Bluetooth, DASH7, 6LoWPAN, AllJoyn, Thread, or Weightless. By applying ABE schemes, we are convinced that security can be increased and advantages, such as grouping sensors that share a certain characteristic and apply attributes to them for a (brokered) end-to-end encryption approach, are feasible.

Singh et al. [9] provide an approach combining MQTT and ABE. They introduce an ABE-secured MQTT and MQTT for sensor networks protocol. A trusted third party, the MQTT broker, generates the secret master key and the access policies. Their approach requires a different trust relation of entities compared to the previous one. Touati et al. [10] introduce an efficient attribute management for CP-ABE. They focus on revocation of attributes, thus access rights, which is achieved without the need of a third party (e.g. a proxy).

There are use cases within the healthcare data domain that are based on ABE ciphertext-policies to protect electronic health records (EHR), namely Wang et al. [11], Alshehri et al. [12] and Akinyele et al. [13]. Within the project Curcuma[1] ABE is deployed in a location-based access control scenario whereas the location information is encoded inside the ABE attributes, see Zickau et al. [14] and Denisow et al. [3]. There exist solutions where ABE is applied within publish subscribe messaging-patterns (pub/sub-MP) [2]. In the related work from Picazo-Sanchez et al. [15] ABE is utilized to access medical sensor and devices data and to change the settings of sensors. The work describes a pub/sub environment. It also deploys an authority, called *key generation center* (KGC). It distributes the public keys, holds the secret key, generates the private keys for the users, and is responsible for the key distribution. The work describes sensors, which encrypt the data. They communicate with a so-called *policy service* to receive the access structure for the encryption. The policy service is an entity, which checks the rights of the user and generates the access structures for the ABE encryption. The access structure becomes then part of an access policy. The access policy has additional information, such as a time stamp, and is valid for a given period.

In Yang et al. [16] an overall encryption via ABE for private data in a pub/sub environment is discussed. The pub/sub-MP is represented by a cloud big data storage system. Publishers and subscribers depend on cloud storage to exchange confidential data. For example, in such a data-exchange environment, publishers who publish the data could be banks, investment firms or research institutions. Data-subscribers are normal users, e.g., business or financial analysts, who receive the data by submitting their subscribed trapdoors. Their two goals are *data privacy* of the cloud environment, i.e., access restrictions for unauthorized users and *subscription privacy* of the cloud environment, i.e., information restrictions about the interests of the users from the subscribed trapdoors. The work introduces an ABE variant called bi-policy attribute-based encryption (BP-ABE). For the cloud-storage, it is possible to select which users have a subscription to a topic. The so-called privacy preserving BP-ABE matching makes sure a cloud server is able to match the subscriptions of the users and additionally it also checks if the users have the required attributes to decrypt the data. The content of the subscription trapdoors (the topics) of the users remain encrypted for the cloud server. The BP-ABE combines a two-level policy for the encrypted data, the access policy to the data, and subscription policy for the trapdoors topics. While the access policy is constructed of attributes, the subscription policy is constructed of data tags. The real interests (topics) are hidden behind these tags.

Cachet [17] is a 'Decentralized Architecture for Privacy Preserving Social Networking with Caching'. Cachet aims to enable confidentiality, integrity and availability of user content, as well as the privacy of user relationships in online social networks (OSN). Based on a distributed hash table (DHT), Cachet relies on a distributed pool of nodes to store and ensure availability. Furthermore, a hybrid structured and unstructured overlay concept is applied, in which a DHT is augmented with social links between users. In EASiER [18], an access control architecture for OSNs is described. Based on Bethencourt's CP-ABE implementation, Jahid et al. provides a revocation scheme by introducing a minimally trusted proxy and an integration of group communication schemes with ABE. A full decentralized, peer-to-peer OSN is introduced in [19]. Jahid et al.'s goal is to protect the integrity, availability and confidentiality of user content and the privacy of user relationships by dappling CP-ABE and a DHT for availability of user data and cryptographic functions.

Thatmann et al. [20] introduced the notion of ciphertext expiration, which supports temporal storage of cryptographic keys in a DHT in a secure manner. They provide concrete algorithms and an implementation deploying the appropriate (temporal) secret attribute keys to target users. In addition an insight view of a generic and user-centric file encryption and exchange service is presented. Use cases related to sharing personal health or fitness data with physicians and OSN-related scenarios, such as sharing picture galleries protected by ABE, are presented.

V. Implementations

There are multiple implementations for ABE. The following subsections provide an overview of ABE schemes with corresponding implementations circulating on the Internet. The schemes are described regarding programming language, expressiveness, performance, and other details.

