
# PROJECTE FINAL DE CARRERA

## Revocation in Anonymous Authentication Systems

## (Revocació en Sistemes de Autenticació Anònims)

*Estudis: Enginyeria de Telecomunicació*

*Autor: María Rosa Fueyo Pestaña*

*Director/a:* Javier Herranz Sotoca

*Any: 2015*

# Abstract

**Keywords:** Attribute-based signatures, RSA, privacy, unforgeability, revocation, Polynomial Evaluation, Zero-Knowledge Proof, discrete logarithm, non-membership proof.

**MSC2000:** 92A60, 92A62

An attribute-based signature with respect to a signing policy chosen by the signer, convinces the verifier that the signer sustains a subset of attributes satisfying that signing policy. The verifier must not obtain any other information about the identity of the signer or the attributes he holds. This type of signatures have a lot of applications in real life scenarios that demand both authentication and privacy properties. The ability of revoking users that have misbehaved or lost their attributes, so that they can not compute more valid signatures, is very desirable for real life applications of attribute-based signatures.

In this project, the main goal consists in studying different protocols of revocation and incorporating them into an already existing RSA attribute-based signature. In order to achieve these objectives, two different protocols were chosen from those available in the literature, taking into account the efficiency of the existing protocols and the necessity that the protocol is built in an anonymous way: the user must not reveal his identity when proving that he is not in the revocation list.

The first one is based on a polynomial evaluation argument, and some of its main advantages are that this argument has logarithmic communication cost in the number of revoked users in contrast to other protocols with cubic root complexity at best, thus obtaining a more efficient protocol and besides its security relies only on the discrete logarithm assumption.

The second one was based on a protocol for special cases when the revoked elements are coprime, since in the attribute based signature the elements are prime, this protocol was considered suitable. While demonstrating its soundness it was found that the original protocol was not secure because we found a particular attack (that we describe in this work). Thus, a new secure protocol was designed which fits with the attribute-based signature.

Finally, the previous protocols have been incorporated into an existing RSA attribute-based signature scheme, and both resulting signatures have been analyzed in terms of efficiency of the communication cost. The second protocol is shown to be always more efficient than the first one, even for the case with a single revoked user.

# Resum

**Paraules clau:** Signatura basada en atributs, RSA, privacitat, infalsicabilitat, revocació, Avaluació de polinomis, prova de coneixement zero, logaritme discret, prova de no-pertinença.

**MSC2000:** 92A60, 92A62

Una signatura basada en atributs en relació amb una política de signatures triada pel signant, convenç al verificador que el signant té un subconjunt d'atributs que satisfan la política de signatures. El verificador no obtindrà cap informació sobre la identitat del signatari o sobre els atributs que té. Aquest tipus de signatures té moltes aplicacions en situacions de la vida real que demanden autenticació i privacitat al mateix temps. L'habilitat per revocar usuaris que s'han comportat malament o han perdut els seus atributs i ja no poden computar més signatures vàlides, és molt desitjable per a aplicacions de signatures basades en atributs en la vida real.

En aquest projecte, el principal objectiu consisteix a estudiar diferents protocols de revocació i incorporar-los en un esquema signatura RSA basada en atributs ja existent. Per poder aconseguir aquests objectius, dos protocols van ser triats entre els disponibles en la literatura existent, tenint en compte l'eficiència dels protocols existents i la necessitat que el protocol hagi estat construït de manera anònima: l'usuari no pot revelar la seva identitat en el moment de provar que ell no està en la llista de revocació.

El primer protocol està basat en un argument d'avaluació de polinomis, i algun dels seus principals avantatges és que té cost de comunicació logarítmica en el nombre d'usuaris revocats en contrast amb altres protocols amb complexitat cúbica en el millor cas, per tant s'obté un protocol més eficient i a més la seva seguretat depèn solament de la conjectura del logaritme discret.

El segon protocol està basat en un protocol per a situacions especials quan els elements de la llista de revocació són coprimers, com en la signatura basada en atributs els elements són primers, aquest protocol va ser considerat com a adient. Mentre s'estava demostrant la solidesa del protocol ens vam adonar que el protocol original no era segur perquè vam trobar un atac (que està descrit en el projecte). Per tant, un nou protocol segur va ser dissenyat que serveix per a la signatura basada en atributs.

Finalment, els protocols previs han sigut incorporats a la signatura RSA basada en atributs existent, i les dues signatures resultants han sigut analitzades en termes de l'eficiència del cost de comunicació. El segon protocol es demostra que sempre és més eficient que el primer, fins i tot en el cas d'un sol usuari revocat.

# Resumen

**Palabras clave:** Firma basada en atributos, RSA, privacidad, infalsicabilidad, revocación, Evaluación de polinomios, prueba de conocimiento cero, logaritmo discreto, prueba de no pertenencia.

**MSC2000:** 92A60, 92A62

Una firma basada en atributos con relación a una política de firmas elegida por el firmante, convence al verificador que el firmante tiene un subconjunto de atributos que satisfacen la política de firmas. El verificador no obtendrá ninguna información sobre la identidad del firmante o sobre los atributos que tiene. Este tipo de firmas tiene muchas aplicaciones en situaciones de la vida real que demandan autentificación y privacidad al mismo tiempo. La habilidad para revocar a usuarios que se han comportado mal o han perdido sus atributos y ya no pueden computar más firmas válidas, es muy deseable para aplicaciones de firmas basadas en atributos en la vida real.

En este proyecto, el principal objetivo consiste en estudiar diferentes protocolos de revocación e incorporarlos a un esquema de una firma RSA basada en atributos ya existente. Para poder alcanzar estos objetivos, dos protocolos fueron elegidos entre los disponibles en la literatura existente, teniendo en cuenta la eficiencia de los protocolos existentes y la necesidad de que el protocolo haya sido construido de manera anónima: el usuario no puede revelar su identidad en el momento de probar que él no está en la lista de revocación.

El primer protocolo está basado en un argumento de evaluación de polinomios, y una de sus principales ventajas es que tiene coste de comunicación logarítmica en el número de usuarios revocados en contraste con otros protocolos con complejidad cúbica en el mejor caso, por lo tanto se obtiene un protocolo más eficiente y además su seguridad sólo depende de la conjetura del logaritmo discreto.

El segundo protocolo está basado en un protocolo para situaciones especiales cuando los elementos en la lista de revocación son números coprimos dos a dos, como en la firma basada en atributos los elementos son primos, este protocolo fue considerado como una buena posibilidad. Mientras se estaba demostrando la solidez del protocolo nos dimos cuenta que el protocolo original no era seguro porque encontramos un ataque (que está descrito en el proyecto). Por lo tanto, un nuevo protocolo seguro fue diseñado que sirve para la firma basada en atributos.

Finalmente, los protocolos previos fueron han sido incorporados a la firma RSA basada en atributos existente, y las dos firmas resultantes han sido analizadas en términos de la eficiencia del coste de comunicación. El segundo protocolo se demuestra que siempre es más eficiente que el primero, incluso en el caso de un solo usuario revocado.

# Notation

| | |
|---|---|
| $\mathbb{N}$ | Natural numbers |
| $\mathbb{Z}$ | Integer numbers |
| $N = PQ$ | RSA modulus |
| $\phi(\cdot)$ | Euler Function |
| $\mathbb{Z}_N^*$ | Set of integers less than $N$ and relative prime to $N$ |
| $QR(N)$ | Set of quadratic residues modulo $N$ |
| $\mathsf{RSA.Inst}$ | RSA generated probabilistic algorithm |
| $neg(\cdot)$ | Negligible function |
| $\mathcal{A}$ | Polynomial time adversary |
| $\mathsf{PK}\{\cdot\}$ | Zero-Knowledge Proof of Knowledge |
| $P$ | Prover |
| $V$ | Verifier |
| $H$ | Hash function |
| $S$ | Probability space |
| $c_x$ | Pedersen commitment of $x$ |
| $x \leftarrow_R S$ | Chosen at random according to $S$ |
| $\mathcal{P}$ | Set of Attributes |
| $(\mathcal{P}, \Gamma)$ | Signing Policy |
| $\kappa, \gamma_1, \gamma_2 \in \mathbb{N}$ | Security parameters |
| $\Delta$ | Set of integers in the interval $[2^{\gamma_1} - 2^{\gamma_2} + 1, 2^{\gamma_1} + 2^{\gamma_2} - 1]$ |
| $O()$ | Communication complexity |
| $\mathcal{L}$ | Revocation list |

# Contents

# Chapter 1
# Introduction

Attribute-based cryptography has emerged in the last years as a powerful tool to handle the privacy issues that have appeared in the last decades with the expansion of computer technologies. An attribute-based signature can only be acomplished by an user who holds a subset of attributes that satisfies some policy. The most important property of this type of signatures is that a successful execution will not leak any information about the identity of the user or the attributes that he holds, apart from the fact that these attributes satisfy the given policy.

Attribute-based signatures were introduced expressly in the first version of [**21**]. In an attribute-based signature scheme, each user receives from a master entity a secret key which depends on the attributes that he holds. Later on a user can choose a signing policy (a family of subsets of attributes) satisfied by his attributes, and use his secret key to compute a signature on a message, for this signing policy. The verifier of the signature is positive that some user holding a set of attributes satisfying the signing policy is the author of the signature, but the verifier will not obtain any other information about the actual identity of the signer or the attributes he holds. In addition to the general applications of any attribute-based cryptosystem such as private access control, this specific type of signature have many applications in scenarios where both authentication and privacy properties are desired, such as the leakage of secrets and anonymous polls.

The attribute-based signature scheme that we consider in this project was designed by Herranz [**18**] and is the first attribute-based signature scheme that employs RSA-like keys and operations; particularly, it is the first scheme that does not need bilinear pairings. The main advantage of using RSA attribute-based signature schemes is that it is more desirable to design cryptographic protocols that use keys and operations with RSA so as to benefit from the very specialized hardware and software optimizations available.

Including revocation of users into an attribute-based signature scheme would be imperative so as to use it in real-life scenarios where users want to maintain a certain level of privacy. In cases where a user misbehaves or loses his attributes, it is compulsory to have the option of revoking these users from the system. Because of the privacy property, implementing revocation into attribute-based signatures is not trivial in the least.

Analyzing different methods of revocation and their possible incorporation into the above mentioned RSA attribute-based signature is the main goal of this project. Furthermore, analyzing the efficiency of the resulting protocols and their main advantages and disadvantages will be discussed. In order to do that, different methods and papers on the topic were read through to find the more efficient ones and also to acknowledge the different cryptographic assumptions and concepts needed to carry out this project.

In Chapter 2, a number of cryptographic assumptions and mathematical concepts are introduced that will be used during the dissertation. Moreover, the main security properties and their relation to other properties of attribute-based signatures are laid out. Additionally, the Pedersen and Fujisaki Okamoto commitment schemes which are used several times in this project and some of their main properties are described.

In Chapter 3, the concept of interactive zero-knowledge proof of knowledge is introduced, and the main properties and some examples are described. Also, the concept of non-interactive proof of knowledge is introduced and its obtaining with hash functions is exemplified by means of what is called Schnorr signatures.

