A Rivest Shamir Adleman Approach to Attribute Based Security System

L.A. Nwosu, B.K. Alese, A.F.Thompson, O.O. Obe
Department of Computer Science
Federal University of Technology
Akure, Nigeria

Abstract

Traditional attribute architectures and cryptosystems are ill-equipped to provide security in the face of diverse access requirements and environments. The glitches of confidentiality, authenticity and anonymity induced by the exponential upsurge in digital technology have generated some security concerns, compelling security parameters review for enhanced security. Consequently, in this paper, an ABS was constructed based on groups with bilinear pairings with a powerful feature that it can be readily used in a multi-authority setting. The developed system overcomes the shortcomings of other schemes in the standard model. Implementations of RSA Encryption schemes were done with C#; the scheme was compared with earlier constructed schemes to verify its performance. I was observed that these processes: signing, encryption, decryption and verification proved more efficient when compared to existing schemes using standard parameters.

1. Introduction

Attribute-based signature (ABS) system is a cryptographic system that is employed in environments where the users roles depend on the possess attributes combination. In such systems, one or more attribute authorities are obtained by users multiple attributes with user's capabilities in accessing resources based on inherent attributes. The cryptographic system authentication of signature does not rely on individual identity that signs a message, rather it is about the attributes the signer obtained from an attributes authority. Unlike in traditional cryptography where the intended recipient or signer's identity is clearly known, in an attribute based systems users need only specify the attributes or identities of the recipients or the signer in form of a predicate that is to be satisfied. This feature enables secure data sharing even in a decentralized setting, providing both fine-grained control on access and some degree of anonymity for the participants.

Anonymity means that no one can tell who generates the signature among the users with the attribute sets A, even if many signatures from the same signature using the same subset of attributes are provided.

Attribute-based signature offers an unforge-ability guarantee for the verifier, that the signature was produce by a single party whose attributes satisfy a certain predicate or policy being made; unforge-ability means that no one can forge the signature with the attribute sets without being issued the certificate for the attribute sets. Besides these two semantics, another important property of it is called collusion resistance which means that different parties are not allowed to pool together their attributes to sign a message without the satisfaction of the two. For instance, if two or more users possessing different keys combine to decrypt the cipher text, they will be successful if and only one of the users could have decrypted it individually. This feature enables secure data sharing even in a decentralized setting, providing both fine-grained control on access and some degree of anonymity for the participants.

2. Fundamentals of Cryptography

The term cryptography was derived from the Greek words "Kryptos" and "Graphein" meaning "Hidden" and "to write" in English language. The word 'Cryptography' occurs in an article of Sir Thomas Browne's Discourse (1658) titled: "the strange cryptography of Gaffarel in his starrie Brooke of Heaven". Cryptography has been in existence for thousands of years ago. In the 20th century, the complex mechanical and electrochemical machines were invented: the Enigma rotor machine. These are entirely unsuited to pen and paper [1]. Perfect security is not available at any price. Rather, the risk and probability of violations can be reduced to acceptable levels only. The cost of the security measures, in fact, must be in balance with the risks to the system, and provision must be made for the possibility of recovery after a security violation has actually occurred. Cryptography is the basis for all secured communications [2].

Cryptography is concerned with linguistics and mathematical techniques for securing information in communications [3]. Conversely, cryptography has broadened its horizon, cryptography provides a various applications including authentication, digital signatures, digital cash, electronic voting and so on [4].

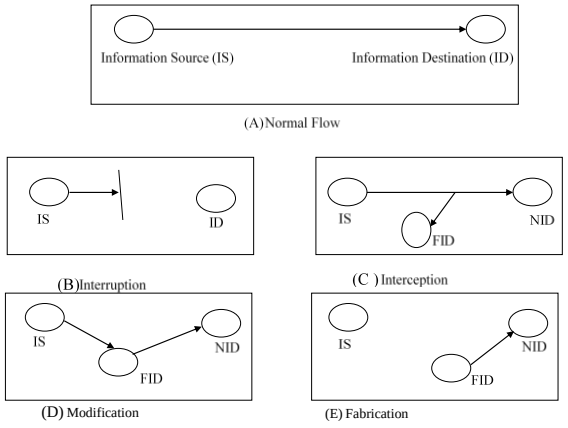
In Meneze et al, [5], cryptography is described as the use of mathematical techniques in relation to information security such as, data integrity, confidentiality non-repudiation and so on. Data integrity is a service, which addresses the authorized alteration of data. While authentication implies entering into communication should be authenticated as to origin, date of origin, time sent, data content and so on. Confidentially is a service attached to the content of information from all but those authorized. Non-repudiation is a service, which prevents an entity from denying previous commitments or actions. When disputes arise due to an entity denying that certain actions were taken, a means to resolve the situation is necessary.

Most algorithms undergo years of scrutiny by the world's best cryptographers who validate the strength of the algorithm. Publicizing the algorithm, the cryptographers get free consulting from a large number of academic cryptologist who are eager to break the system so that they can publish papers demonstrating their smartness. If many experts have tried to break the algorithm for five years after its publication without success, such algorithm is probably solid [6].

In public key cryptography, the algorithms are widely known and available; it is the key that is kept secret and provides the required security. The key is analogous to the combination lock. Although the concept of combination lock is well known, you can't open a combination lock easily without knowing the combinations. In addition to this, the more numbers a given combination has the more work must be done to guess the combination. The same is true for cryptographic keys, the more bits that are in a key, the less susceptible a key is being compromised by a third party.

2.1. Attacks

One fundamental feature of cryptography is intrusion resulting from attacks on information-based systems. As Simmons [7] perceptively points out, information security is premised on aversion of specific attacks on organizations or individuals across networks. He said communications across networks are susceptible to various attacks from opponent or eavesdropper. These, he remarked depend on situations stimulating such attacks (Simmons) [7]. Stallings emphasized that attacks on the computer system or network security are best characterized by viewing the function of information provided by the computer system. For instance, information flow from file main memory as a source to a destination as well as a user is as shown in Figure 1:



NID→Normal Information Destination, FID→False Information Destination

Figure 1. Security Threat (Stallings, [8])

Interruption, Interception, Modification and Fabrication as depicted in Figure 1 b, c , d, and e respectively.

A useful categorization of these attacks is in terms of passive attacks and active attacks as shown in Figure 2 and Figure 3. This categorization is proposed by Steve Kent [9]. This breakdown is still valid till today and is the basis for most descriptions of security attacks.