Table I provides an overview on the details of all tested ABE implementations. Next to the developers (authors) and the implemented schemes, the table shows the programming language, the library their implementations are based on, the open source license and when it was last updated.

Since the goals of the related projects were to write prototype applications for mobile devices and low-end computers, such as the Raspberry Pi, it was necessary to check if these implementations can be deployed there. Other criteria for choosing the libraries include the available features, the extensibility, and the general ease of installation and usage.

A. cpabe

cpabe [5] was the first implementation of an CP-ABE scheme. It was written by John Bethencourt in 2007. For

[1] http://curcuma-project.net/

TABLE I
ABE IMPLEMENTATION LIBRARIES OVERVIEW

Section	Library Name / Implementation	Implementation Author(s)	Scheme(s)	Language	Based on Library	License	Updated
V-A	cpabe [21]	John Bethencourt	BSW07 [5]	C	PBC	GPLv2	03/2011
V-B	libfenc [22]	Matthew Green, Joseph Ayo Akinyele	LSW08 [23], Waters11 [24], GPSW07 [4]	C++	PBC	GPLv2	02/2011
V-C	Charm [25]	Joseph Ayo Akinyele	BSW07 [5], LSW08 [23], Waters11 [24]	Python	PBC, MIRACL or RELIC	LGPLv3	07/2013
		Gary Belvin	LW11 [26]			LGPLv3	11/2013
		Artjom Butyrtschik	LWW14 [27], YCT14 [28], YJ13 [29], YJ14b [30]			LGPLv3	07/2015
		Alexander Förster	YAHK14 [31]			LGPLv3	07/2015
		Yannis Rouselakis	RW15 [32], LW12 ([33]) (KP), OT12 [34] (KP+CP), RW12 [35] (KP+CP)			Unlicensed	11/2012
V-D	cpabe (Java) [36]	Junwei Wang	BSW07 [5]	Java	JPBC	GPLv2	03/2015
V-E	JCPABE [37]	Iwailo Denisow	BSW07 [5]	Java	JPBC	GPLv2	06/2015
V-F	jTR-ABE [38]	Artjom Butyrtschik	LW14 [39]	Java	JPBC	GPLv2	09/2015
V-G	KPABE [40]	Yao Zheng	GPSW07 [4]	C	PBC	GPLv2	11/2014
V-H	DCPABE [41]	Stefano Braghin	LW11 [26]	Java	JPBC	Unlicensed	11/2012
V-I	DET-ABE [42]	Miguel Morales-Sandoval	BSW07 [5]	Java	JPBC	Public Domain	06/2015
V-J	PIRATTE [43]	Sonia Jahid	JB12 [6]	C	PBC	GPLv2	04/2013
V-K	arcanum [44]	Angelo De Caro	GVW13 [45], GGHVV13 [46], GGHSW13 [47], BNS13 [48]	Java	arcanum-pbc	LGPL v3	04/2015
V-L	LSSS2 [49]	Eric Zavattoni et al.	Waters11 [24]	C++	ate-pairing	3-clause BSD	10/2013
V-M	NEON ABE [50]	Ana Helena Sánchez	Waters11 [24]	C++	-	Public domain	02/2013
V-N	AndrABEn [51]	Moreno Ambrosin, Mauro Conti, Tooska Dargahi	BSW07 [5], GPSW07 [4]	Java	PBC	GPLv2	10/2014

its calculations it depends on the pairing-based cryptography library [52]. While it only supports CP-ABE, it features a parser to transform the Boolean formulas into an internal format. This implementation has also support for numerical attributes in the private key and their comparison operators in the encryption policies.

Besides supporting *and* and *or* cpabe supports the *of* operator. This implementation has been written in C and required some workarounds to compile. Attempts to run this on Android have failed, largely due to requiring dependencies that would also have to be modified/compiled for Android.

The software package consists of two parts, libbswabe contains the four basic operations of the scheme: setup, keygen, encrypt and decrypt, which can only encrypt and decrypt group elements. cpabe contains the command line tools to run the libbswabe operations on actual data utilizing a hybrid encryption approach with AES-128 in Cipher Block Chaining (CBC) mode provided by OpenSSL.

B. libfenc

libfenc [53] is a library for functional encryption schemes. It has been written in 2010 by Akinyele et al. and supports a CP-ABE and a KP-ABE scheme. The library reuses the code of the parser from Bethencourt's implementation, and thus also supports numerical attributes. libfenc was written in C/C++, making it difficult to run on Android, but it has been successfully tested on mobile devices, as can be seen in [13], where libfenc encrypts and decrypts data on an iOS device. Because there have been no code changes since 2011 this project is most likely abandoned.