In Chapter 4, an extensive description of the algorithms that form an attribute-based signature is presented. Moreover, the RSA attribute-based signature scheme proposed in [**18**] which is the particular one that revocation will be incorporated to is described.

In Chapter 5 we describe the first method to incorporate revocation. It is based in a polynomial evaluation argument developed by Bayer and Groth [**4**], which was chosen because it is the first one to have logarithmic communication cost in the degree of the polynomial. Some additional modifications were added in order to apply it to considered the attribute-based signature scheme.

After designing the first protocol, the original purpose was to implement it practically, but when comparing the protocol with the one in [**15**] based on accumulators, we found that if the revocation list is made public (a feature which is necessary when using [**4**], but which was not used in [**15**]), then a more efficient protocol could be found using the basic idea of accumulators but without the necessity of a master entity that maintains the accumulator. The most efficient way to implement this new idea that we found in the literature was to use a protocol described by Peng and Bao [**23**], but while implementing the protocol we realized that the protocol in [**23**] is not secure: we found a particular attack against it, which is described in Chapter 6.

In Chapter 7, we describe a modification of the protocol in [**23**] in order to achieve the necessary security properties and also some changes to adapt the Peng and Bao protocol to our attribute-based signature scheme, as our revocation list consists of prime numbers and some steps can be simplified.

Finally, in Chapter 8, both protocols described in Chapter 5 and Chapter 7, are incorporated into the RSA attribute-based signature described in Chapter 4. An analysis of the communication cost of both protocols is performed, and for the most

efficient protocol, an efficiency analysis of the resulting attribute-based signature scheme with revocation is done, in order to know how expensive the addition of the revocation property will result. Finally, we compare it with the protocols described in [**15**] to see how much more efficient the new protocol is.

At last, in Chapter 9, the conclusions of the project are outlined and furthermore an option for future work related to this project is described.

# Chapter 2
# Mathematical Settings and Assumptions

In this chapter, some number-theoretic assumptions, mathematical concepts and cryptographic schemes are introduced which will appear in the development of this project.

## 1. Number-Theoretic assumptions

The necessary mathematical parameters are generated using the RSA.Inst probabilistic algorithm. RSA.Inst takes as input a security parameter $\lambda \in \mathbb{Z}^+$, afterwards picks two random prime numbers, $P, Q$, each one being $\lambda/2$-bits long, such that both $p = \frac{P-1}{2}$ and $q = \frac{Q-1}{2}$ are also prime. Let $QR(N) = \{z^2 \bmod N \mid z \in \mathbb{Z}_N^*\} \subset \mathbb{Z}_N^*$ be the set of quadratic residues modulo $N$. $QR(N)$ is a cyclic group of order $pq$. The algorithm RSA.Inst generates at random a generator $g \in QR(N)$ such that $QR(N) = \langle g \rangle$. An execution of this algorithm is denoted as $(P, Q, N, g) \leftarrow \mathsf{RSA.Inst}(1^\lambda)$. The following mathematical problems in $QR(N)$ will be the base of the security analysis of the attribute-based signature scheme and revocation scheme.

The strong RSA assumption was independently proposed by Fujisaki and Okamoto [**16**] and by Barić and Pfitzmann [**3**]. This new definition fortifies the widely accepted RSA assumption that finding $e^{th}$-roots modulo $N$ for any $e > 1$ is hard. The formal definition can be seen hereunder:

**Definition** 1. (*Strong RSA Problem*) Given an RSA modulus $N = PQ$ and a random $x \leftarrow_R \mathbb{Z}_N^*$, the strong RSA problem consist in finding $e > 1$ and $y \in \mathbb{Z}_N^*$, such that $y^e = x \bmod N$

**Assumption** 1. (*The Strong RSA Assumption*) The Strong RSA Assumption affirms that the probability that any algorithm $\mathcal{A}_{sRSA}$ solves the Strong RSA problem in polynomial time is negligible in $\lambda$, the length in bits of the RSA modulus. This implies that the probability decreases, as $\lambda$ increases, faster than the inverse of any

polynomial. Formally, for any probabilistic polynomial time algorithm $\mathcal{A}_{sRSA}$,

$$\mathsf{Pr}\left[\begin{array}{c}(P,Q,N,G) \leftarrow \mathsf{RSA.Inst}(1^\lambda), x \leftarrow_R \mathbb{Z}_N, (y,e) \leftarrow \mathcal{A}_{sRSA}(N,x) : \\ y^e = x (\mathrm{mod}\, N) \wedge 1 < e < N\end{array}\right] = neg(\lambda)$$

where $neg(\lambda)$ is a negligible function.

**Lemma** 1. *For any integer $N$, given integers $u, v \in \mathbb{Z}_N^*$ and $a, b \in \mathbb{Z}$ such that $u^a = v^b \bmod N$ and $\gcd(a,b) = 1$, one can efficiently compute $x \in \mathbb{Z}_N^*$ such that $x^a = v \bmod N$.*

Proof. Since $\gcd(a,b) = 1$, one can find $c, d \in \mathbb{Z}$ using the extended Euclidean algorithm, such that $bd = 1 + ac$. Let $x = (u^d v^{-c} \bmod N)$, then

$$x^a = u^{ad} v^{-ac} = (u^a)^d v^{-ac} = (v^b)^d v^{-ac} = v (\mathrm{mod}\, N)$$

□

**Definition** 2. (*Decisional Diffie-Hellman Problem in $QR_N$, with known factorisation*)

Given the cyclic group $QR_N$ of order $pq$, which is obtained by the set of quadratic residues modulo N; and a random generator $g \in QR_N$. An algorithm $\mathcal{A}_{DDH}$ resolves the Decisional Diffie-Hellman problem in $QR_N$, with known factorisation, if it is capable to distinguish between the two probability distributions $(N, P, Q, g, g^x \bmod N, g^y \bmod N, g^{xy} \bmod N)$ and $(N, P, Q, g, g^x \bmod N, g^y \bmod N, g^z \bmod N)$, where $(P, Q, N, g) \leftarrow \mathsf{RSA.Inst}(1^\lambda)$ and $x, y, z \leftarrow_R \mathbb{Z}_{pq}$.

**Assumption** 2. (*Decisional Diffie-Hellman Assumption*) The DDH Assumption in $QR_N$, with known factorisation, affirms that the success probability of any such algorithm $\mathcal{A}_{DDH}$ is negligible in $\lambda$. Formally, for any algorithm $\mathcal{A}_{DDH}$ running in polynomial time, the *advantage*, as seen below,

$$\left|\mathsf{Pr}\left[\begin{array}{c}1 \leftarrow \mathcal{A}_{DDH}(N, P, Q, g, g^x \bmod N, g^y \bmod N, g^{xy} \bmod N); \\ (P, Q, N, g) \leftarrow \mathsf{RSA.Inst}(1^\lambda); x, y \leftarrow_R \mathbb{Z}_{pq}.\end{array}\right](\lambda) - \mathsf{Pr}\left[\begin{array}{c}1 \leftarrow \mathcal{A}_{DDH}(N, P, Q, g, g^x \bmod N, g^y \bmod N, g^z \bmod N); \\ (P, Q, N, g) \leftarrow \mathsf{RSA.Inst}(1^\lambda); x, y, z \leftarrow_R \mathbb{Z}_{pq}.\end{array}\right](\lambda)\right|$$

is negligible in the security parameter $\lambda$.

The following result is proved in the full version of [?]: the Decisional Diffie-Hellman problem in $QR_N$, with known factorisation, is equivalent to the Decisional Diffie-Hellman problem in a cyclic subgroup of $QR_N$ of either prime order $p$ or prime order $q$. The Decisional Diffie-Hellman problem in a cyclic group of big prime order is considered to be computationally hard, and therefore the Decisional Diffie-Hellman Assumption in $QR_N$, with known factorisation, makes perfect sense.

**Definition** 3. (*The Discrete Logarithm Problem*) Given a group $QR_N$ and a generator $g$, produced by running the algorithm $\mathsf{RSA.Inst}(1^\lambda)$, and given a random element $h \in QR_N$, the discrete logarithm problem in $QR_N$ consists in finding an integer $x$ such that $h = g^x \bmod N$.

**Assumption** 3. (*The Discrete Logarithm Assumption*) The discrete logarithm assumption holds for $QR_N$ if for all non-uniform probabilistic polynomial time algorithms $\mathcal{A}_{DLA}$,

$$\mathsf{Pr}\left[(P, Q, N, g) \leftarrow \mathsf{RSA.Inst}(1^\lambda), h \leftarrow QR_N, x \leftarrow \mathcal{A}_{DLA}(N, g, h) : x \in \mathbb{Z} \wedge g^x = h \bmod N\right] = neg(\lambda)$$

where $(P, Q, N, g) \leftarrow \mathsf{RSA.Inst}(1^\lambda)$ is an algorithm that generates a RSA modulus and $neg(\lambda)$ is a negligible function. The hardness of finding discrete logarithms relies on the type of the group. In our case $N = PQ$, where $P, Q$ are safe primes, because they are obtained in a way that $P = 2p+1$, $Q = 2q+1$, where $p, q$ are also prime numbers. Using this type of prime numbers the discrete logarithm problem in $QR_N$ cannot be efficiently solved.

**Lemma 2.** *Under the Strong RSA Assumption, if the adversary $\mathcal{F}$ is able to obtain values $L \in QR(N)$, $a_1, a_2, v_1, v_2 \in \mathbb{Z}$ such that $L^{a_1-a_2} = g^{v_1-v_2} \bmod N$, then it must hold $\frac{v_1-v_2}{a_1-a_2} \in \mathbb{Z}$.*

Proof. Lemma 2. Let $d = \gcd(v_1 - v_2, a_1 - a_2)$ be the largest integer dividing both $v_1 - v_2$ and $a_1 - a_2$. This means (Euclides algorithm) that there exist integers $\rho_a, \rho_v \in \mathbb{Z}$ such that $\rho_a(a_1 - a_2) + \rho_v(v_1 - v_2) = d$. Let us assume, for the sake of contradiction, that $\frac{v_1-v_2}{a_1-a_2} \notin \mathbb{Z}$. Therefore, it must hold $a_1 - a_2 > d$ and, so, $e := \frac{a_1-a_2}{d} > 1$.