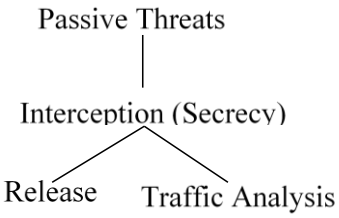


Figure 2. Passive Network Security Threat

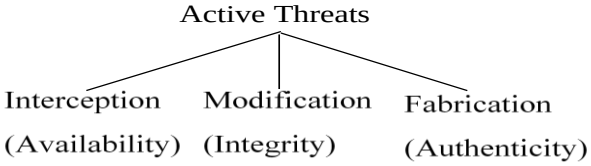


Figure 3. Active Network Security Threat (Kent, 1992) [9]

2.2. Hash Function

One of the fundamental primitives in modern cryptography is the cryptographic hash function, also called message digests and one encryption, often informally called one-way hash function. Hash functions, are algorithms that, in some sense, use no key. Instead, a fixed-length hash-value is computed based upon the plaintext that makes it possible for plaintext contents or length recovery.

Hash algorithms are characteristically used to offer a digital fingerprint of a file's content. It ensures that file is unaltered by an intruder or virus. Hash functions are also commonly employed by many operating systems to encrypt passwords. Hash functions are sometimes misunderstood and some source claims that no two files can have the same hash value. Strictly speaking, this is not true. Consider a hash function that provides a 128-bit hash value. There are obviously 2^128 possible hash values. But there are lot more than 2^128 possible files. Therefore, there have to be multiple files that can have the same hash value. What is, indeed very hard to do is to try to create a file that has a given hash value so as to force a hash-value collision. Nechvata [10] gave the features of a suitable and cryptographically secure algorithm for a hash function. It must be:

(i) Consistent, that is, the same input must always create the output.

(ii) random or give the appearance of randomness- to prevent guessing of the original message.

(iii) unique; that is, it should be nearly impossible to find two messages that produce the same message digest

(iv) one-way; if you are given the output, it must be impossible to ascertain the input message

2.3. Some Existing Signatures

2.3.1. Digital Signature. A digital signature is an electronic identifier that uses cryptography to ensure the integrity, authenticity, and non-repudiation of the information to which it corresponds [11], [12], [13]. A digital signature is also described as an encrypted message digest that is appended to a document. It can be used to confirm the identity of the sender and the integrity of the document [14]. Digital signatures are based on a combination of public-key encryption and one-way hash function algorithms. It is paramount to note however, digital signatures do not provide confidentiality of the message contents, but it is frequently more expedient to produce proof of the originator of a message than to conceal the contents of the message. It is possible that you may want authentication and integrity of messages and of routing updates passed in a network code, without confidentiality, as in this case. The routing contents may not be identical, but it is important to verify that the originator of the routing update is a trusted source. In situation where there is no complete trust between sender and receiver, something than authentication is needed (stallings) [8].

The digital signature is analogous to the handwritten signature. It must have the following properties:

(i) It must be able to verify the author and the date and time of the signature.

(ii) It must be able to authenticate the contents at the same time of the signature.

(iii) The signature must be verifiable by third parties, to resolve disputes.

Thus, the digital signature function includes the authentication function. A variety of approaches have been proposed for the digital signature function. These approaches fall into two categories: direct and arbitrated. The direct digital signature involves only the communication parties (source and destination). It is assumed that the destination knows the public key of the source. The problem associated with direct digital signatures can be addressed by an arbiter, as with direct signature schemes, there are a variety of arbitrated signature schemes. The arbiter plays a very sensitive and crucial rule in this sort of scheme, and all parties must have a great deal of trust that the arbitration mechanism is working properly. Some common public key digital signature algorithms are rivest-shamir-aleman (rsa) and digital signature standard (dss). Dss was proposed by nist based on the el gamal public key algorithm. Comparing rsa and dss; dss is faster for key generation and has about the same performance for generating signatures but it is slower for signature verification [15]. Another form is the ring signature, this permits the user to sign messages on behalf of a "ring" of legitimate signers without revealing the signer's identity [16]. Practical ring signature schemes are also proposed with threshold [17], identity-based ring signature [18], ring signature with signer-admission [19] and proxy ring signature [20]. The first efficient ring signature scheme based on standard assumptions without random oracles was proposed by Shacham and Waters [21]. This is different with the proposed hidden attribute-based signatures. In Subhashini, [22], it was opined that with the anonymity features of attribute-based signatures becoming more prominent, it seemed natural to incorporate it in ring signatures.

2.3.2. Group Signatures. Khader [23] proposed a notion called attribute-based group signature. It allows a verifier to request a signature from a member of a group who possesses certain attributes, and the signature can prove certain attributes ownership. When necessary, the identity of the signer could be revealed by a designated manager. For instance, in a group signature, the signature only reveals the fact that the message was endorsed by one of a set of possible signers. It is conceptually useful to think of these primitives as instances of more abstract "claim-and-endorse" primitives. Indeed the privacy property of the claim-and-endorse primitives is similar to that of a zero-knowledge

proof. NIZKs have been used in the context of group signature constructions Further, Katz et al, [24] proposed a generic notion called identity-based NIZK that, we observe, can in fact be adopted to be an ABS scheme. However, this generic construction will be very inefficient. We point out that in some cases specialized NIZK proofs have been used as part of efficient schemes like the group signature scheme of which uses the NIZK proofs. Group signatures, introduced by Chaum and Van-Heyst [25] provided anonymity for signers. Any member of the group can sign messages, but the resulting signature keeps the identity of the signer secret. In some systems there is a third party that can trace the signature, or undo its anonymity, using a special trapdoor. Some systems support revocation where group membership can be selectively disabled without affecting the signing ability of unrevoked members [25]. Currently, the most efficient constructions are based on the Strong-RSA assumption introduced by Baric and Pfitzman [26]. In the last two years a number of projects have emerged that require the properties of group signatures. The first is the Trusted Computing effort that, among other things, enables a desktop PC to prove to a remote party what software it is running via a process called attestation. Group signatures are needed for privacy-preserving attestation.