C. Charm

The follow-up project from libfenc is Charm. Charm [54] is a framework for prototyping cryptosystems. It has been in development since 2011. It features implementations of various ABE schemes, amongst others a reimplementation of Bethencourt's scheme. There is also an implementation of a KP-ABE scheme. The major drawback for this library is that, while there is a parser for policies, it does not support numerical attributes. The parser also does not handle the *of* operator. ABE is based on elliptic curve cryptography. The abstraction that is implemented in Charm enables the usage of different curves for every scheme. Charm has been written in Python. A package for Android phones is offered.

All schemes, except YCT14, only provide the basic ABE construct of encrypting a random group element. This random element can then be hashed in order to create the key bytes for a symmetric block cipher. This so-called key encapsulation mechanism can also be employed in Charm and enables all of the schemes to be deployed in real world applications.

1) Schemes overview: The implemented schemes can be divided in 3 groups: plain ABE without any additional functionality as shown in Section III, multi-authority schemes and proxy re-encryption schemes.

The schemes BSW07 [5], LSW08 [23], Waters11 [24], YCT14 [28], RW12 [35], OT12 [34], LW12 [33] and YAHK14 [31] implement the four basic algorithms of setup, keygen, encrypt and decrypt, using different techniques. LSW08 does not fully implement the scheme from the article it is based on, because revocation or negated attributes are not compatible with the standard *ABEnc* superclass. The YCT14 scheme is the only one without pairing and is therefore the fastest scheme under the decisional Diffie-Hellman assumption over elliptic curves (ECDDH). Both RW12 and OT12 describe CP-ABE as well as KP-ABE constructions for which there are implementations. LW12 describes different CP-ABE constructions, but the scheme implementation follows the KP-ABE model with the same techniques. YAHK14 provides a non-monotonic and unbounded CP-ABE scheme. The attributes are represented as integers in $\mathbb{Z}_p$.

The schemes in LW11 [26], RW15 [32], YJ13 [29] and YJ14 [30] provide a multi-authority setting. The YJ13 and YJ14 implementations offer user attribute revocation by updating the attribute secret component for all non-revoked users. LW11 and RW15 are the only schemes that do not have a central authority. The attribute authorities are autonomous in these two implementations, but they rely on global public parameters.

The implementations LWW14 [27] and YJ13 [29] provide proxy re-encryption for partially different use cases. In LWW14 the proxy transforms the ciphertext for time locking the decryption to (multiple) time spans with a granularity of one day. Only users that have the matching time-based attribute secret keys can decrypt the ciphertexts that are produced from the proxy. In YJ13 the proxy re-encrypts the ciphertext relieving the client from performing the pairing operation. The client only needs to do one exponentiation to recover the ABE-ciphertext.

2) Benchmark: The schemes that are implemented in Charm are very diverse. They have different feature sets and are based on different security assumptions. The benchmark results in Figures 1 and 2 show the total time of a full cycle (setup, keygen, encrypt, decrypt) and include the detailed time for each phase. Some schemes have additional phases such as Authority Setup (*authsetup*) for the setup phase of attribute authorities in the multi-authority setting and *re-encrypt* for schemes, which feature some kind of re-encryption proxy. The results show a cumulative execution in seconds of 100 iterations of full cycles for each scheme. Table II contains the results for Figure 1. The encryption was done with a single policy: *THREE and (ONE or TWO)*. The authsetup phase is necessary within multi-authority schemes. That phase can be regarded as part of the setup phase for schemes, which have only a single authority. This can be seen when comparing the BSW07 and LW11 graphs. A non-monotonic scheme such as YAHK14 needs to setup more components during keygen, because the negation of attributes has to be represented. This can be seen in the graph where the keygen phase is nearly twice as slow when compared to other single-authority and proxy-less schemes. This is still more efficient than the naive approach of adding every negated attribute as its own attribute to the user secret keys.

YCT14 is the fastest scheme, because it does not need bilinear pairings, which is the overall slowest operation. YJ13 employs the proxy re-encryption specifically to relieve the client of the decryption burden by producing ElGamal-style ciphertexts. The decryption requires only a single exponentiation and is therefore very fast compared to the other schemes.

Dual Pairing Vector Spaces (DPVS) is a technique for LW12 and OT12 to simulate orthogonal subgroups, which appear in composite order groups as a basis for ABE. The three implementations that apply DPVS are roughly 10 times slower than comparable schemes that are based on prime-order groups.