Now we have that $g = g^{\frac{\rho_a(a_1-a_2)+\rho_v(v_1-v_2)}{d}} = g^{\rho_a e} L^{\rho_v e} = (g^{\rho_a} L^{\rho_v})^e \bmod N$. Therefore, we would solve an instance of the Strong RSA problem with input $\omega = g$, because $g^{\rho_a} L^{\rho_v}$ is an $e$-th root of $g$, for $e = \frac{a_1-a_2}{d} > 1$. This would contradict the Strong RSA Assumption, and so we can conclude that $\frac{v_1-v_2}{a_1-a_2} \in \mathbb{Z}$. □

## 2. The Euclidean Algorithm

The Euclidean algorithm gives a method to efficiently compute the greatest common divisor of two integers $a$ and $b$.
Considering that the divisors of an integer $a$ are the same as those of $-a$, it is sufficient to consider the case in which $a$ and $b$ are both positive.
The method consists in performing the following divisions. It can be presumed that in the first $n$ divisions the remainder is positive, while in the last one it is zero:

$$\begin{array}{lcl}
1. & a = bq_1 + r_1, & 0 < r_1 < b, \\
2. & b = r_1 q_2 + r_2, & 0 < r_2 < r_1, \\
3. & r_1 = r_2 q_3 + r_3, & 0 < r_3 < r_2, \\
\vdots & \vdots & \vdots \\
i+2. & r_i = r_{i+1} q_{i+2} + r_{i+2}, & 0 < r_{i+2} < r_{i+1}, \\
\vdots & \vdots & \vdots \\
n-1. & r_{n-3} = r_{n-2} q_{n-1} + r_{n-1}, & 0 < r_{n-1} < r_{n-2}, \\
n. & r_{n-2} = r_{n-1} q_n + r_n, & 0 < r_n < r_{n-1}, \\
n+1 & r_{n-1} = r_n q_{n+1} + 0. &
\end{array}$$

Dividing in this way, a remainder equal to zero is found within at most $b$ divisions, since $b > r_1 > r_2 > r_3 > \cdots > r_n$ is a strictly decreasing sequence of non-negative integers. It can be noted that the common divisors of $a$ and $b$ are the same ones as the common divisors of $b$ and $r_1$, actually, if an integer divides both $a$ and $b$, it divides each multiple of $b$, and the difference $a - bq_1$ which is $r_1$. On the other hand, if an integer divides $b$ and $r_1$, it also divides $a = bq_1 + r_1$. Using the second equation, the common divisors of $b$ and $r_1$ are the common divisors of $r_1$ and $r_2$.

Proceeding with the following equations, it can be found that the common divisors of $a$ and $b$ are the common divisors of $r_{n-1}$ and $r_n$. Since $r_{n-1}$ is a multiple of $r_n$, the common divisors of $r_{n-1}$ and $r_n$ are the divisors of $r_n$.
The last remainder in the sequence of the divisions is defined as $d = r_n$. $d$ is a common divisor of $a$ and $b$. Furthermore, it is the greatest among the common divisors of $a$ and $b$, given that if $d'$ divides both $a$ and $b$ then $d'$ divides $d$ as seen before. $d$ is define as the greatest common divisor and the symbol $\gcd(a, b)$ denotes it. If $\gcd(a, b) = 1$, the numbers $a$ and $b$ are coprime, since they do not have non trivial common divisors.

**2.1. Bézout's identity.** The Euclidean algorithm gives a way of proving that the following relation holds:

$$\gcd(a, b) = ka + lb$$

with $k$ and $l$ suitable integers. The proof of this identity consists in showing that all the remainders of the successive divisions can be written as combinations of $a$ and $b$. Given the following equations,

$$r_1 = a - bq_1$$
$$r_2 = a - bq_2$$
$$\vdots$$
$$r_n = r_{n-2} - r_{n-1}q_n$$

It can be noted that,

$$r_2 = b - r1q2 = b - (a - bq_1)q_2 = (-q_2)a + (1 + q_1q_2)b$$

This means, that $r_1$ and $r_2$ can be written as combinations of $a$ and $b$. Thus $r_3$ being a combination with integer coefficients of $r_1$ and $r_2$, is also a combination with integer coefficients of $a$ and $b$. At last, $d = r_n$ is a combination with integer coefficients of $r_{n-1}$ and $r_{n-2}$, and so of $a$ and $b$.

PROPOSITION 3. *Let $a$ and $b$ be two positive integers. They are coprime if and only if there exist two integers $k, l$ such that*

$$ka + lb = 1$$

PROOF. If $a$ and $b$ are coprime, $\gcd(a, b) = 1$ and the affirmation follows from the Bézout's identity.
On the other hand, lets suppose $ka + lb = 1$ holds. If $d$ is a common divisor of $a$ and $b$, then $d$ divides $ka + lb$ too, and so divides 1. Hence $d = 1$ or $d = -1$, ergo $a$ and $b$ are relatively prime. □

# 3. The Pedersen and Fujisaki-Okamoto Commitments Scheme

The Pedersen commitment scheme [**22**], and the variant of Fujisaki-Okamoto for groups with unknown order, are used several times in this project. The security the scheme provides is based on the discrete logarithm assumption. Pedersen described his scheme in groups of prime and public order. Here we detail the variant proposed

by Fujisaki-Okamoto [**16**], for groups of unknown order.

The probabilistic algorithm $G(1^\lambda)$ with security parameter $\lambda$, chooses two random prime numbers, $P, Q$ of length $\lambda/2$-bits long, where $p = \frac{P-1}{2}$ and $q = \frac{Q-1}{2}$ are also prime. Let $QR(N) = \{z^2 \bmod N | z \in \mathbb{Z}_N^*\} \subset \mathbb{Z}_N^*$ be the set of quadratic residues of modulo $N$. $QR(N)$ is a cyclic group of order $pq$ and the algorithm generates random generators $g, h \in QR(N)$ such that $QR(N) = \langle g \rangle$, $QR(N) = \langle h \rangle$. To commit to $x \in \mathbb{Z}_N$ the committer picks randomness $r \in \mathbb{Z}_N$ and computes

$$c_x = g^x h^r \bmod N$$

Both the Pedersen and Fujisaki Okamoto commitment schemes are computationally binding under the discrete logarithm assumption, that is, a non-uniform probabilistic polynomial time adversary $\mathcal{A}$, given $(g, h, c_x)$, cannot find $x \neq x'$ such that $c_x$ is a valid commitment for both $x$ and $x'$.

One of the main properties of both commitment schemes is that they are homomorphic. For all $x, y$ and $r, s$ the following equation holds

$$c_x c_y = g^x h^r g^y h^s = g^{x+y} h^{r+s} = c_{x+y} \bmod N$$

# Chapter 3
# Proofs of Knowledge

## 1. Interactive Proofs of Knowledge

In cryptography, a proof of knowledge is an interactive proof in which the *prover* succeeds 'convincing' a *verifier* that it knows something, for instance, a solution of an equation. The trivial solution consists in the *prover* sending to the *verifier* what he knows, and the *verifier* authenticates it. But in some cases, the *prover* wants to keep his information private, and just wants to prove the fact that he knows it; this method receives the name of zero-knowledge proof of knowledge.
A so called zero-knowledge proof of knowledge allows a prover to demonstrate the knowledge of a secret with respect to some public information such that no other information is revealed in the process.
Some examples of zero-knowledge proofs of knowledge are:

- Given two graphs $G, H$, the *prover* wants to prove to the verifier that $G$ is isomorphic to $H$, without revealing the isomorphism.
- Given a cyclic group $\mathbb{G} = \langle \tilde{g} \rangle$ and $y \in \mathbb{G}$, the *prover* wants to prove that he knows $x \in \mathbb{Z}$ such that $\tilde{g}^x = y$, which is the discrete logarithm of $y$ in basis $\tilde{g}$, without revealing $x$.
- Given a public key $pk$, the *prover* wants to prove that he knows the matching secret key $sk$, without revealing it. This is known as identification protocol.

### 1.1. Properties of a Zero-Knowledge Proof of Knowledge.

For any given zero-knowledge proof of knowledge, it satisfies the following properties. Let $L$ be a language in nondeterministic polynomial-time, and given $\alpha \in L$, let $W(\alpha)$ be the set of witnesses of the fact that $\alpha \in L$. The relation $R$ can be defined as $R = \{(\alpha, \omega) \ : \ \alpha \in L, \omega \in W(\alpha)\}$.
Given $L$ and $\alpha$ which are public. The secret input for the *prover* may be the witness $\omega$. The properties are:

- **Completeness:** if the *prover* knows $\omega$ such that $(\alpha, \omega) \in R$, then the *verifier* always accepts the *prover*'s proof.

- **Proof of Knowledge:** if the *prover*'s proofs for $\alpha$ are accepted with probability $\epsilon$, then it is possible to extract a witness $\omega$ such that $(\alpha, \omega) \in R$ with probability $\geq \epsilon$, given oracle access to the *prover*.
- **Zero-Knowledge:** if the *prover* knows $\omega$ such that $(\alpha, \omega) \in R$, then even a malicious *verifier* $V'$ obtains no new information on $\omega$ from the execution of the protocol. This means that for any such malicious *verifier*, there exists a Simulator algorithm $\mathcal{S}$ such that:

$$\mathcal{S}(\alpha) \approx \text{Outputs}[P(\alpha, \omega) \leftrightarrow V'(\alpha)].$$

The description of a zero-knowledge proof of knowledge for the first two examples can be seen below:

- **A Zero-Knowledge Proof of Knowledge Protocol for Graph Isomorphism**
  $L_{G_0}$ are the graphs isomorphic to $G_0$. A graph $G_1 \in L_{G_0}$ if and only if there exists an isomorphism $\pi$ such that $\pi(G_1) = G_0$. Therefore, $R = \{(G_1, \pi) : G_1 \in L_{G_0}, \pi(G_1) = G_0\}$. Suppose the *prover* knows $\pi$ such that $\pi(G_1) = G_0$.
  (1) The *prover* chooses isomorphism $\rho$ at random and sends $H = \rho(G_0)$ to $V$.
  (2) The *verifier* chooses random bit $b \in \{0, 1\}$ and sends $b$ to the *prover*.
  (3) If $b = 0$, $P$ replies $\psi = \rho$. If $b = 1$, the *prover* replies $\psi = \rho \circ \pi$.
  (4) The *verifier* outputs 1 if and only if $\psi(G_b) = H$.

  If this process is repeated $n$ times in parallel, the cheating probability of a dishonest *prover* is $2^{-n}$. This process fulfils the subsequent properties:

  - **Completeness:** trivial.
  - **Proof of Knowledge:** extractor runs the *prover* until step 3, with $b = 0$, and obtains $\psi_0$. Then rewinds back to step 2, chooses $b = 1$ and lets the *prover* output $\psi_1$. The extracted witness $\pi = \psi_0^{-1} \circ \psi_1$ satisfies $\pi(G_1) = G_0$.
  - **Zero-Knowledge:** a transcript $(H, b, \psi)$ of the protocol between the *prover* and a malicious *verifier* can be simulated by $\mathcal{S}$ as follows:
    (1) choose a random permutation $\psi$,
    (2) choose bit $b \in \{0, 1\}$ with the same distribution as the malicious *verifier* does,
    (3) compute $H = \psi(G_b)$.

- **A Zero Knowledge Proof of Knowledge Protocol for Discrete Logarithm**
  $L_{(\mathbb{G}, \tilde{g})}$ are the elements in the cyclic group $\mathbb{G} = \langle \tilde{g} \rangle$. A witness for $y \in G$ is $x \in \mathbb{Z}$ such that $\tilde{g}^x = y$. Therefore, $R = \{(y, x) : y \in \mathbb{G}, \tilde{g}^x = y\}$. $\mathbb{G}$ has public prime order $\tilde{p}$. Suppose the *prover* knows $x \in \mathbb{Z}_{\tilde{p}}$ such that $\tilde{g}^x = y$.