2.3.3. Mesh Signatures. Xavier [27] introduced the mesh signature primitive as an anonymous signature that borrows from ring signatures, but with added modularity and a much richer language for expressing signer ambiguity. The language can represent complex access structures, and in particular allows individual signature components to be replaced with modular certificate chains. As a result, withholding one's public key from view is no longer a shield against being named as a possible cosignatory; and hence, a mesh signature may be used as a ring signature substitute with compulsory enrollment. Ring signatures can thus only ever implicate individuals who, by the very act of publishing their key, are proclaiming their consent. Mesh signatures generalize this notion to monotone access structures represented as a tree, whose interior nodes are AND, OR, and Threshold gates, and whose leaves are regular "atomic" signatures. The atomic signatures may be "static" and reusable, as opposed to fresh; hence PKI certificates are eligible even if the mesh signer lacks the CA's signing key. Since furthermore the access structure is powerful enough to express disjunctions of certificate chains, we are no longer beholden to the prior publication of all the ring keys.

2.3.4 Identity-Based Encryption. An Identity Base Encryption (IBE) scheme is a public-key cryptosystem where any string is a valid public key. In particular, email addresses and dates can be public keys. The IBE email system is based on the first practical IBE. The cryptosystem has chosen cipher text security in the random oracle model assuming an elliptic curve variant of the computational Diffie-Hellman problem. The original motivation for identity-based encryption is to help the deployment of a public key infrastructure [28]. In 1984 Shamir asked for a public key encryption scheme in which the public key can be an arbitrary string. In such a scheme there are four algorithms: (1) setup, (2) extract (3) encrypt and (4) decrypt decrypts messages using the corresponding private key. Shamir's original motivation for identity-based encryption was to simplify certificate management in e-mail systems. Different IBE systems were presented by Boneh et al [29]; Cocks [30]; Canetti et al [31]; Boneh and Boyen [32]; Waters [33] and Gentry [34].

2.3.5 Fuzzy Identity-Based Signature. The concept of fuzzy identity based encryption (IBE) was introduced by Sahai and Waters [35]. In a nutshell a fuzzy identity based encryption allows a user with the private key for identity ID0 to decrypt a cipher text encrypted for identity ID1 if and only if ID0 andID1 are within certain distance based on some metrics. In this application, a user can issue a signature on behalf of the group that has a certain set of attributes. For instance, an IT firm might want a C# senior programmer whose age is above 50 to sign the technical report. In this scenario, it will sign to the identity {"C#", "senior programmer", "above 45"}. Any user who has an identity that contains all of these attributes could issue the signature. As a similar notion to ABS, fuzzy identity-based signature was proposed and formalized in Wang et al [36] which enables users to generate signatures with part of their attributes. But Tan et al [37] pointed out that this scheme is vulnerable to the partial key replacement attack. Moreover, in these works, authors do not consider any notion of privacy, resulting in leaking attributes used in producing signatures to the verifier.

3. Related Works

Traditional attribute architectures and cryptosystems are ill-equipped to provide security and authentication requirements in the face of diverse access requirements and environment [38]. Consequently, there is a great need to meet this security and authentication requirements using attribute-based signature system. The introduction of public-key cryptography by Diffie and Hellman in 1976 was an important watershed in the history of cryptography. The work sparked off interest in the cryptographic research community and soon several public-key schemes were proposed and

implemented. The Rivest, Shamir and Adleman (RSA), being the first realisation of this abstract model, is the most widely used public-key scheme today [39].

Fuzzy identity based encryption (IBE) was introduced by Sahai and Waters [35]. It is an identity based encryption that allows a user with the private key for identity ID0 to decrypt a ciphertext encrypted for identity ID1; if and only if ID0 and ID1 are within a certain distance based on some metrics. Fuzzy IBS can be directly applied to identity based signature system that uses biometric identities.

An attribute-based group signatures which is a primitive hiding only the identity of the signer but reveals which attributes the signer used to satisfy the predicate was proposed by Khader[23]. It allows a group manager to identify the signer of any signature which is similar to the semantics of group signatures. In contrast, we require signer privacy to hold against everyone, including all authorities.

Maji et al, [40] presented an identity-based signature called Attribute-Based Signature (ABS). A signer posseses a set of attributes rather than a single string representing the signer‘s identity.

An Attribute-based cryptography is a natural solution for fine-grained access control with respect to security policies [41]. In the case of attribute-based signatures (ABS), users obtain from an authority their secret keys as a function of the attributes they hold, with which they can later sign messages for any predicate satisfied by their attributes. Another related notion to ABS is fuzzy identity-based signature which was proposed and formalized in (Shanqing; and Yingpei [42]; Yang; et al. [43]. It allows a user with identity ω to issue a signature which could be verified with identity ω‘ if and only if ω and ω‘ are within a distance judged by some metric. However, this kind of signatures does not consider the anonymity for signer.

4. Design

This section highlights the logic tools, algorithms and design procedure for our attribute-Based Signature system. The processes are stated as follows:

Attribute-based signature system consists of four main algorithms, namely, setup algorithm setup**,** key generation algorithm keyGen, signing algorithm Sign and verification algorithm Verify. Others are the encryption algorithm Enc and decryption algorithm Dec. in order to reach the goals of implementing these algorithms, several functions necessary for their constructions were created.

Let A be the universe of possible attributes. A claim-predicate over A is a monotone Boolean function, whose inputs are associated with attributes A. Thus, say that an attribute set A $\underline{C}$ A satisfies a claim-predicate Y if Y (**A**) = 1 (where an input is set to be true if its corresponding attribute is present in (**A**).

SETUP: The authority obtains a key pairs (PK, SK) ← Setup (), and outputs public key PK and keeps a private secret key SK.

Encryption: The Algorithm Enc (M, Y, PK) is a randomized algorithm that takes as input the message M to be encrypted, the access structure Y which needs to be satisfied and the public parameters PK to output the cipher text CT. Consequently, the encryption algorithm embeds the access policy (Y) in the Cipher text such that only those users with attributes satisfying Y will be able to decrypt and retrieve the message M.

i) Rivest, Shamir, And Adleman (RSA) Encryption Technique

The standard algorithm for implementing public-key cryptography can be used for both encryption, key generation and decryption called RSA algorithm is used. The basic algorithm is outlined below.

Input: RSA public key (n,e), plaintext m $\in${0,n-1}.

Output: Cipher text c

Compute c=m^e mod n

Return (c)

Key-Generation: A user is assigned a set of attributes A $\underline{C}$ A, the authority computes a signing key SK_A ← Key-Gen (MK, PK,A) and gives it to the user. The algorithm takes as input the secret key value SK, the public key PK and the attribute set A of the user, and outputs for the user a set of decryption keys SK_A which confirms the users possession of the attributes in A and no other external attribute.

ii) RSAKey Generation Technique

The public and matching secret key is generated as highlighted below:

- Choose two large distinct primes, p and q
- Compute the product (modulus) n=p * q
- Compute Euler’s phi function Φ(n)=(p-1) * (q-1)
- Randomly choose an encryption key e, such that e and Φ(n) are relatively prime.