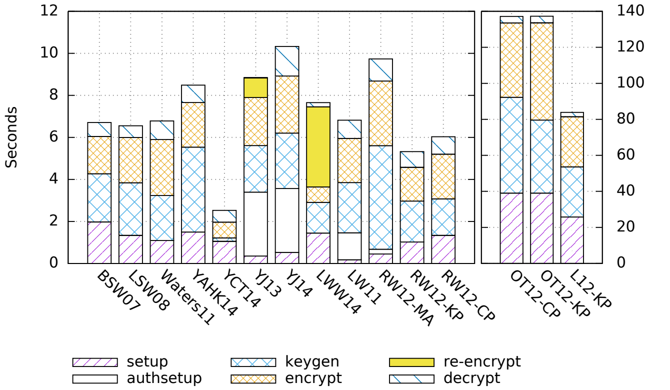

Fig. 1. Charm Benchmark on a i7-4710HQ @ 3.40 GHz (cumulative time for 100 iterations)

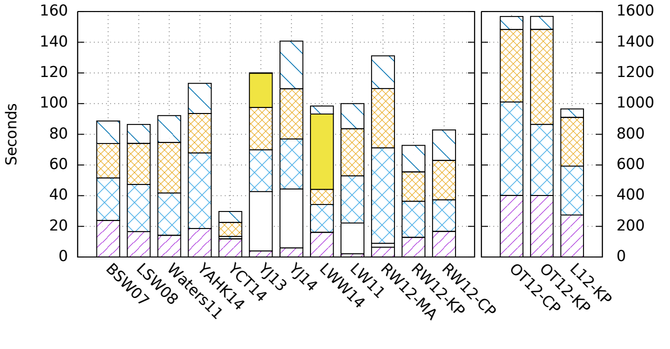

Fig. 2. Charm Benchmark on a Raspberry Pi 2 @ 900 MHz (cumulative 100 iterations)

3) Charm details: All of the implemented ABE schemes are based on bilinear pairing, which is defined over elliptic curves. Charm supports the three pairing backends: Pairing-

TABLE II
CHARM BENCHMARK ON A I7-4710HQ @ 3.40GHZ (CUMULATIVE TIME FOR 100 ITERATIONS)

Scheme	Setup	Auth-Setup	Keygen	Enc	ReEnc	Dec
BSW07	1.979	0.000	2.289	1.775	0.000	0.661
LSW08	1.338	0.000	2.501	2.154	0.000	0.559
Waters11	1.092	0.000	2.141	2.665	0.000	0.885
YAHK14	1.496	0.000	4.037	2.123	0.000	0.830
YCT14	1.056	0.000	0.161	0.749	0.000	0.557
YJ13	0.355	3.038	2.217	2.286	0.934	0.014
YJ14	0.526	3.037	2.638	2.716	0.000	1.408
LWW14	1.447	0.000	1.458	0.733	3.813	0.200
LW11	0.177	1.283	2.386	2.103	0.000	0.868
RW12-MA	0.450	0.227	4.929	3.072	0.000	1.053
RW12-KP	1.023	0.000	1.947	1.608	0.000	0.741
RW12-CP	1.340	0.000	1.733	2.130	0.000	0.828
OT12-CP	36.141	0.000	49.184	38.233	0.000	3.324
OT12-KP	36.116	0.000	37.522	50.019	0.000	3.322
L12-KP	23.854	0.000	25.754	25.772	0.000	2.193

based Cryptography (PBC)[2] (default), the MIRACL[3] cryptographic library and the RELIC[4] cryptographic meta-toolkit. Each of which brings its own elliptic curves, some of which are the Type A symmetric pairing over super singular curve SS512 or the Type D asymmetric pairing MNT224. The schemes LW11 [26], LWW14 [27], YJ13 [29] and YJ14 [30] only support symmetric pairings.

The security proof for LW11 is only available for groups of composite order instead of prime order. Since Charm does not provide easy access to groups of composite order, the scheme was implemented over groups of prime order which has unknown security bounds and should be strictly regarded as prototype implementations.

D. cpabe (Java)

A Java version of cpabe has been written in 2013 by Junwei Wang [55] as part of a thesis. It is a direct port of the implementation from John Bethencourt's cpabe. This became apparent, because it is based on the same internal structure, and rarely takes advantage of Java's object-oriented paradigm. This implementation applies the same internal format of policies that the C implementation introduced. Since there is no policy parser, there is no way to conveniently enter policies. Instead the user is required to manually enter the reverse polish notation of the tree of the intended policy [55]. The only dependency of the library is the JPBC [56], which is needed for the cryptographic primitives within the ABE scheme. It is possible to run cpabe (Java) on any machine running Java, including Android.