  (1) The *prover* chooses $r \in_R \mathbb{Z}_{\tilde{p}}^*$ at random and sends $R = \tilde{g}^r$ to the *verifier*.
  (2) The *verifier* chooses $h \in_R \mathbb{Z}_{\tilde{p}}$ at random and sends $h$ to the *prover*.
  (3) The *prover* computes $s = r + x \cdot h \bmod \tilde{p}$ and sends $s$ to the *verifier*.
  (4) The *verifier* outputs 1 if and only if $\tilde{g}^s = R \cdot y^h$.

  The previous protocol satisfies this properties:

- **Completeness:** trivial.
- **Proof of Knowledge:** the extractor runs the *prover* until step 3, with random $h$, and obtains $s$. Then rewinds back to step 2, chooses a different $h' \neq h$ and lets the *prover* output $s'$. The extracted witness $x = \frac{s-s'}{h-h'} \bmod p$ satisfies $\tilde{g}^x = y$.
- **Zero-Knowledge:** a transcript $(R, h, s)$ of the protocol between the *prover* and a malicious *verifier* can be simulated by $\mathcal{S}$ as follows:
  (1) choose at random $s \in \mathbb{Z}_{\tilde{p}}$,
  (2) choose $h \in \mathbb{Z}_{\tilde{p}}$ with the same distribution as the malicious *verifier* does,
  (3) compute $R = \tilde{g}^s \cdot y^{-h}$.

# 2. From Interactive 3-Round Proofs to Non-Interactive Proofs

In the previous section, two examples of interactive zero-knowledge proofs of knowledge with 3 rounds of communication: commitment, challenge, final answer were explained. It can be noted that the only task of $V$ was to choose a random challenge.
If the prover $P$ chooses this random challenge by himself, then the verifier $V$ does not participate, and the protocol becomes non-interactive. This new type of proof will be known as a zero-knowledge non-interactive proof of knowledge.

But in order to prevent success from a dishonest prover $P$, the random challenge must be linked to the commitment. For instance, challenge= $H$(commitment), where $H$ is a good hash function that behaves as a random oracle, heuristically. Besides, more inputs can be added to $H$, for example, a message can be signed.

**2.1. Application to Discrete Logarithm Zero-Knowledge Proofs of Knowledge: Schnorr Signatures.** Given $H$, which is a hash function, and a commitment $c = \tilde{g}^r$, the value $h = H(c, \cdot)$ is a challenge, and $s = r + x \cdot h \bmod \tilde{p}$ is the final answer, the resulting signature scheme can be called $\Sigma$ and was introduced by Schnorr [**25**]. The description of the signature scheme can be seen hereunder:

- **Key generation,** $(sk, pk) \leftarrow \Sigma.\mathbf{KG}(1^k)$ **:** The group $\mathbb{G} = \langle \tilde{g} \rangle$ has prime order $\tilde{p}$, with $k$ bits. $x \in_R \mathbb{Z}_{\tilde{p}}^*$ is taken and $y = \tilde{g}^x$ computed. A good hash function is chosen $H : \{0,1\}^* \to \mathbb{Z}_{\tilde{p}}$.
  The resulting public key is $pk = (\mathbb{G}, \tilde{g}, y, H)$, and the secret key is $sk = x$.
- **Signature,** $\sigma \leftarrow \Sigma.\mathbf{Sign}(m, sk)$ **:** in order to sign a message $m \in \{0,1\}^*$ the following steps need to be fulfilled.
  (1) $r \in_R \mathbb{Z}_{\tilde{p}}^*$ is chosen at random and $c = \tilde{g}^r$ is computed,
  (2) $h = H(m, c)$ is computed,
  (3) finally $s = r + x \cdot h \bmod \tilde{p}$ is computed and the signature is defined as $\sigma = (c, s)$.

- **Verification,** 1 **or** $0 \leftarrow \Sigma.\mathbf{Vfy}(m, \sigma, pk)$ **:** given a tuple $(m, c, s)$ of a message or a signature, the value 1 is returned if and only if
$$\tilde{g}^s = c \cdot y^{H(m,c)}.$$

The proof of the Schnorr signature scheme in proved in the random oracle model, and is proved by reduction to the hardness of the discrete logarithm problem. The key point of the proof is the forking lemma; once $\mathcal{A}$ forges a signature $\sigma = (c, s)$ on $m$, such that $h = H(m, c)$, if the attacker $\mathcal{A}$ is replied with the same randomness until $(m, c)$ is queried to the random oracle, then the answer $h' \neq h$ is provided. With some probability, the second execution gives another valid signature $\sigma' = (c, s')$ on $m$ and from the two forgeries, it is easy to extract the discrete logarithm of $y$.

# Chapter 4
# Attribute-Based Signatures from RSA

In this chapter, the concept and protocols of attribute-based signatures will be introduced. Furthermore, a complete description of the attribute-based signature from RSA developed by Herranz in [**18**], in which revocation of users is being incorporated, is done in this chapter.

## 1. Attribute-Based Signatures

This first section focuses on describing the concept and protocols of attribute-based signatures. These specific protocols were developed in [**18**] over the protocols of [**?**] in order to deal explicitly with the identity of users. An attribute-based signature is conjoint to a determined signing policy $(\mathcal{P}, \Gamma)$: a set $\mathcal{P}$ of attributes and a monotone increasing family $\Gamma \subset 2^{\mathcal{P}}$ of subsets of $\mathcal{P}$. A valid signature entails that a signer possessing all the attributes of some of the subsets in $\Gamma$ is the author of the signature. The monotonicity property secures that $S_1 \subset S_2, S_1 \in \Gamma \Rightarrow S_2 \in \Gamma$. A simple example of such a monotone increasing family of subsets is the threshold case. Given a $(\ell, n)$-threshold signing policy, with a set $\mathcal{P}$ which contains $n$ attributes, and $\Gamma = \{S \subset \mathcal{P} \ : \ |S| \geq \ell\}$, a verifier authenticates a threshold attribute-based signature if he is assured that the author of the signature holds at least $\ell$ of the attributes included in the set $\mathcal{P}$.

### 1.1. Syntactic Definition.

An attribute-based signature scheme consists of four probabilistic polynomial-time algorithms:

- $\mathsf{Setup}(1^\lambda)$. The setup algorithm takes as input a security parameter $\lambda$ and outputs the initial public parameters $\mathsf{pms}$ and the master secret key $\mathsf{msk}$ for the master entity. Within the public parameters appear the possible universe of attributes $\tilde{\mathcal{P}} = \{\mathsf{at}_1, \ldots, \mathsf{at}_n\}$.
- $\mathsf{KeyGen}(\mathsf{id}, S, \mathsf{msk}, \mathsf{pms})$. The key generation algorithm takes as input the master secret key $\mathsf{msk}$, the public parameters $\mathsf{pms}$, furthermore, an identity $\mathsf{id}$ that satisfies a set of attributes $S \subset \tilde{\mathcal{P}}$ is required. The output is a private key $\mathsf{sk}_{\mathsf{id},S}$.

- $\mathsf{Sign}(m, \mathcal{P}, \Gamma, \mathsf{sk}_{\mathsf{id},S}, \mathsf{pms})$. The signing algorithm takes as input a message $m$, a signing policy $(\mathcal{P}, \Gamma)$ where $\mathcal{P} \subset \tilde{\mathcal{P}}$ and $\Gamma \subset 2^{\mathcal{P}}$, a secret key $\mathsf{sk}_{\mathsf{id},S}$ and the public parameters $\mathsf{pms}$, and outputs a signature $\sigma$.
- $\mathsf{Verify}(\sigma, m, \mathcal{P}, \Gamma, \mathsf{pms})$. The verification algorithm takes as input the signature $\sigma$, the message $m$, the signing policy $(\mathcal{P}, \Gamma)$ and the public parameters $\mathsf{pms}$. The outputs are 1 if the signature is accepted or 0 if it is rejected. signature.

Such a scheme fulfils the correctness' property, if for a signature $\sigma$ with respect to a signing policy $(\mathcal{P}, \Gamma)$ that is calculated by using $\mathsf{sk}_{\mathsf{id},S}$ such that $S \in \Gamma$, the signature $\sigma$ is always accepted as valid by the verification protocol.

### 1.2. Security Definitions.

*Privacy.* The privacy property entails that given a valid signature, nobody can obtain any information about the real author of the signature. That is to say, given two pairs $(\mathsf{id}_0, S_0)$ and $(\mathsf{id}_1, S_1)$, with $S_0, S_1 \subset \mathcal{P}^*$, and a valid signature $\sigma \leftarrow \mathsf{Sign}(m, \mathcal{P}, \Gamma, \mathsf{sk}_{\mathsf{id}_b,S_b}, \mathsf{pms})$ for a signing policy $\Gamma$ such that $S_0, S_1 \in \Gamma$, nobody would be able to guess the bit $b$ with probability significantly bigger than $1/2$. The privacy property is formally defined via the following experiments $\mathbf{Exp}^{\mathsf{priv}}_{b,\mathcal{B}}(\lambda)$, for $b = 0, 1$, involving an adversary $\mathcal{B}$.

$\underline{\mathbf{Exp}^{\mathsf{priv}}_{b,\mathcal{B}}(\lambda)}$
$(\mathsf{pms}, \mathsf{msk}) \leftarrow \mathsf{Setup}(1^\lambda)$
$(m, \mathcal{P}, \Gamma, \mathsf{id}_0, S_0, \mathsf{sk}_{\mathsf{id}_0,S_0}, \mathsf{id}_1, S_1, \mathsf{sk}_{\mathsf{id}_1,S_1}, st_1) \leftarrow \mathcal{B}(\mathsf{pms}, \mathsf{msk})$
Verify that $\mathsf{sk}_{id_i,S_i}$ is a valid secret key for $S_i$, for $i = 0, 1$
Verify that $S_0 \cap \mathcal{P} \in \Gamma$ and $S_1 \cap \mathcal{P} \in \Gamma$
$\sigma^* \leftarrow \mathsf{Sign}(m, \mathcal{P}, \Gamma, \mathsf{sk}_{\mathsf{id}_b,S_b}, \mathsf{pms})$
$b' \leftarrow \mathcal{B}(\sigma^*, \mathsf{pms}, \mathsf{msk}, st_1)$
Output $b'$

The advantage of $\mathcal{B}$ in breaking the privacy property is defined as

$$\mathsf{Adv}^{\mathsf{priv}}_{\mathcal{B}}(\lambda) = \left| \mathsf{Pr}[\mathbf{Exp}^{\mathsf{priv}}_{0,\mathcal{B}}(\lambda) = 1] - \mathsf{Pr}[\mathbf{Exp}^{\mathsf{priv}}_{1,\mathcal{B}}(\lambda) = 1] \right|.$$

**Definition** 4. An attribute-based signature scheme is private if, for any adversary $\mathcal{B}$ that runs in polynomial time, the advantage $\mathsf{Adv}^{\mathsf{priv}}_{\mathcal{B}}(\lambda)$ is negligible in the security parameter $\lambda$.