Two numbers are relatively prime when they share no factors in common other than 1.

- Finally, calculate the decryption key d, the multiplicative inverse of e mod Φ(n) such that

d=e^{-1}mod Φ (n)

d=e^{-1} mod (p-1)*(q-1)

d and n are relatively prime. The numbers e and n are the public keys. The number d is the secret key.

iii) Decryption: The decryption algorithm Dec(CT, SK, PK) takes as input the cipher text CT, the user secret keys SK and the public parameters PK, and it output the encrypted M←Dec (CT, SK, PK), if and only if the attribute sets A embedded in SK satisfy the access policy Y which was used

while encrypting the cipher text CT. That is, if Y (A) = 1 then M is output else, it outputs Reject.

RSA Decryption Technique

Input: RSA public key (n,e), RSA private key d, ciphertextc.

Output: plaintext m.

- Compute m=C^d mod n.
- Return (m)

Sign:To sign a message M with a claim – predicate Y and a set of attributes A such that Y (A) = 1. The user computes a signature σ← sign (PK, SK_A M, Y,).

Verify:To verify a signature σ on a message M with a claim – predicate Y, a user runs verify (PK, M, Y, σ) which outputs a Boolean value, accept if Y (A) = 1, where A is the subset of attributes the signer embedded in σ that satisfies the predicate, else it outputs reject. The minimal correctness property of attribute-based signature system is that honestly–generated signatures pass the verifications check.

4.1. Computing Digital Signature

The RSA digital signature private keys are used to sign, and the corresponding public keys are used to verify. RSA is reversible, since (Md)e = (Me)d = Med (mod n); one can raise to the public exponent (e) first, and raise to the private exponent (d) second, or vice versa, and either way obtain the original message back. It is claimed that the security of RSA is equally good both ways. When using RSA for digital signatures, signing of a message M, in principle, is just transforming the message M with the signer's private RSA key. This signature is appended to the message itself, like any authentication tag. Verification of the signature is done by applying the reverse transform to the signature with Alice's public key, and checking that the result is equal to the received message. The signed message can be passed around, and the signature can be re-verified as needed by anyone possessing Alice's public key. For various technical reasons, this simple scheme is modified in practice. For example, if Eve succeeds in having Alice sign messages M1 and M2, then she can claim that Alice also signed M3, where M3 is the product of M1 and M2: (M3)d = (M1 × M2)d = M1 d × M2 d (mod n). Thus, if Eve sends M3 to Bob, then Bob erroneously believes that Alice signed message M3. To avoid this problem (and some others) Alice usually creates a cryptographic hash of the message, and creates an authentication tag by signing this hash. (This also has the pleasant side effect that her signature doesn't have to be as long as the original message.) Bob verifies the signature by first re-computing the hash from the message, then applying the reverse transform to the received authentication tag with Alice's public key to obtain the received hash, and, finally, checking if the recomputed hash and the received hash are the same (Saltzer et at [12]).

A. Algorithm for Security Requirements

Other algorithms for the secure ABS are:

i) Correctness: We call an ABS scheme correct if for all (PK, SK) ←setup, all messages M, all attribute sets A, all signing keys SK_A←key-Gen (SK, A), and all the claim-predicates Y such that Y (A) =1, Verify (PK, M,Y, sign (PK, SK_A, M, Y)) = Accept with probability 1 over the randomness of all the algorithms.

ii) Anonymity/Privacy: An ABS scheme is completely private if, for all (PK, SK)←Setup, all attribute sets A_1, A_2, all SK_1←Key-Gen (SK, A_1), SK_2←Key-Gen (SK, A_2), All messages M, and all claim - predicates Y such that Y (A_1) = Y (A2) = 1, the distributions sign (PK, SK_1, M, Y,) and sign (PK, SK_2, M, Y,) are equal.

iii) Unforge-ability: An ABS scheme is unforgeable if the success probability of any polynomial – time adversary in the following design is negligible.

B. Attribute-Based Signature System Design

We present a construction of an ABS scheme using Cryptographic tools recently developed in the context of attribute-based encryption.

Preliminaries

i) Groups with Bilinear Pairings

Let G,H, G_T, be cyclic (Multiplicative) groups of order P, where P is a prime. Let g be a generator of G, and h be a generator of H. then e:G x H→G_T is bilinear pairing if e(g, h) is a generator of G_T, and $e(g^a, h^b) = e(g,h)^{ab}$ for all a,b.

ii) Monotone Span Programs

The essence of the monotone span program is to take care of the constant signature size in our scheme by using only AND threshold gate. Let Y be a monotone Boolean function. A monotone span program for Y over a field F is an ℓ x t matrix M with entries in F, along with a labeling function υ that associates each row of M with input variable of Y.

We call ℓ the length and t the width of the span program. Every monotone Boolean function can be represented by some monotone span program. The size of the signature in our scheme depends on the dimensions of claim – predicate's monotone span program. As an example, a monotone span program for the predicate given involving five (5) attributes with three (3) AND gates (which are binary threshold gates with thresholds 3) is given by:

$$M = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 \end{bmatrix}^T$$

M has four (4) rows, labeled by the four (4) attributes and three (3) columns corresponding to the three thresholds AND gates.

C Hardness Assumptions: Bilinear Diffie-Hellman Exponent Assumption

Let e: G*G→ GT be an efficiently computable bilinear map, where G has prime order p with generator g. Let a, s € Zp be chosen at random. Let g^i denote g^a. The q-Bilinear Diffie-Hellman Exponent(q-BDHE) assumption holds in G if, given the following vector of 2q + 1 elements, y = (g, g1,…, g_q, g_{q+2},…, g_{2q}, g^s) (note that the g_q+1 is not in the list); it is infeasible for a polynomial time adversary to compute e (g, g)$^{aq+1s}$.

5. Simulation

TheAttribute-Based Signature scheme consists of four algorithms, namely, setup algorithm Setup, private key generation algorithm Key-Gen , signing algorithm Sign, and verification algorithm Verify.