E. JCPABE

JCPABE is based on cpabe (Java). It adds to and improves the existing functionality. A policy parser has been added, so that policies and numerical attributes can be entered in the same way as with the Bethencourt version of cpabe. The parser has been written with JavaCC and JJTree. Being based on cpabe (Java), JCPABE inherits the platform independence. No new dependencies have been added. The library has been tested to run on Android devices. The additions to the functionality are described in [3]. They include the generation and appending of attributes to an existing private key as well as an experimental feature that enables dynamic location information within policies.

F. jTR-ABE

jTR-ABE is based on an early version of JCPABE. It also supports expressive policies with numerical attributes and threshold gates in the same way JCPABE does. The implemented LW14 scheme enables public blackbox traitor tracing. It is possible to determine the users, who shared their private keys to create a decryption blackbox, by using special iterated encryption and decryption challenges that can be executed publicly, outside of the trusted central authority. For this scheme the maximum number of unique users throughout the lifetime of the system needs to be specified at the setup phase.

G. KPABE

Yao Zheng implemented the GPSW06 [4] scheme, which was the first published KP-ABE scheme, and applied it to personal health records (PHR) for privacy-preservation [57]. The implementation is derived from the Bethencourt implementation of CP-ABE [5]. It is also divided up into the ABE library and a set of command line tools. The policy syntax supports conjunctions, disjunctions and general threshold gates, but lacks support for numerical attributes.

H. DCPABE

Stefano Braghin implemented the decentralized CP-ABE scheme LW11 [26]. It is based on a linear secret sharing scheme (LSSS) for applying the policy and supports conjunctions and disjunctions. Numerical attributes are not available.

I. DET-ABE

Miguel Morales-Sandoval implemented DET-ABE [42], which is another realization of the BSW07 scheme [5]. It supports three different security levels by selecting appropriate AES key sizes (128, 192 or 256 bit) and elliptic curves for bilinear pairings (Type A of 256 bit size or Type F of 384 / 512 bit sizes). The library currently supports threshold policies in postfix notation without explicit *and* and *or* operators or numerical attributes. The library class model represents a framework, because the ABE scheme implementation is encapsulated into different components such as TrustedAuthority or DETABECipher. Those components cannot be directly deployed, because they lack the standalone functionality as a server (e.g., servlet) or client (e.g., command-line tool).

[2] https://crypto.stanford.edu/pbc/

[3] https://github.com/CertiVox/MIRACL

[4] https://github.com/relic-toolkit/relic

J. PIRATTE

In [6] Jahid et al. introduce a proxy-based immediate revocation scheme and implementation [43] based on Bethencourt's CP-ABE scheme and source code, closer described in V-A. The minimally trusted PIRATTE Proxy handles user and attribute revocation. In a case study, PIRATTE is the underlying cryptography scheme in EASiER, an access control architecture for Online Social Networks (OSN) and in DECENT, a decentralized architecture for OSNs. The EASiER implementation consists of a Facebook application demonstrating ABE.

K. arcanum

Arcanum, the successor of JPBC, is a Java library, which implements four ABE schemes that are based on Garbled or Arithmetic Circuits built on top of Lattices or Multilinear Maps: GVW13 [45], GGHVV13 [46], GGHSW13 [47], BNS13 [48]. GVW13 [45] additionally utilizes a Two-to-One Recoding scheme. The library is built modularly supporting many primitives.

L. LSSS2

The LSSS2 project implements the large universe construction of Waters11 [24]. The associated article [58] describes efficient pairing algorithms and other optimizations for multiplication and exponentiation. It is based on the ate-pairing library, which applies the pairing-friendly Barreto-Naehrig curves. The software package requires Xbyak[5] – a just-in-time (JIT) assembler – and it is only supported on 64-bit systems. LSSS2 only supports encrypting group elements, but could be extended with a hybrid encryption approach to encrypt arbitrary data.

M. NEON

NEON is a single instruction multiple data (SIMD) set for ARM processors, which can be used to implement a fast ABE scheme. Sánchez et al. [59] have proposed such an implementation based on the Waters11 [24] CP-ABE scheme. The implementation is very rudimentary, i.e., it does not provide policy parsing or even secret sharing. It can be built for NEON-based chips, but the software package does not contain any build instructions. We could successfully execute the included benchmark code with a few tweaks on a Raspberry Pi 2 with the ARMv7 architecture, which provides the NEON instruction set. It lacks an example implementation of the full encryption/decryption cycle, which prevents verification whether the ABE scheme is implemented correctly.