Seeing that the adversary $\mathcal{B}$ can obtain the master secret key, other properties are implied by the privacy property, such as anonymity and unlinkability. The anonymity property means that given a valid signature, identifying the actual signer is computationally hard and the unlinkability property implies that deciding whether two different valid signatures were computed by the same user is computationally hard.

*Unforgeability.* An attribute-based signature scheme must fulfil the property of existential unforgeability against chosen message and signing policy attacks. Such

property is defined by the following experiment $\mathbf{Exp}^{\mathsf{unf}}_{\mathcal{F}}(\lambda)$ involving an adversary $\mathcal{F}$.

$\underline{\mathbf{Exp}^{\mathsf{unf}}_{\mathcal{F}}(\lambda)}$
$(\mathsf{pms}, \mathsf{msk}) \leftarrow \mathsf{Setup}(1^\lambda)$
$(\sigma^*, m^*, \mathcal{P}^*, \Gamma^*) \leftarrow \mathcal{F}^{\mathsf{KeyGen}(\cdot,\mathsf{msk},\mathsf{pms}),\ \mathsf{Sign}(\cdot,\mathsf{pms})}(\mathsf{pms})$
Output 1 if the three following statements are true:
(i) $\mathsf{Verify}(\sigma^*, m^*, \mathcal{P}^*, \Gamma^*, \mathsf{pms})$ returns 1;
(ii) $\mathcal{F}$ has not made any secret key query $(\mathsf{id}, S)$ such that $S \cap \mathcal{P}^* \in \Gamma^*$;
(iii) $(m^*, \mathcal{P}^*, \Gamma^*, \sigma^*)$ is not the result of any signature query from $\mathcal{F}$.
Otherwise, output 0

The advantage of $\mathcal{F}$ in breaking the unforgeability of the scheme is defined as $\mathsf{Adv}^{\mathsf{unf}}_{\mathcal{F}}(\lambda) = \mathsf{Pr}[\mathbf{Exp}^{\mathsf{unf}}_{\mathcal{F}}(\lambda) = 1]$. We stress that $\mathcal{F}$ is allowed to make adaptive queries for secret keys of pairs $(\mathsf{id}, S)$ of his choice, and adaptive signing queries for tuples $(m, \mathcal{P}, \Gamma)$ of his choice, where $\Gamma \subset 2^{\mathcal{P}}$. The last kind of queries are answered by choosing a random subset $S \subset \mathcal{P}$ with $S \in \Gamma$, and then by running $\mathsf{sk}_{\mathsf{id},S} \leftarrow \mathsf{KeyGen}(\mathsf{id}, S, \mathsf{msk}, \mathsf{pms})$ and $\sigma \leftarrow \mathsf{Sign}(m, \mathcal{P}, \Gamma, \mathsf{sk}_{\mathsf{id},S}, \mathsf{pms})$.

**Definition** 5. An attribute-based signature scheme is unforgeable if, for any adversary $\mathcal{F}$ that runs in polynomial time, the advantage $\mathsf{Adv}^{\mathsf{unf}}_{\mathcal{F}}(\lambda)$ is negligible in the security parameter $\lambda$.

As well as in the privacy definition, the unforgeability definition implies the collusion resistance property. A group of colluding users (even if it is comprised of all the users) that pool together their secret keys, will not be able to sign messages for a signing policy that none of the attribute sets of these users satisfies.

## 2. Existing Attribute-Based Scheme for a Threshold Signing Policy

In this section, it will be explained the main aspects of this signature scheme that will be used in Chapter 7 once the revocation method is implemented. The security of this scheme (privacy and unforgeability of the signatures) is established under the Strong RSA Assumption and the Decisional Diffie-Hellman Assumption in $QR_N$ with known factorisation (Assumptions 1 and 2 in Chapter 2).
The original attribute-based signature scheme was developed in [**18**]. For simplicity of explanation, the signature scheme is developed in the case of threshold signing policies, a pair $(\mathcal{P}, \Gamma)$ will be depicted as $(\mathcal{P}, \ell)$, where $1 \leq \ell \leq |\mathcal{P}|$. The four existing algorithms (Setup, KeyGen, Sign, Verify), are described hereunder.

$\mathsf{Setup}(1^\lambda)$. The setup algorithm begins by running $(P, Q, N, g) \leftarrow \mathsf{RSA.Inst}(1^\lambda)$, where $N = PQ$, $P = 2p+1$ and $Q = 2q+1$. Choose security parameters $\kappa, \gamma_1, \gamma_2 \in \mathbb{N}$ and $\epsilon \in \mathbb{R}$, $\epsilon > 1$, such that $\gamma_1 - 2 > \epsilon(\gamma_2 + \kappa) > \lambda$. We denote as $\Delta$ the set of integers in the interval $[2^{\gamma_1} - 2^{\gamma_2} + 1, 2^{\gamma_1} + 2^{\gamma_2} - 1]$.

A prime number $q'$ in the interval $[2^{\kappa-1}, 2^{\kappa}]$ is chosen. Two cryptographic hash functions $H_0 : \{0,1\}^* \to QR(N)$ and $H_1 : \{0,1\}^* \to \mathbb{Z}_{q'}^*$ are also chosen. Finally, the global set of attributes $\tilde{\mathcal{P}}$ has to be chosen.

The public parameters are $\mathsf{pms} = (\kappa, \gamma_1, \gamma_2, \epsilon, \Delta, \lambda, N, g, H_0, H_1, q', \tilde{\mathcal{P}})$, whereas the master secret key is $\mathsf{msk} = (P, Q)$.

$\mathsf{KeyGen}(\mathsf{id}, S, \mathsf{msk}, \mathsf{pms})$. The key generation algorithm takes as input an identity $\mathsf{id}$, a subset of attributes $S \subset \tilde{\mathcal{P}}$ satisfied by $\mathsf{id}$, the master secret key $\mathsf{msk}$ and the public parameters $\mathsf{pms}$. The master entity chooses at random a prime number $e \xleftarrow{R} \Delta$ such that $\gcd(e, pq) = 1$ and, for each $\mathsf{at}_i \in S$, computes the value $\mathsf{sk}_i = H_0(\mathsf{at}_i)^{1/e} \bmod N$ (using the knowledge of the prime numbers $P, Q$). The global secret key is $\mathsf{sk}_{\mathsf{id},S} = (e, \{\mathsf{sk}_i\}_{\mathsf{at}_i \in S})$.

$\mathsf{Sign}(m, \mathcal{P}, \ell, \mathsf{sk}_{\mathsf{id},S}, \mathsf{pms})$. The signing algorithm takes as input a message $m$, a set of attributes $\mathcal{P} \subset \tilde{\mathcal{P}}$, a threshold $\ell$, a secret key $\mathsf{sk}_{\mathsf{id},S} = (e, \{\mathsf{sk}_i\}_{\mathsf{at}_i \in S})$ and the public parameters $\mathsf{pms}$. The algorithm selects a minimally authorised set $S'$, this is, a subset of $S \cap \mathcal{P}$ of cardinality exactly $\ell$. Without loss of generality, let us assume $\mathcal{P} = \{\mathsf{at}_1, \ldots, \mathsf{at}_n\}$ and $S' = \{\mathsf{at}_1, \ldots, \mathsf{at}_\ell\}$. To generate the signature, it proceeds as follows:

(1) Choose $h \xleftarrow{R} QR(N)$ and $r \xleftarrow{R} \mathbb{Z}_N$. Compute $A = g^r \bmod N$, $B = g^e \cdot h^r \bmod N$.

(2) For $j = \ell + 1, \ldots, n$, choose $c_j \xleftarrow{R} \mathbb{Z}_{q'}$, $C_j, Z_j \xleftarrow{R} QR(N)$, $u_j \xleftarrow{R} \pm\{0,1\}^{\epsilon(\gamma_2+\kappa)}$, $v_j \xleftarrow{R} \pm\{0,1\}^{\epsilon(\lambda+\kappa)}$, $w_j \xleftarrow{R} \pm\{0,1\}^{\epsilon(\gamma_1+\lambda+\kappa+1)}$, and compute the values $D_j = \frac{A^{u_j - c_j 2^{\gamma_1}}}{g^{w_j}} \bmod N$, $E_j = \frac{g^{v_j}}{A^{c_j}} \bmod N$, $F_j = g^{u_j - c_j 2^{\gamma_1}} \cdot h^{v_j} \cdot B^{c_j} \bmod N$ and $G_j = \frac{C_j^{u_j - c_j 2^{\gamma_1}} \cdot H_0(\mathsf{at}_j)^{c_j}}{Z_j^{w_j}} \bmod N$.

(3) For $i = 1, \ldots, \ell$, choose $Z_i \xleftarrow{R} QR(N)$, $\alpha_i \xleftarrow{R} \pm\{0,1\}^{\epsilon(\gamma_2+\kappa)}$, $\beta_i \xleftarrow{R} \pm\{0,1\}^{\epsilon(2\lambda+\kappa)}$, $\delta_i \xleftarrow{R} \pm\{0,1\}^{\epsilon(\gamma_1+2\lambda+\kappa+1)}$, and compute the values $C_i = \mathsf{sk}_i \cdot Z_i^r \bmod N$, $D_i = \frac{A^{\alpha_i}}{g^{\delta_i}} \bmod N$, $E_i = g^{\beta_i} \bmod N$, $F_i = g^{\alpha_i} \cdot h^{\beta_i} \bmod N$ and $G_i = \frac{C_i^{\alpha_i}}{Z_i^{\delta_i}} \bmod N$.

(4) Compute the hash value $c = H_1(m, \mathcal{P}, \ell, h, A, B, \{C_i, D_i, E_i, F_i, G_i, Z_i\}_{\mathsf{at}_i \in \mathcal{P}})$.

(5) Find the (only) polynomial $f(x) \in \mathbb{Z}_{q'}[X]$ with degree at most $n - \ell$ such that $f(0) = c \bmod q'$ and $f(j) = c_j \bmod q'$ for all $j = \ell + 1, \ldots, n$.

(6) For $i = 1, \ldots, \ell$, compute $c_i = f(i) \bmod q'$ and then compute the values $u_i = \alpha_i - c_i \cdot (e - 2^{\gamma_1})$, $v_i = \beta_i - c_i \cdot r$ and $w_i = \delta_i - c_i \cdot e \cdot r$, over the integers.

The resulting signature is $\sigma = (f(x), h, A, B, \{(C_i, u_i, v_i, w_i, Z_i)\}_{\mathsf{at}_i \in \mathcal{P}})$.