We construct our proposed system based on our case study (Computer science Department of FUTA). For instance, in Computer Science Department of the Federal University of Technology Akure, grade-sheets of a class may be accessible only to a Professor handling the Course and the Teaching Assistants (TAs) of that Course. The policy is expressed in terms of a predicate as:

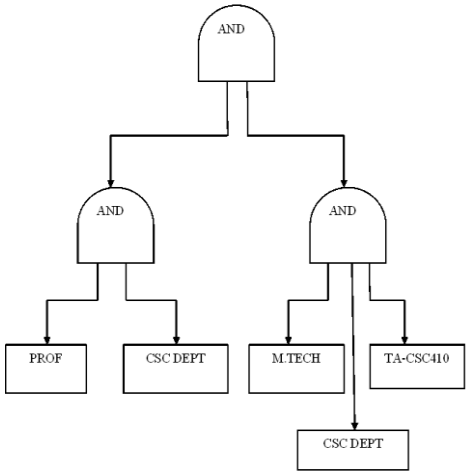
((Professor ^ CSC dept.) ^ (M.tech student ^ TA CSC-410 ^ CSC dept.))

The access policy is represented as a tree structure as in Figure 4. The access control policy above is a policy that defines the kind of users who have permissions to encrypt or decrypt a message in our proposed Attribute-Based Signature system. User's access to information or resources depends on the combinations of these attributes.

i) Set Up: the setup algorithm chooses bilinear group triple (G_1, G_2,G_T) of prime order p and e as a bilinear map, e:G×G→G. The algorithm also picks generators g of G_1 and h of G_2. Then it chooses three (3) random exponents $\propto$, β, γ in Z*$_{P.}$ it then sets u=g$^{\alpha\gamma}$and υ=e(g$^{\alpha}$, h).Choose a collision resistant hash functionH:{0,1}*→z$^*_p$.Let the universe of attributes be A ={CSC DEPT(C), PROF (P), M.TECH (M), TA-CSC410 (T) }. So, m = 4. Now, for simplicity we'll say that C,P,M and T are all values in Z *p and the function Y when applied on these attributes give the same value.

PK (Public Key): {A, υ, υ, h$^{\beta}$, Y }

SK (Secret Key): {α, β, Y, g, h}



((PROF ^ CSC DEPT) ^ (M.TECH ^ TA CSC-410 ^ CSC DEPT))

Figure 4. Access Policy of the Proposed Attribute-Based Signature System

ii) Generating the Keys

Key-Gen (PK, A, SK)

We generate the keys for the person with attributes A ={CSC DEPT,PROF, M.TECH, TA-CSC410}. Here, A□P, the central authority picks an r € Z*$_P$ at random and computes the secret key for the user as follows:

$SK_A = \{\{g^{\frac{r}{\gamma+c}}, g^{\frac{r}{\gamma+p}}, g^{\frac{r}{\gamma+M}}, g^{\frac{r}{\gamma+T}}\}, g^{\frac{\alpha(1-r)}{\beta}}\}$

iii) Encrypting a Message

Enc (PK, Y, M)

For every non-leaf node x of the access policy Y, we choose a polynomial q_x. We proceed in a top down manner in selecting the polynomials, starting from the root R. For a node x we set the degree of the node $d_x = k_x - 1$, one less than the threshold value that needs to be satisfied at the gate at that node.Now, beginning at the root, we choose a random s € R Z*$_P$ and set $q_r(0) = s$.

Then choose d_r other points of the polynomial q_r to define it completely. For all other non-leaf nodes x, we set $q_x(0) = q_{parent(x)}$ (index(x)) and choose d_x other points to completely define q_x.

The cipher text is given by:

CT = {M.e(g, h)$^{\alpha s}$, C_0 = h$^{\beta s}$, C_1, C_2}, where C_1 and C_2 are the values of node 1 and node 2.

iv) Decrypting the Ciphertext

Dec (CT, SK, PK)

We will show the decryption for the case of the given secret key components.

Now, we can decrypt the message from M. e(g, h)$^{\alpha s}$ as follows:

- $e(g^{\frac{\alpha(1-r)}{\beta}}, h^{\beta s}) = e(g,h)^{\alpha s-\alpha rs}$
- Multiply the above with $e(g,h)^{\alpha rs}$ to get $e(g,h)^{\alpha s}$
- M = $\frac{\text{M.e (g,h)}\alpha s}{e(g,h)^{\alpha s}}$

v) Signing a Message

Sign (PK, SK_A, M, Y)

If Y (A) = 1, then convert Y to its Corresponding monotone span program M € ($z^*_p)^{\ell xt}$, with row labeling υ: [ℓ] →A. Also compute the vector υ that corresponds to the satisfying assignment A. compute μ = H (m||Y). Pick random $k_0 \leftarrow Z^*_p$ and $k_1, \ldots k_L \leftarrow Z_p$ and compute:

Y = $g^{\frac{r\,k0}{\gamma+c}}$ W = $g^{\frac{rk0}{\gamma+p}}$ S = $g^{\frac{rk0}{\gamma+M}}$ V = $g^{\frac{rk0}{\gamma+T}}$ Q = $g^{\frac{\alpha(1-r)k0}{\beta}}$ K = $\prod_{i=1}^{l}(g^{\alpha\gamma}, e(g^{\alpha},h)^{M.Ki}$

Signature: σ = (Y, W, S, V, Q, K)

vi) Verifyinga Message

Verify (PK, σ = (Y, W, S, V, Q, K), M, Y)

First convert Y to its corresponding monotone span program M € ($z^*_p)^{\ell xt}$with row labeling u: [ℓ] →A compute μ= H(M//Y). If Y=1 output reject. Otherwise check the following constraint:

e (W, υ) = $\dot{e}(Y, h^{\beta})$

Returns accept if the above check succeeds, and reject otherwise.

A. Probabilistic Verification of a Message

To probabilistically verify a signature, proceed as in the normal verification algorithm, but replace the final t checks with the following random one. Choose random $r_1 \ldots r_t \leftarrow Z^*_{P,}$ and check the single constraint:

$\prod_{i=1}^{l} e\,(s \prod_{i=1}^{t}(g^{\alpha\gamma}, e(g^{\alpha},h)^{M.ri}{}_{=}$

$e(y, h1^{ri})\; e(gh^{\mu}, \prod_{i=1}^{t} k^{\text{ri}})$

This is essentially a random linear combination of that original constraint. Legitimate signatures pass such a check with probability 1, while invalid signatures pass with probability at most 1/k.