N. AndrABEn

The authors Ambrosin et al. designed the AndrABEn library specifically for Android devices [60]. It is based on the cpabe and KPABE C libraries and implements the BSW07 and GPSW07 schemes. As it includes libbswabe it should be licensed under the GPLv2. The related paper focuses on the performance of the implementation tested on Android devices.

VI. Conclusion and Outlook

This publication provides an overview of applied ABE schemes and the information about their implemented libraries. Additionally publication describes real world use cases and the fundamentals of ABE. Furthermore, it covers the extensions the authors made during the process of using ABE in applied scenarios. As there is a variety of ABE schemes published within the mathematical and theoretical research community another finding of the research is that only few of them are actually published as usable up-to-date and supervised libraries or even implemented by the authors themselves. We believe that ABE is a viable and interesting contribution to applied cryptography in general, but the lack of implementations should be considered as a motivation for computer scientists in the present and future.

Acknowledgment

The work presented in this publication was performed in the context of the projects *Curcuma*[6] and *Entrance*[7] within the Software Campus program. The projects are funded by the German Federal Ministry of Education and Research (BMBF) under grant number 01IS12056.

References

[1] D. Boneh, A. Sahai, and B. Waters, "Functional encryption: a new vision for public-key cryptography," *Communications of the ACM*, vol. 55, no. 11, 2012.

[2] D. Thatmann, S. Zickau, A. Förster, and A. Küpper, "Applying attribute-based encryption on publish subscribe messaging patterns for the internet of things," in *Proceedings of the 8th IEEE International Conference on Internet of Things (iThings 2015)*. IEEE, 2015.

[3] I. Denisow, S. Zickau, F. Beierle, and A. Küpper, "Dynamic location information in attribute-based encryption schemes," in *Proceedings of the 9th International Conference on Next Generation Mobile Applications, Services and Technologies (NGMAST 2015)*. IEEE, 2015.

[4] V. Goyal, O. Pandey, A. Sahai, and B. Waters, "Attribute-based encryption for fine-grained access control of encrypted data," in *Proceedings of the 13th ACM conference on Computer and communications security*. ACM, 2006.

[5] J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-policy attribute-based encryption," in *Security and Privacy, 2007. SP'07. IEEE Symposium on*. IEEE, 2007.

[6] S. Jahid and N. Borisov, "PIRATTE: Proxy-based immediate revocation of attribute-based encryption," *arXiv preprint arXiv:1208.4877*, 2012.

[7] A. Sahai, H. Seyalioglu, and B. Waters, "Dynamic credentials and ciphertext delegation for attribute-based encryption," in *Advances in Cryptology–CRYPTO 2012*. Springer, 2012.

[8] R. Cramer and V. Shoup, "Design and analysis of practical public-key encryption schemes secure against adaptive chosen ciphertext attack," *SIAM Journal on Computing*, vol. 33, no. 1, 2003.

[9] M. Singh, M. Rajan, V. Shivraj, and P. Balamuralidhar, "Secure mqtt for internet of things (iot)," in *Communication Systems and Network Technologies (CSNT), 2015 5th Int. Conf. on*, April 2015.

[10] L. Touati and Y. Challal, "Efficient cp-abe attribute/key management for iot applications," in *Computer and Information Technology (CIT), 2015 IEEE International Conference on*. IEEE, 2015.

[11] C. Wang, X. Liu, and W. Li, "Implementing a personal health record cloud platform using ciphertext-policy attribute-based encryption." in *INCoS*, 2012.

[12] S. Alshehri, S. P. Radziszowski, and R. K. Raj, "Secure access for healthcare data in the cloud using ciphertext-policy attribute-based encryption," in *Data Engineering Workshops (ICDEW), 2012 IEEE 28th International Conference on*. IEEE, 2012.

[5] https://github.com/herumi/xbyak

[6] http://curcuma-project.net/

[7] http://entrance.snet.tu-berlin.de/

[13] J. A. Akinyele, M. W. Pagano, M. D. Green, C. U. Lehmann, Z. N. Peterson, and A. D. Rubin, "Securing electronic medical records using attribute-based encryption on mobile devices," in *Proceedings of the 1st ACM workshop on Security and privacy in smartphones and mobile devices*. ACM, 2011.

[14] S. Zickau, F. Beierle, and I. Denisow, "Securing mobile cloud data with personalized attribute-based meta information," in *Proceedings of the 3rd IEEE International Conference on Mobile Cloud Computing, Services, and Engineering*. IEEE, 2015.

[15] P. Picazo-Sanchez, J. E. Tapiador, P. Peris-Lopez, and G. Suarez-Tangil, "Secure publish-subscribe protocols for heterogeneous medical wireless body area networks," *Sensors*, vol. 14, no. 12, 2014.