$\mathsf{Verify}(\sigma, m, \mathcal{P}, \ell, \mathsf{pms})$. The verification algorithm takes as input a message $m$, the signature $\sigma$ of $m$, the threshold signing policy $(\mathcal{P}, \ell)$, with $n = |\mathcal{P}|$, and the public parameters $\mathsf{pms}$. It proceeds as follows:

(1) Verify that the degree of $f(x)$ is at most $n - \ell$. For all $\mathsf{at}_i \in \mathcal{P}$, verify that $u_i \in \pm\{0,1\}^{\epsilon(\gamma_2+\kappa)}$, $v_i \in \pm\{0,1\}^{\epsilon(\lambda+\kappa)}$ and $w_i \in \pm\{0,1\}^{\epsilon(\gamma_1+\lambda+\kappa+1)}$. Return 0 if this is not the case.
(2) For all $\mathsf{at}_i \in \mathcal{P}$, compute $c_i = f(i)$ and then compute the values $D_i = \frac{A^{u_i - c_i 2^{\gamma_1}}}{g^{w_i}} \bmod N$, $E_i = \frac{g^{v_i}}{A^{c_i}} \bmod N$, $F_i = g^{u_i - c_i 2^{\gamma_1}} \cdot h^{v_i} \cdot B^{c_i} \bmod N$ and $G_i = \frac{C_i^{u_i - c_i 2^{\gamma_1}} \cdot H_0(\mathsf{at}_i)^{c_i}}{Z_i^{w_i}} \bmod N$.
(3) Return 1 if $f(0) = H_1\left(m, \mathcal{P}, \ell, h, A, B, \{C_i, D_i, E_i, F_i, G_i, Z_i\}_{\mathsf{at}_i \in \mathcal{P}}\right)$, and return 0 otherwise.

This signature scheme is obtained through the transformation explained in section 2 of the previous Chapter, that is, the signature is a non-interactive zero-knowledge proof of knowledge of an integer $e$ and at least $\ell$ $e-th$ roots modulo $N$ of the values $H_0(at_i)$, for $i = 1, \ldots, n$.

# Chapter 5
# Non-membership based on Polynomial Evaluation Argument

The first model studied to implement revocation into the RSA attribute-based signature, was developed by Bayer and Groth in [**4**]. The revocation method is based on a polynomial evaluation argument, this construction is a public-coin 3-move special honest verifier zero-knowledge argument which only relies on the discrete logarithm assumption and has logarithmic communication cost in the size $D$ of the revocation list.

This model was chosen because it is an improvement from previous methods with communication complexity to $O(\sqrt{D})$ elements such as the ones developed by Brands, Demuynck and De Decker [**5**] and Peng [**24**]. Another reason for selecting this method instead of accumulators such as the one proposed by Li, Li and Xue in [**19**], which was previously studied in [**15**], is that accumulators rely on a trusted third party to maintain the accumulator, which could cause problems if this party was corrupted.

Moreover, in contrast to accumulator protocols such as [**19**] that work in groups of hidden order, where operations must be done in $\mathbb{Z}$ and the resulting elements are too big, Bayer and Groth protocol is implemented in groups with public prime order, hence the elements of the proofs although are bounded values, even though there can be more elements than in an accumulator protocol.

A complete description of the polynomial evaluation argument is exposed in the next section.

## 1. Polynomial evaluation argument

Bayer and Groth polynomial evaluation argument is explained for polynomials in $\mathbb{Z}_{\tilde{p}}[X]$, where $\tilde{p}$ is a public prime, but the attribute-based RSA signature is developed in $QR_N$, which has order $pq$ that is not public. Hence, a new execution of the $\mathsf{RSA.Inst}(1^\lambda)$ algorithm is needed, the commitment key is $ck = (\mathbb{G}, \tilde{p}, \tilde{g}, \tilde{h})$, where

$\mathbb{G}$ is a cyclic group of $k$-bit prime order $\tilde{p}$ and random generators $\tilde{g}, \tilde{h}$.

Given a degree $D$ polynomial $P(U) = \sum_{i=0}^{D} a_i U^i$ and two Pedersen commitments $c_0, c_v$ to values $u$ and $v$ in the group $\mathbb{G}$, an argument of knowledge for the values $u$ and $v$ that satisfy $P(u) = v$ will be described. It must be noted that the notation for $c_0$ for the commitment to $u = u^{2^0}$ is in accordance with the other commitments $c_j$ to $u^{2^j}$.
D is defined as $D = 2^{d+1} - 1$, also it is convenient to write $i$ in binary $i = i_d \cdots i_0$ where $i_j \in \{0, 1\}$. The term $U^i$ in the polynomial can be rewritten as $U^i = U^{\sum_{j=0}^{d} i_j 2^j} = \prod_{j=0}^{d} (U^{2^j})^{i_j}$. The polynomial obtained replacing this term is

$$P(U) = \sum_{i=0}^{D} a_i U^i = \sum_{i_0, \cdots, i_d = 0}^{1} a_{i_d \cdots i_0} \prod_{j=0}^{d} (U^{2^j})^{i_j}$$

The commitments $c_1, \cdots, c_d$ of the values $u^{2^1}, \cdots, u^{2^d}$ are inserted into the rewritten polynomial and the relation $P(u) = v$ becomes $\sum_{i_0, \cdots, i_d}^{1} a_{i_d \cdots i_0} \prod_{j=0}^{d} (u^{2^j})^{i_j} = v$. The prover only makes a logarithmic number of commitments seeing that $d = \lfloor \log D \rfloor$ which will be helpful in order to achieve efficiency.

A new polynomial is defined to show the committed powers of $u$ in $c_0, \cdots, c_d$ evaluate to the committed $v$, so the prover chooses random values $f_0, \cdots, f_d$ and the following polynomial is obtained.

$$Q(X) = \sum_{i_0, \cdots, i_d = 0}^{1} a_{i_d \cdots i_0} \prod_{j=0}^{d} (X u^{2^j} + f_j)^{i_j} X^{1 - i_j} = X^{d+1} P(U) + X^d \delta_d + \cdots + X \delta_1 + \delta_0$$

The choice of $Q(x)$ is based on that for each $i_j$ either an $X u^{2^j}$ factor is included or an $X$ factor is included so the coefficient of $X^{d+1}$ is $P(u)$. The contrary happens with $f_j$, each $f_j$ is not multiplied by $X$, hence it will only affect the lower degree coefficients $\delta_0, \cdots, \delta_d$ of $Q(x)$.

Afterwards the prover will demonstrate that the coefficient of $X^{d+1}$ in the secret $Q(X)$ is the same as $v$ in a sense that cancels out the $\delta_0, \cdots, \delta_d$ coefficients. The prover forwards the commitments $c_{f_0}, \cdots, c_{f_d}$, $f_0, \cdots, f_d$, $c_{\delta_0}, \cdots, c_{\delta_d}$, $\delta_0, \cdots, \delta_d$ to the verifier. Subsequently the verifier will choose a random challenge $x \leftarrow \mathbb{Z}_{\tilde{p}}$. The prover will open the corresponding products of the commitments so as to the verifier will be able to check that the committed values $u, v$ satisfy $Q(X) = x^{d+1} v + x^d \delta_d + \cdots + \delta_0$. Literally, after receiving the challenge $x$, the prover opens the products $c_j^x c_{f_j}$ to $\bar{f}_j = x u^{2^j} + f_j$. Moreover the prover opens $c_v^{x^{d+1}} \prod_{j=0}^{d} c_{\delta_j}^{x^j}$ to $\bar{\delta} = \sum_{i_0, \cdots, i_d = 0}^{1} a_{i_d \cdots i_0} \prod_{j=0}^{d} \bar{f}_j^{i_j} x^{1 - i_j}$. The verifier will only accept if the opening satisfies

$$\sum_{\bar{\delta} = i_0, \cdots, i_d = 0}^{1} a_{i_d \cdots i_0} \prod_{j=0}^{d} \bar{f}_j^{i_j} x^{1 - i_j} = x^{d+1} v + x^d \delta_d + \cdots + x \delta_1 + \delta_0$$

The previous identity has negligible probability of being true unless $P(u) = v$.

The prover will also need to show that $c_1, \cdots, c_d$ include the correct powers of $u$, $u^{2^1}, \cdots, u^{2^d}$. The prover sends commitments $c_{f_{u_j}}$ to $f_j u^{2^j}$ to the verifier and later opens the commitments $c_{u_{j+1}}^{x} c_{u_j}^{-\bar{f}_j} c_{f_{u_j}}$ to

$$xu^{2^{j+1}} - (xu^{2^j} + f_j)u^{2^j} + f_j u^{2^j} = 0$$

**1.1. Polynomial evaluation argument description.** The complete polynomial argument is mentioned below.

**Common reference string:** $ck = (\mathbb{G}, \tilde{p}, \tilde{g}, \tilde{h}) \leftarrow \mathsf{RSA.Inst}(1^\lambda)$

**Statement:** $P(U) = \sum_{i=0}^{D} a_i U^i = \sum_{i_0,\cdots,i_d}^{1} a_{i_d \cdots i_0} \prod_{j=0}^{d} (U^{2^j})^{i_j} \in \mathbb{Z}_{\tilde{p}}[U]$ and $c_0, c_v \in \mathbb{G}$

**Prover's witness:** $\mathsf{PK}\{(u, v, r_0, t) : c_0 = \tilde{g}^u \tilde{h}^{r_0} \wedge c_v = \tilde{g}^v \tilde{h}^t \wedge P(u) = v\}$

**Initial message:** Compute

(1) $c_1 = \tilde{g}^{u^{2^1}} \tilde{h}^{r_1}, \cdots, c_d = \tilde{g}^{2^d} \tilde{h}^{r_d}$ where $r_1, \cdots, r_d \leftarrow \mathbb{Z}_{\tilde{p}}$

(2) $c_{f_0} = \tilde{g}^{f_0} \tilde{h}^{s_0}, \cdots, c_{f_d} = \tilde{g}^{f_d} \tilde{h}^{s_d}$ where $f_0, s_0, \cdots, f_d, s_d \leftarrow \mathbb{Z}_{\tilde{p}}$

(3) $\delta_0, \cdots, \delta_d \in \mathbb{Z}_{\tilde{p}}$ such that

$$\sum_{i_0,\cdots,i_d=0}^{1} a_{i_d \cdots i_0} \prod_{j=0}^{d} (Xu^{2^j} + f_j)^{i_j} X^{1-i_j} = X^{d+1} v + \sum_{i=0}^{d} X_j \delta_j$$

(4) $c_{\delta_0} = \tilde{g}^{\delta_0} \tilde{h}^{t_0}, \cdots, c_{\delta_d} = \tilde{g}^{\delta_d} \tilde{h}^{t_d}$ where $t_0, \cdots, t_d \leftarrow \mathbb{Z}_{\tilde{p}}$

(5) $c_{f_{u_0}} = \tilde{g}^{f_0 u^{2^0}} \tilde{h}^{\xi_0}, \cdots, c_{f_{u_{d-1}}} = \tilde{g}^{f_{d-1} u^{2^{d-1}}} \tilde{h}^{\xi_{d-1}}$ where $\xi_0, \cdots, \xi_{d-1} \leftarrow \mathbb{Z}_{\tilde{p}}$