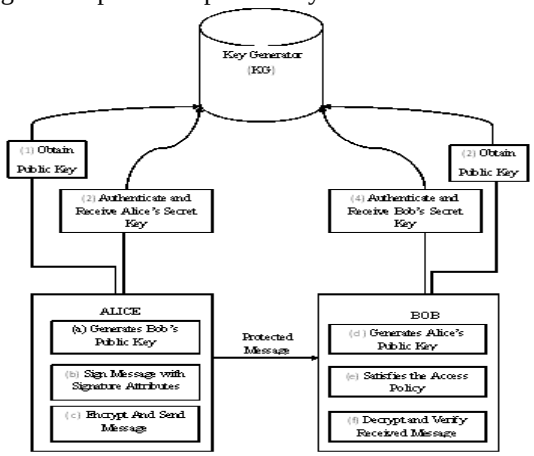


Figure 5. Proposed Attribute-Based Signature System Block Diagram

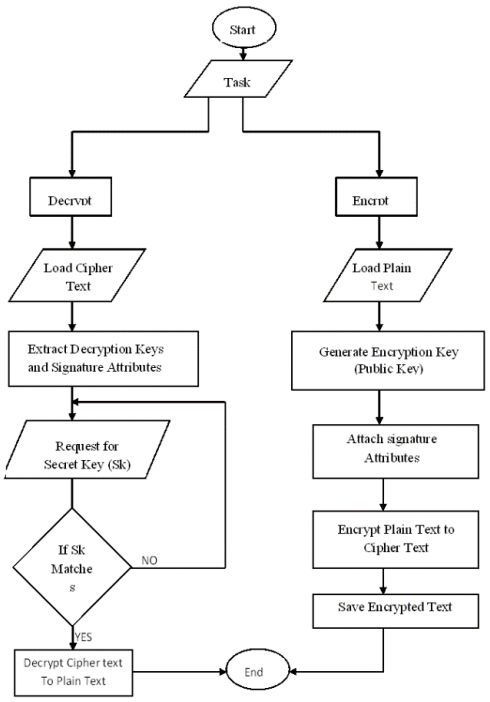


Figure 6. Flow Chart of the proposed Attribute-Based Signature System

The simulation of the algorithm was carried out using C# programming language on the .NET platform running on a 2GB RAM, 32-bit Dou Core windows 7 Operating System. The result is an Attribute-Based System as shown in Figure 6.

Attribute-based signatures allow users possessing a set of attributes to sign documents; although the attributes of the signer can be verified, signers can still continue to retain a reasonable degree of anonymity. Here we present the various steps involved in attribute-based signature system using the RSA public-key cryptographic technique. From the block diagram above in Figure 5, each person gets a pair of keys, called the public key and the secret key. The public key is published and widely distributed, while the secret key is never revealed. The need for exchanging secret keys is eliminated as all communications only involve public keys. No secret key is ever transmitted or shared.

Thus, when user Alice wishes to send an encrypted message to Bob, she looks up Bob's public and her secrete key in the key generator, uses it to encrypt the message, and sends it off to Bob. Bob then uses his secret key (SK) to decrypt and verify the received message. In this scheme, the true identity of the sender is not revealed. Rather, only her attributes which satisfy the access policy are being revealed. There is also a guarantee of unforge-ability and collusion- resistance in our proposed system. Anyone who has access to Bob's public key can send him an encrypted message, but no one else apart from Bob can decrypt it.

6. Comparative Analysis and Implementation

In this work, a secured ABS scheme based on standard model on decisional parallel bilinear Diffie-Hellman exponent assumptions is proposed. The scheme proposed is more practical than the generic group model of Maji et al. [40]. Maji et al. [44].and Okamoto and Takashima [45] outlined that the future work of ABS; theoretically, based on the ABS security of standard hardness assumption, the efficiency for the most part even though it is still preserved.

The proposed ABS scheme is efficient and practical. Our scheme was compared with the existing ABS schemes in the standard model which are the works Maji et al. [44]., Okamoto and Takashima's [45], and Escala et al. [46], as well as the ABS scheme in the generic group model of Maji et al. [40]. All of these schemes were implemented over a pairing group and the size of a group element is about the size of Z*p.

Our Comparison further revealed that our proposed attribute-based signature system offers considerable bandwidth savings by using constant signature size. In our proposed system, Constant signature size was achieve by using only the AND threshold gate, fixed Secret Key length and constant size cipher text. Other schemes on the other hand, scales poorly in terms of bandwidth savings since such schemes were constructed based on both AND and OR threshold gates, variable Secret Key and cipher text length. In Table 1 we summarize the comparison.

In Table l and r represent the size of the underlying access structure matrix M for a predicate, that is, M $\in Z^{l\times r}$. We also give comparison of one example, the predicate with 4 AND and 5 OR gates as well as 10 variables which is expressed by a 10 × 5 matrix, λ is the security parameter (e.g. 128). As the above comparison, our design is the most efficient ABS scheme especially in terms of signature size and model in the standard model compared with Maji et al. [44], Okamoto and Takashima [45], Escala et al. [46] constructions.

Screen shots of Interfaces showing some selected outputs.

Here, the user is given the choice to select an access policy or signature attributes, generate keys and finally encrypt the plaintext using the "ENCRYPT" icon.

Table 1. ABS Systems Comparison

Parameters / Authors	Signature Size	Model	Security	Assumptions	Predicates	SigSize Example (L=10,R=5, λ=128)
Maji;Prabhakaran; Rosulek, (2008)	1 + r + 2	Generic Group Model	Full	CR HASH	Monotone	17
(Maji;Prabhakaran; Rosulek, 2011) Boneh-Boyen Based)	511 + 2r+ 18λl	Standard Model	Full	q- SDH and DLIN	Monotone	23560
Maji , Prabhakaran; Rosulek, (2011) Waters Based	36l + 2r+9λ + 12	Standard Model	Full	DLIN	Monotone	1534
Okamoto ;Takashima, (2011)	7l + 11	Standard Model	Full	DLIN andCR hash	Non-Monotone	81
(Escala;Herranz;Morillo, (2011)	9l + 7	Standard Model	Full	CHD and Subgroup Decision	Monotone	97
Proposed	5l+3	Standard Model	Full	q- BDHE and CR hash	Monotone	53