[16] K. Yang, X. Jia, K. Zhang, and X. S. Shen, "Privacy-preserving data publish-subscribe service on cloud-based platforms."

[17] S. Jahid, S. Nilizadeh, P. Mittal, N. Borisov, and A. Kapadia, "Cachet: A Decentralized Architecture for Privacy Preserving Social Networking with Caching," in *Proceedings of the 8th Int. Conf. on Emerging Networking Experiments and Technologies*, ser. CoNEXT. ACM, 2012.

[18] S. Jahid, P. Mittal, and N. Borisov, "EASiER: Encryption-based Access Control in Social Networks with Efficient Revocation," in *Proceedings of the 6th ACM Symposium on Information, Computer and Communications Security*, ser. ASIACCS '11. New York, NY, USA: ACM, 2011.

[19] S. Jahid, S. Nilizadeh, P. Mittal, N. Borisov, and A. Kapadia, "DECENT: A decentralized architecture for enforcing privacy in online social networks," in *Pervasive Computing and Communications Workshops (PERCOM), 2012 IEEE Int. Conf. on*, March 2012.

[20] D. Thatmann, A. Butyrtschik, and A. Küpper, "A Secure DHT-based Key Distribution System for Attribute-based Encryption and Decryption," in *Proceedings of the 9th Intl. Conference on Signal Processing and Communication Systems (ICSPCS 2015)*. Cairns, AU: IEEE, Dec. 2015.

[21] J. Bethencourt, "cpabe." [Online]. Available: http://hms.isi.jhu.edu/acsc/cpabe/

[22] M. Green and J. A. Akinyele, "libfenc." [Online]. Available: https://code.google.com/p/libfenc/

[23] A. Lewko, A. Sahai, and B. Waters, "Revocation systems with very small private keys," Cryptology ePrint Archive, Report 2008/309, 2008.

[24] B. Waters, "Ciphertext-policy attribute-based encryption: An expressive, efficient, and provably secure realization," in *Public Key Cryptography – PKC 2011*, ser. Lecture Notes in Computer Science, D. Catalano, N. Fazio, R. Gennaro, and A. Nicolosi, Eds. Springer Berlin Heidelberg, 2011, vol. 6571.

[25] "Charm." [Online]. Available: https://code.google.com/p/charm-crypto/

[26] A. Lewko and B. Waters, "Decentralizing attribute-based encryption," in *Advances in Cryptology - EUROCRYPT 2011*, ser. Lecture Notes in Computer Science, K. Paterson, Ed. Springer Berlin, 2011, vol. 6632.

[27] Q. Liu, G. Wang, and J. Wu, "Time-based proxy re-encryption scheme for secure data sharing in a cloud environment," *Information Sciences*, vol. 258, no. 0, 2014.

[28] X. Yao, Z. Chen, and Y. Tian, "A lightweight attribute-based encryption scheme for the internet of things," *Future Generation Computer Systems*, vol. 49, no. 0, 2015.

[29] K. Yang, X. Jia, K. Ren, B. Zhang, and R. Xie, "DAC-MACS: Effective Data Access Control for Multiauthority Cloud Storage Systems," *Information Forensics and Security, IEEE Transactions on*, vol. 8, no. 11, Nov 2013.

[30] K. Yang and X. Jia, "Expressive, efficient, and revocable data access control for multi-authority cloud storage," *Parallel and Distributed Systems, IEEE Transactions on*, vol. 25, no. 7, July 2014.

[31] S. Yamada, N. Attrapadung, G. Hanaoka, and N. Kunihiro, "A framework and compact constructions for non-monotonic attribute-based encryption," in *Public-Key Cryptography – PKC 2014*, ser. Lecture Notes in Computer Science, H. Krawczyk, Ed. Springer Berlin Heidelberg, 2014, vol. 8383.

[32] Y. Rouselakis and B. Waters, "Efficient statically-secure large-universe multi-authority attribute-based encryption," in *Financial Cryptography and Data Security*, ser. Lecture Notes in Computer Science. Springer Berlin, 2015.

[33] A. Lewko and B. Waters, "New proof methods for attribute-based encryption: Achieving full security through selective techniques," Cryptology ePrint Archive, Report 2012/326, 2012. [Online]. Available: http://eprint.iacr.org/2012/326

[34] T. Okamoto and K. Takashima, "Fully secure unbounded inner-product and attribute-based encryption," Cryptology ePrint Archive, Report 2012/671, 2012.

[35] Y. Rouselakis and B. Waters, "New constructions and proof methods for large universe attribute-based encryption," Cryptology ePrint Archive, Report 2012/583, 2012.