**Challenge:** $x \leftarrow \mathbb{Z}_{\tilde{p}}$

**Answer:** Compute for all $j$

$$\bar{f}_j = xu^{2^j} + f_j \qquad \bar{r}_j = xr_j + s_j \qquad \bar{t} = x^{d+1} t + \sum_{i=0}^{d} t_i x^i \qquad \bar{\xi}_j = xr_{j+1} - \bar{f}_j r_j + \xi_j$$

Send: $\bar{f}_0, \bar{r}_0, \cdots, \bar{f}_d, \bar{r}_d, \bar{t}, \bar{\xi}_0, \cdots, \bar{\xi}_{d-1}$

**Verification:** Accept if and only if for all $j$

$$c_j^x c_{f_j} = \tilde{g}^{\bar{f}_j} \tilde{h}^{\bar{r}_j} \qquad c_{j+1}^x c_j^{-\bar{f}_j} c_{f_{u_j}} = \tilde{h}^{\bar{\xi}_j}$$

and

$$c_v^{x^{d+1}} \prod_{i=0}^{d} c_{\delta_i}^{x^i} = \tilde{g}^{\sum_{i_0,\cdots,i_d=0}^{1} a_{i_d\cdots i_0} \prod_{j=0}^{d} \bar{f}_j^{i_j} x^{1-i_j}} \tilde{h}^{\bar{t}}$$

This polynomial evaluation argument satisfies the properties of a zero-knowledge proof. The complete proof can be found in [**4**]. This proof uses a similar definition to zero-knowledge proof called argument of knowledge that was first explained by Groth and Ishai [**17**], which uses the term witness-extended emulation described by Lindell [**20**].

Bayer and Groth [**4**] employ the term special honest verifier zero-knowledge $SHVZK$, which is not full zero-knowledge and in real life applications, sometimes special honest verifier zero-knowledge may not suffice since a malicious verifier could give non-random challenges. Nonetheless, it is always possible to convert an argument into a full zero-knowledge argument secure against arbitrary verifiers. This conversion only costs a small overhead.

**1.2. Efficiency.** An estimation of the communication cost for the polynomial evaluation argument was made which will be used later to assess the efficiency of the revocation method. The communication size for a degree $D = 2^{d+1}$ polynomial is approximately $4d$ elements from $\mathbb{G}$ and $3d$ elements from $\mathbb{Z}_{\tilde{p}}$ elements.

The prover employs $8d$ exponentiations in order to compute the commitments. The values $\delta_0, \cdots, \delta_d$ that satisfy

$$\sum_{i_0,\cdots,i_d=0}^{1} a_{i_d\cdots i_0} \prod_{j=0}^{d} (Xu^{2^j} + f_j)^{i_j} X^{1-i_j} = X^{d+1}v + X^d\delta_d + \cdots + X\delta_1 + \delta_0$$

also need to be calculated.

The $D$ polynomials $\prod_{j=0}^{d}(Xu^{2^j} + f_j)^{i_j} X^{1-i_j}$ with degree $d+1$, can be calculated using a binary-tree algorithm for all choices of $i_0, \cdots, i_d \in \{0,1\}$ with a cost of $dD$ multiplications in $\mathbb{Z}_{\tilde{p}}$. Multiplying with the $a_{i_d\cdots i_0}$ are another $dD$ multiplications. The total cost for the prover is $8d$ exponentiations in $\mathbb{G}$ and $2dD$ multiplications in $\mathbb{Z}_{\tilde{p}}$.

The verifier checks the argument using $6d$ exponentiations in $\mathbb{G}$ because the exponent $x$ is twice employed in the verification equations. In addition, this sum $\sum_{i_0,\cdots,i_d=0}^{1} a_{i_d\cdots i_0} \prod_{j=0}^{d} \bar{f}_j^{i_j} x^{1-i_j}$ for all choices $i_0, \cdots, i_d \in \{0,1\}$ must be calculated using $2D$ multiplications in $\mathbb{Z}_{\tilde{p}}$.

## 2. Non-membership Argument

Finally, a non-membership argument is constructed using the polynomial argument in the previous section.

Given a public revocation list $\mathcal{L} = \{e_1, \cdots, e_D\}$, where the values are chosen from $\Delta$ the set of integers in the interval $[2^{\gamma_1} - 2^{\gamma_2} + 1, 2^{\gamma_1} + 2^{\gamma_2} - 1]$ from the RSA attribute-based signature scheme of Chapter 4, the purpose is to demonstrate that the committed value $u \notin \mathcal{L}$. This value $u$ corresponds to the prime number $e$ given in the secret key of a user of the RSA attribute-based signature scheme. It must be noted that a commitment $B = g^e \cdot h^r \bmod N$ to $e$ is part of each attribute-based signature.

The following public polynomial is defined $P(X) = \prod_{i=1}^{D}(X - e_i)$ with the revoked elements set as roots. This polynomial is defined in $\mathbb{Z}_{\tilde{p}}[X]$, where $\tilde{p}$ is a prime number bigger than $2^{\gamma_1} + 2^{\gamma_2} + 1$ and a group $\mathbb{G}$ of order $\tilde{p}$ is chosen with generators $\tilde{g}, \tilde{h}$.

Given this polynomial, $u \in \mathcal{L}$ if and only if $P(u) = 0$. The prover has the commitment $c_u = \tilde{g}^u \tilde{h}^r \in \mathbb{G}$ and needs to demonstrate that the value $u$ is not in $\mathcal{L}$ showing that $P(u) \neq 0$.

In first place, since the RSA attribute-based signature scheme and the polynomial evaluation argument are developed in different groups, it is necessary that the prover proves to the verifier that the value committed in $B$ in the ABS,in bases $(g, h)$ is the same as the value committed in $c_u$ in bases $(\tilde{g}, \tilde{h})$: In second place, the prover computes $v = P(u)$ and calculates the commitment $c_v = \tilde{g}^v \tilde{h}^t$. Using the polynomial evaluation argument described in the previous section, the prover can prove that the commitment $c_v$ contains $v = P(u)$. Afterwards to prove non-membership the prover only needs to prove that $v \neq 0$. The majority of the cost of the revocation method lies in the polynomial evaluation argument.

A description of the non-membership steps is hereunder:

**Common reference string:** A key generation algorithm chooses a group $\mathbb{G}$ of order $\tilde{p}$ and random generators $\tilde{g}, \tilde{h}$. The commitment key is $ck = (\mathbb{G}, \tilde{p}, \tilde{g}, \tilde{h})$

**Statement:** $\mathcal{L} = \{e_1, \cdots, e_D\}, P(X) = \prod_{i=1}^{D}(X - e_i) \in \mathbb{Z}_{[\tilde{p}]}$ and $c_u \in \mathbb{G}$

**Prover's witness:**
*Step 1:* $\mathsf{PK}\{(u, s, r) : B = g^u h^s \wedge c_u = \tilde{g}^u \tilde{h}^r\}$

*Step 2:* $\mathsf{PK}\{(u, r) : c_u = \tilde{g}^u \tilde{h}^r \wedge u \notin \mathcal{L}\}$

**Argument:** *Step 1:* The prover proves to the verifier that the value committed in $B$ in bases $(g, h)$ is the same as the value committed in $c_u$ in bases $(\tilde{g}, \tilde{h})$:

$$\mathsf{PK}\{(u, s, r) : B = g^u h^s \wedge c_u = \tilde{g}^u \tilde{h}^r\}$$

(1) The prover chooses $y \in [0, 2^{(\gamma_1+\gamma_2+\kappa)}]$, $\rho_1, \rho_2 \in [0, 2^{(2\lambda+\kappa)}]$ at random and sends $Y_1 = g^y h^{\rho_1}$ and $Y_2 = \tilde{g}^y \tilde{h}^{\rho_2}$ to the verifier.

(2) The verifier chooses random $z \in [0, 2^{\kappa}]$ and sends it to the prover.

(3) The prover sends $x = y + zu$, $w_1 = \rho_1 + zs$ and $w_2 = \rho_2 + zr$.

(4) The verifier checks that $g^x h^{w_1} = Y_1 (B)^z$ and $\tilde{g}^x \tilde{h}^{w_2} = Y_2 (c_u)^z$.

*Step 2:* Pick $t_1, t_2 \leftarrow \mathbb{Z}_{\tilde{p}}$, $v = P(u)$, $w = v^{-1}$ are computed, the commitments $c_v = \tilde{g}^v \tilde{h}^{t_1}$, $c_w = \tilde{g}^w \tilde{h}^{t_2}$ are calculated. The prover proves to the verifier that $v \neq 0$.

$$\mathsf{PK}\{(v, w, t_1, t_2) : c_v = \tilde{g}^v \tilde{h}^{t_1} \wedge c_w = \tilde{g}^w \tilde{h}^{t_2}\}$$

(1) The prover chooses $\alpha, \beta \in [0, 2^{(\gamma_1+\gamma_2+\kappa)}]$, $\rho_1, \rho_2 \in [0, 2^{(2\lambda+\kappa)}]$ and $\rho_3 \in [0, 2^{(\gamma_1+\gamma_2+2\lambda+\kappa)}]$ at random and sends $Y_1 = \tilde{g}^\alpha \tilde{h}^{\rho_1}$, $Y_2 = \tilde{g}^\beta \tilde{h}^{\rho_2}$ and $Y_3 = (c_w)^\alpha \tilde{h}^{\rho_3}$ to the verifier.

(2) The verifier chooses random $z \in [0, 2^\kappa]$ and sends it to the prover.

(3) The prover sends $\psi_1 = \alpha + zv$, $\psi_2 = \beta + zw$ $\theta_1 = \rho_1 + zt_1$ and $\theta_2 = \rho_2 + zt_2$, $\theta_3 = \rho_3 - z\alpha t_2$ and sends $\psi_1, \psi_2, \theta_1, \theta_2, \theta_3$ to the verifier.

(4) The verifier checks that $\tilde{g}^{\psi_1} \tilde{h}^{\theta_1} = Y_1 (c_v)^z$, $\tilde{g}^{\psi_2} \tilde{h}^{\theta_2} = Y_2 (c_w)^z$ and $(c_w)^{\psi_1} \tilde{h}^{\theta_3} = Y_3 g^z$.

*Step 3:* The prover will need to prove in zero-knowledge that $P(u) = v$ using the polynomial evaluation argument previously described.

**Verification:** The verifier accepts $u \notin \mathcal{L}$ if and only if the value committed in $B$ in the ABS in bases $(g, h)$ is the same as the value committed in $c_u$ in bases $(\tilde{g}, \tilde{h})$ and the commitment $c_v \in \mathbb{G}$ and the zero-knowledge arguments are valid.