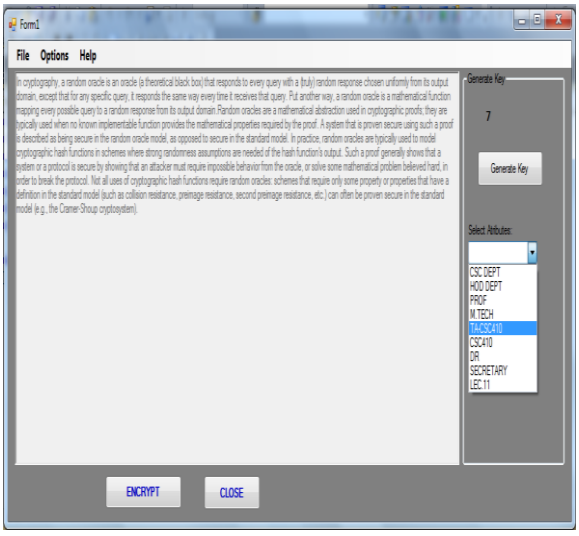


Figure 7. Encryption interface with signature attributes, plain text and key generator

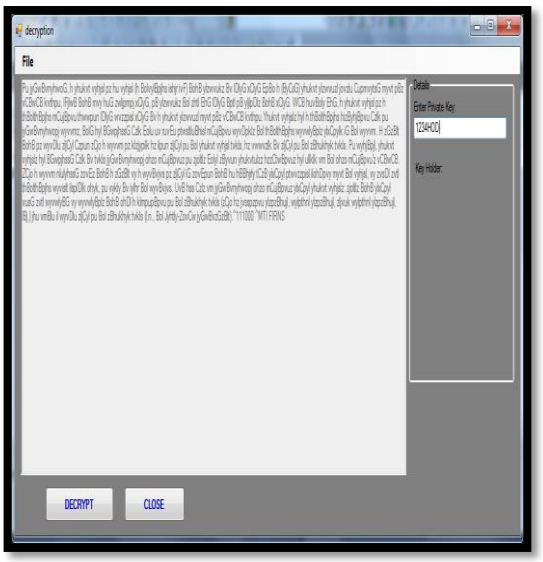


Figure 8. Decryption interface with the cipher text

The user, having supplied the right secret key is now given the option to decrypt the cipher text by just clicking the "DECYPT" icon above.

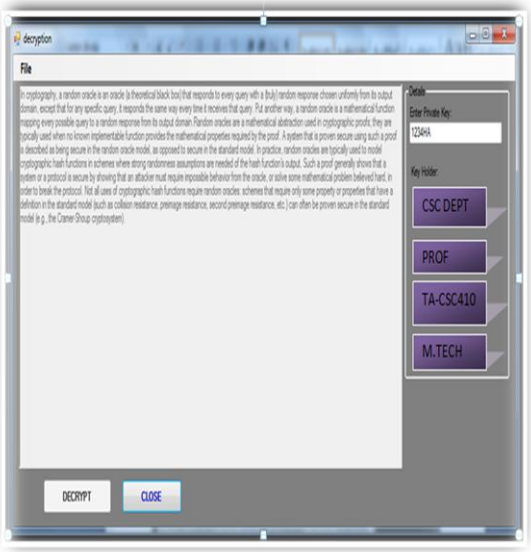


Figure 9. Decryption interface with the decrypted text (original plain text) and signature attributes of the signer

At this interface, the user is given the authority to read his decrypted message and to view the signature attributes (CSC DEPT, PROF, TA-CSC410 and M.TECH) of the signer. The true identity of signer is not revealed (that is, anonymity/privacy is ensured). Unforge-ability and collusion-resistance is also guaranteed.

7. Conclusion

We have presented an efficient and fully secure attribute-based signature system that is expressive and provably secure under decisional q-parallel BDHE assumption in the standard model. The security definitions and models were also suggested and formalized. Unforge-ability and anonymity are defined for the Attribute-based signature. More specially, unforge-ability requires that any user without certain attributes cannot generate signature with attributes that he does not have. Anonymity allows the signer to generate a signature with part of his attributes while being anonymous among all the users with the same attributes purported in the signature. Our method of embedding a monotone span program into the public parameters allowed us to create clean, modular proof of security.

In real life, one requires signatures from people who satisfy certain criteria like that they should possess some specific attributes. In order to satisfy this kind of need, an efficient attribute-based signature scheme is proposed, where the signing member has to have certain attributes. Attribute-based signature system has a lot of applications in real scenarios requiring both authentication and privacy properties, such as anonymous authentication and attribute-based messaging system. In this research work, we present an efficient Attribute-Based Scheme that controls the access of score sheet in computer science department of Federal University of Technology Akure which supports threshold predicates based on RSA algorithm. The proposed Attribute-Based Scheme has proven to be existentially unforgeable in the standard model for the selective adversary and can achieve complete privacy with collusion-resistance when compared with some existing works. Our construction provides better efficiency in terms of computational cost and signature size.

8. References

[1] CYCOM Cipher Research Laboratory (2006), A brief history of cryptography, lecture notes in computer science, volume 630, pp. 75-96.

[2] Falaki, O.S. (2002), Information Technology in Nigeria, now or never, Inaugral lecture series 29 Delivered at the Federal University of Technology, Akure.

[3] Alese, B.K. (2000), Vulnerability Analysis of Encryption/Decryption techniques of computer Network security, Master's Thesis, Federal University of Technology, Akure, Nigeria.

[4] Kessler, G.C. (2001), An overview of Cryptography, Auerbach publishing, pp. 125-140.

[5] Menezes, A. J., Van-Oorschot, P.C and Vastone, S.A. (1996), Handbook of applied cryptography, CRC Press, pp. 816-820.

[6] Tanenbaum, A.S. (1996), Computer Networks, Prentice-Hall, USA, pp. 314-322.

[7] Simmons, G.J. (1993), Contemporary Cryptography: The Science of Information Integrity, IEEE Press, New York, 1992. pp. 24-26.

[8] Stallings, W. (1999), Cryptography and Network Security: Principles and Practice, Prentice-Hall, USA, pp. 205-215.

[9] Kent, S. (1992), Architectural Security for the Internet in the Internet System Handbook, Addison Wesley, pp.433-449.

[10] Nechvatal, J. (1992), Public Key Cryptography, Science of information Integrity. IEEE Press, pp.35-37.

[11] Kaeo, M. (1999), Designing Network Security, Macmillan Technical publishing, pp. 506-517.

[12] Saltzer, H.J. and Schroeder, (2005), Protection of computer information, Proceedings of the IEEE, Volume 63, No 9, pp.1278-1308.

[13.] James, H.D. and Dalia, K. (2009), Digital signatures: “What you are" versus “Who you are", Lecture Notes in Computer Science, pp. 1-2.