[36] J. Wang, "cpabe (java)." [Online]. Available: https://github.com/junwei-wang/cpabe

[37] I. Denisow, "Jcpabe." [Online]. Available: https://github.com/TU-Berlin-SNET/JCPABE

[38] A. Butyrtschik, "jtr-abe." [Online]. Available: https://github.com/TU-Berlin-SNET/jTR-ABE

[39] Z. Liu and D. S. Wong, "Practical Attribute Based Encryption: Traitor Tracing, Revocation, and Large Universe," *IACR Cryptology ePrint Archive*, 2014.

[40] Y. Zheng, "Kpabe." [Online]. Available: https://github.com/gustybear/kpabe

[41] S. Braghin, "Dcpabe." [Online]. Available: https://github.com/stefano81/dcpabe

[42] M. Morales-Sandoval and A. Diaz-Perez, "Det-abe: A java api for data confidentiality and fine-grained access control from attribute based encryption," in *Information Security Theory and Practice*, ser. Lecture Notes in Computer Science. Springer International Publishing, 2015.

[43] S. Jahid, "Piratte," 2013. [Online]. Available: https://bitbucket.org/hatswitch/piratte

[44] A. D. Caro, "arcanum." [Online]. Available: https://github.com/adecaro/arcanum

[45] S. Gorbunov, V. Vaikuntanathan, and H. Wee, "Attribute-based encryption for circuits," in *Proceedings of the Forty-fifth Annual ACM Symposium on Theory of Computing*, ser. STOC '13. New York, NY, USA: ACM, 2013.

[46] C. Gentry, S. Gorbunov, S. Halevi, V. Vaikuntanathan, and D. Vinayagamurthy, "How to compress (reusable) garbled circuits," Cryptology ePrint Archive, Report 2013/687, 2013.

[47] S. Garg, C. Gentry, S. Halevi, A. Sahai, and B. Waters, "Attribute-based encryption for circuits from multilinear maps," Cryptology ePrint Archive, Report 2013/128, 2013.

[48] D. Boneh, V. Nikolaenko, and G. Segev, "Attribute-based encryption for arithmetic circuits," Cryptology ePrint Archive, Report 2013/669, 2013.

[49] E. Zavattoni, "Lsss2." [Online]. Available: https://bitbucket.org/herumi/lsss2/src

[50] A. H. Sánchez, "Neon abe." [Online]. Available: http://delta.cs.cinvestav.mx/~francisco/codigo.html

[51] M. Ambrosin, M. Conti, and T. Dargahi, "Andraben," 2014. [Online]. Available: http://spritz.math.unipd.it/projects/andraben/files/

[52] B. Lynn, "On the implementation of pairing-based cryptosystems," Ph.D. dissertation, Stanford University, 2007.

[53] M. Green, J. A. Akinyele, and M. Rushanan, "libfenc: The Functional Encryption library." [Online]. Available: http://code.google.com/p/libfenc

[54] J. A. Akinyele, C. Garman, I. Miers, M. W. Pagano, M. Rushanan, M. Green, and A. D. Rubin, "Charm: A framework for rapidly prototyping cryptosystems," *Journal of Cryptographic Engineering*, vol. 3, no. 2, 2013.

[55] J. Wang, "cpabe (Java)," 2013. [Online]. Available: https://github.com/junwei-wang/cpabe

[56] A. De Caro and V. Iovino, "jPBC: Java pairing based cryptography," in *Proceedings of the 16th IEEE Symposium on Computers and Communications, ISCC 2011*. IEEE, 2011. [Online]. Available: http://gas.dia.unisa.it/projects/jpbc/

[57] Y. Zheng, "Privacy-preserving personal health record system using attribute-based encryption," Ph.D. dissertation, Worcester Polytechnic Institute, 2011.

[58] E. Zavattoni, L. Dominguez Perez, S. Mitsunari, A. Sanchez-Ramirez, T. Teruya, and F. Rodriguez-Henriquez, "Software implementation of an attribute-based encryption scheme," *Computers, IEEE Transactions on*, vol. 64, no. 5, May 2015.

[59] A. Sanchez and F. Rodriguez-Henriquez, "NEON implementation of an attribute-based encryption scheme," in *Applied Cryptography and Network Security*, ser. Lecture Notes in Computer Science, M. Jacobson, M. Locasto, P. Mohassel, and R. Safavi-Naini, Eds. Springer Berlin Heidelberg, 2013, vol. 7954.

[60] M. Ambrosin, M. Conti, and T. Dargahi, "On the feasibility of attribute-based encryption on smartphone devices," *arXiv preprint arXiv:1504.00619*, 2015.