[14] Alese, B.K. (2004), Design of public key cryptosystem using elliptic curve, Ph.D Thesis, Federal University of Technology Akure, pp. 41-50.

[15] Elgamal, T. (1985), A public key cryptosystems and signature scheme based on discrete logarithms, IEEE Transaction on information theory, Volume 31, pp. 469-472.

[16] Rivest, R.L., Shamir, A. and Tauman, Y. (2001) How to leak a secret, In C. Boyd, editor, ASIACRYPT, Lecture Notes in Computer Science, volume 2248, Springer, pp. 552–565.

[17] Bresson, E., Stern, J. and Szydlo, M. (2002), Threshold ring signatures and applications to ad-hoc groups. In Crypto'02, LNCS 2442, pp. 465-480.

[18] Chow, S.S.M and Chase, M. (2009), Improving privacy and security in multi-authority attribute-based encryption, ACM Conference on Computer and Communications Security, pp. 121–130.

[19] Wang, Y., Li, J., Chen, X. and Yuen, T.H. (2007), Proxy ring signature: formal definitions, efficient construction and new variant, CIS’06, LNCS, volume.4456, Springer, pp. 545–555.

[20] Li, J., Wang, Y., Chen, X. and Yuen, T.H. (2007), Proxy ring signature: formal definitions, efficient construction and new variant, LNCS, volume.4456, Springer, pp. 545–555.

[21] Shacham, H. and Waters, B. (2007), Efficient ring signatures without random oracles, LNCS, volume 4450, Springer, pp.166–180.

[22] Subhashini, V. (2011), Attribute Based Cryptology, lecture notes in computer science, volume 6500, pp. 31-35.

[23] Khader, D. (2007), Attribute based group signature with revocation. Cryptology ePrint Archive, Report 2007/241, http://eprint.iacr.org/, pp.1-3.

[24] Katz, J., Ostrovsky, R. and Rabin, M. O. (2004), Identity-based zero knowledge, In C. Blundo and S. Cimato, editors, SCN, Lecture Notes in Computer Science, volume 3352 Springer, pp.180–192.

[25] Chaum, D. and Van Heyst, E. (1991), Group signatures, In EUROCRYPT, pp. 257–265.

[26] Baric, N. and Pfitzman, B. (1997), Collision-free accumulators and fail-stop signature schemes without trees, In Proceedings of Eurocrypt 1997, Springer-Verlag, pp. 480–494.

[27] Xavier, B. (2007), Mesh signatures, In Proceedings of the 26th annual international conference on Advances in Cryptology, EUROCRYPT, Berlin, Heidelberg, Springer, pp. 210–227.

[28] Dan, B. and Mathew, F. K. (2001), Identity-based encryption from the Weil paring, Lecture Notes in computer science, volume 2139, pp. 213-229.

[29] Boneh, D. and Franklin, M. (2001), Identity-Based Encryption from the Weil Pairing, Advances in Cryptology-CRYPTO'01, pp. 213- 229.

[30] Cocks, C. (2001), An Identity Based Encryption Scheme Based on Quadratic Residues, Proceedings of the 8th IMA International Conference on Cryptography and Coding, pp. 26-28.

[31] Canetti, R. and Katz, J. (2004), Chosen-Ciphertext Security from Identity-Based Encryption, Advances in Cryptology- EUROCRYPT'04, pp. 207-222.

[32] Boneh, D. and Boyen, X. (2004), secure identity based encryption without random oracles, In M. K. Franklin, editor, CRYPTO, Lecture Notes in Computer Science, volume 3152, Springer, pp. 443–459.

[33] Waters, B. (2005), Efficient Identity-Based Encryption without Random Oracles, Advances in Cryptology-EUROCRYPT'05, pp.114- 127.

[34] Gentry, C. (2006), Practical Identity-based encryption without random Oracles, Advances in Cryptology-Eurocrypt'06, pp. 445–464.

[35] Sahai, A. and Waters, B. (2005), Fuzzy identity-based encryption, In R. Cramer, editor, EUROCRYPT, volume 3494 of Lecture Notes in Computer Science, Springer, pp. 457–473.

[36] Wang, C., Chen, W. and Liu, Y. (2009), A Fuzzy Identity Based Signature Scheme, International Conference

on E-Business and Information System Security (EBISS'09), pp. 1-5.

[37] Tan, S., Heng, S. and Goi, B. (2009), On the Security of an Attribute-Based Signature Scheme, Communications in Computer and Information Science, pp. 161-168.

[38] Matthew Pirretti, Patrick Traynor, Patrick McDaniel and Brent Waters (2006). Secure Attribute-Based system, Alexandria, Virginia, USA. Pages 2-3.

[39] Alese B. K. et al. (2012).Comparative Analysis of Public-KeyEncryption Schemes. InternationalJournal of Engineering and Technology Volume 2, No. 9, pages 8-9.

[40] Maji H., Prabhakaran, M. and Rosulek, M. (2008). Attribute-based signatures. In A. Kiayias, editor, Topics in Cryptology, volume 6558 of Lecture Notes in Computer Science, Springer Berlin / Heidelberg. pages 376–392.

[41] Javier Herranz, Fabien Laguillaumie, Benoit Libert, and Carla Rafols (2011). Short Attribute-based signatures for threshold predicates.pages 19–34.

[42] Shanqing, G and Yingpei, Z. (2008). Attribute-based signature scheme. In Proceedings of the 2008 International Conference on Information Security and Assurance (ISA 2008), IEEE Computer Society , Washington, DC, USA, pages 509–511.

[43] Yang,P., Cao, Z. and Dong,X. (2011). Fuzzy identity based signature with applications to biometric authentication. Computers and Electrical Engineering, pages 532 – 540.

[44] Maji, H., Prabhakaran, M. and Rosulek, M. (2011), Attribute-based signatures, In A. Kiayias, editor, Topics in Cryptology , volume 6558 of Lecture Notes in Computer Science, Springer Berlin /Heidelberg, pp. 376–392.

[45] Okamoto, T. and Takashima, K. (2011), Efficient attribute-based signatures for non-monotone predicates in the standard model, Public Key Cryptography, volume 6571 of Lecture Notes in Computer Science, Springer, pp. 35–52.

[46] Escala, A., Herranz, J. and Morillo, P. (2011), Revocable attribute-based signatures with adaptive security in the standard model, In Advances in Cryptology, volume 6737 of Lecture Notes in Computer Science, Springer pp. 224–241.