The first step is a zero-knowledge proof of knowledge of equality of representation in different bases. Completeness of this step is clear, and the zero-knowledge proof of knowledge can be seen below. It will be shown that even a malicious verifier $V'$, would not obtain new information on $u$ from the execution of the protocol.

$$\mathsf{PK}\{(u, s, r) : B = g^u h^s \wedge c_u = \tilde{g}^u \tilde{h}^r\}$$

A transcript $(Y_1, Y_2, z, x, w_1, w_2)$ of the protocol between the prover and $V'$ can be simulated as follows:

(1) Choose at random $x \in [0, 2^{(\gamma_1+\gamma_2+\kappa)}]$, $w_1, w_2 \in [0, 2^{(2\lambda+\kappa)}]$

(2) Choose at random $z \in [0, 2^\kappa]$ with the same distribution as $V'$ does and,

(3) compute $Y_1 = g^x h^{w_1} \cdot B^{-z}$ and $Y_2 = \tilde{g}^x \tilde{h}^{w_2} \cdot (c_u)^{-z}$

In order to show soundness of this step, a similar proof can be found in the following Chapter 6 and also there are more related proofs in [**14**] and [**15**].
The second and third steps above is a zero-knowledge proof of an opening of $c_u$ to $u \notin \mathcal{L}$. The protocol consists of two different zero-knowledge proofs. The third step stems from the polynomial evaluation argument and the second step is a classical zero-knowledge proof of knowledge of discrete logarithms in a group $\mathbb{G}$ with prime and public order $\tilde{p}$ which can be proved in a similar way as step 1. The combination of the two proofs is clear and the completeness of the protocol remains.

Finally, the zero-knowledge proofs from the two steps will prove that the protocol gets openings $u, v$ of the commitments satisfying $v \neq 0$ and $P(u) = v$, and that the

value $u$ from $c_u$ is the same one from $B$ in the ABS. This means that $u$ is not a root of the polynomial $P(X) = \prod_{i=1}^{D}(X - e_i)$ and thus $u \notin \mathcal{L}$.

# Chapter 6
# Peng and Bao non-membership proof for special applications

Subsequent to adding the revocation protocol based on the polynomial evaluation argument in Chapter 5 into the RSA attribute-based signature scheme of Chapter 4, the result was compared with a previous method incorporated in [**15**]. The method previously employed was originally described by Li,Li, Xue [**19**] and was based on accumulators.

One of the main differences between [**19**] and [**4**] is that in the case of [**19**] the revoked list $\mathcal{L} = \{e_1, \cdots, e_n\}$ is private, but in the case of Bayer and Groth [**4**] the revoked list is made public. Bayer and Groth did not considered that maintaining the revocation list private is essential, so the revoked users will lose their anonymity, and instead the necessity of a trusted third party on-line is removed, as there is no need of an entity that maintains the accumulator in a private way and issues non-membership witnesses for non-revoked users on top of updating privately the witnesses every time a new user is revoked.

Taking into consideration the previous remarks, the idea of employing an accumulator was recovered, additionally a new protocol described by Peng and Bao [**23**] was found, which seemed more efficient than the Li, Li, Xue [**19**] protocol used in [**15**] in the case where the revoked elements are loosely coprime (a property that will be defined later), which occurs in the attribute-based signature considered in this work [**18**], because the elements $e_i$ of each signature are prime numbers.

After making the adaptation of the protocol described by Peng and Bao, we tried to prove in detail that the protocol satisfied the required soundness and zero-knowledge properties. Regarding the soundness property, we were not able to succeed. This is a common situation in cryptographic research, when one tries to prove some security property of a specific protocol: if one does not succeed, it may happen that the security proof exists but is very tricky, or it may happen that the protocol is not secure, and so there is no security proof. In this particular case of the protocol in [**23**], we were actually able to find an explicit attack against the soundness property. After that, we modified their protocol in order to achieve soundness, as we will see in the next chapter.

In this Chapter, we first describe the protocol by the Peng and Bao [**23**] and then we explain the specific attack that we found, against the protocol.

## 1. Formulation of the protocol

In this protocol, there is a public list of integer values, $\mathcal{L} = \{s_1, s_2, \ldots, s_n\}$, and a prover has committed an integer element $m$ into a commitment $c_m$. This prover wants to prove, in zero-knowledge, that the committed value $m$ is not in the list $\mathcal{L}$. The protocol of Peng and Bao (in Section 3 of [**23**]) works in the special case where $m$ is loosely coprime with the elements in $\mathcal{L}$; that is, $\gcd(m, \prod_{i=1}^{n} s_i) < \min(|s_1|, |s_2|, \ldots, |s_n|)$.

The steps of the protocol are described below:

*Step 1:* The commitment algorithm corresponds to the one developed by Fujisaki-Okamoto [**16**]. The message $m$ is committed in $c = g^m h^r \bmod N$ where $r$ is randomly chosen from $\mathbb{Z}_N$.

*Step 2:* The prover calculates $C = g^{s_1 s_2 \cdots s_n} \bmod N$.

*Step 3:* The prover uses the Euclidean algorithm and the Euclidean algorithm from chapter 2 to calculate integers $k$ and $l$ in $\mathbb{Z}$ that satisfy

$$km + l \prod_{i=1}^{n} s_i = j$$

where $j = \gcd(m, \prod_{i=1}^{n} s_i)$.

*Step 4:* The prover proves the knowledge of secret integers $k, l, j, R$ such that

$$c_m^k C^l = g h^R \bmod N$$

$$j < s$$

as follows where $R = kr$ and $s = \min(|s_1|, |s_2|, \ldots, |s_n|)$.

(1) The prover publishes $f = g^j h^{r'} \bmod N$ where $r'$ is randomly chosen from $\mathbb{Z}_N$.

(2) The prover proves that the integer committed in $f$ is in $\{1, 2, \cdots, s-1\}$ using the range proof in [**7**].

(3) The prover proves that he knows secret integers $k, l, j, R, r'$ such that

$$c_m^k C^l = g h^R \bmod N$$

$$f = g^j h^{r'} \bmod N$$

Trough the following protocol:

(3.1) The prover randomly chooses integers $u, v$ and $w'$ in $\mathbb{Z}_\rho$. The following equations are computed:

$$y = um + v\prod_{i=1}^{n} s_i$$
$$w = ur$$
$$a = g^y h^{w'} \bmod N$$

where $\rho \gg k\tau$, $\rho \gg l\tau$, $\rho \gg j\tau$, $\rho \gg R_\tau$, $\rho \gg r'\tau$ where $\tau$ is a security parameter, and $a$ is send to the verifier.

(3.2) A random challenge $z$ in $\mathbb{Z}_\tau$ is generated by the verifier.

(3.3) The prover publishes

$$b_1 = u - zk \in \mathbb{Z}$$
$$b_2 = v - zl \in \mathbb{Z}$$
$$b_3 = y - zj \in \mathbb{Z}$$
$$b_4 = w - zR \in \mathbb{Z}$$
$$b_5 = w' - zr' \in \mathbb{Z}$$

(3.4) The verifier checks

$$c^{b_1} C^{b_2} = g^{b_3} h^{b_4} \bmod N$$
$$g^{b_3} h^{b_5} f^z = a \bmod N$$

The proof is accepted if the verification of the range proof (2) and the verification equations in (3.4) are satisfied.

## 2. Description of an attack

A dishonest prover that has been revoked, for example if his number is $s_1 = 3$, and his message $m$ is equal to 3, committed in the value $c_m = g^3 \cdot h^r$. This revoked user would be able fool the verifier with a probability of 1/3, through the following procedure.

Such a revokd user may define $j = 1$, $k = 1/3$ (a rational number!), $l = 0$, and then follow all the steps of the protocol, and trust that the challenge $z$ would be a multiple of 3, which happens with a probability of 1/3 in the interactive case, as the challenge is chosen $z$ is a random integer chosen by the verifier.

The case of non-interactive signatures is even worse, because the dishonest prover can pick random values until the output of the hash function $z = H(\cdots)$ is a multiple of 3.

In such a case, the dishonest prover is able to calculate the value $b_1 = u - zk$, which is an integer value. The same happens with the rest of values $b_2, b_3, b_4, b_5$ (note that $R = kr$ may be rational, as it happens with $k$, but $b_4$ will be an integer with probability 1/3). Summing up, with probability 1/3 the two verification equations are satisfied and the dishonest prover convinces the verifier, which results in a clear attack against the soundness property of the protocol.

# Chapter 7
# New proposal of non-membership proof

After finding that the Peng and Bao [**23**] revocation protocol was not sound, a number of modifications and additions were made in order to create a revocation protocol designed specially for the Attribute-Based Signature of [**18**], because in this case the elements of the revoked list $\mathcal{L} = \{e_1, \cdots, e_n\}$ are prime numbers as the correspond with the element $e_i$ of each signature.

As stated in the previous chapter, after incorporating the revocation protocol based on the polynomial evaluation argument in Chapter 5 into the RSA attribute-based signature scheme it was compared with the previous method incorporated in [**15**]. Since the other method was based on accumulators, the master entity or a trusted third party was needed to maintain the accumulator in a private way and issue non-membership witnesses for non-revoked users. Every time a user was revoked the master entity would have need to update the accumulator, and each user of the system would need to update his non-membership witness. Though the cost of the accumulator protocol is fixed as it does not depend on the number of revoked users in contrast to the polynomial evaluation protocol, the total cost of communication is higher in the case of the accumulator (unless the revocation list grows considerably).

However the main disadvantages of accumulator protocols that Bayer and Groth listed in [**4**], such as accumulators are dynamic, a trusted third party is needed and the constant update between master entity and users, do not apply to our particular attribute-based signature scheme, as the security of the signature would enable to make public the revoked list $\mathcal{L} = \{e_1, \cdots, e_n\}$. Hence a trusted third party would not be needed anymore as the accumulated value $C = g^{e_1 e_2 \cdots e_n} \bmod N$ can be calculated by the user, thus simplifying the revocation method: the non-revoked users can compute their witnesses for the list $\mathcal{L}$, without any dynamic interaction with the master entity, whose only task will be the update of $\mathcal{L}$.

In this Chapter a new revocation protocol will be proposed, based on the idea of accumulators but making the revocation list $\mathcal{L} = \{e_1, \cdots, e_n\}$ public as we considered as Bayer and Groth did in [**4**] that is not essential to maintain this list private, thus simplifying and reducing the cost of the protocol. The basics of the protocol are derived from Peng and Bao [**23**], but a number of modifications will be made to ensure soundness.

Finally in Chapter 8 a comparison between Bayer-Groth protocol and this new solution will be made in the case of revoking user in the RSA attribute-based signature of Chapter 4.

*Step 1:* The commitment algorithm corresponds to the one developed by Fujisaki-Okamoto [**16**]. The value $u$ is committed in $c_u = g^u h^r \bmod N$ where $r$ is randomly chosen from $\mathbb{Z}_N$. This value $u$ corresponds to the prime number $e$ given in the secret key of a user of the RSA attribute-based signature scheme. It must be noted that the commitment $c_u$ is actually equal to $B = g^e \cdot h^r \bmod N$, which is already part of each attribute-based signature.

*Step 2:* The prover calculates $C = g^{e_1 e_2 \cdots e_n} \bmod N$.

*Step 3:* The prover uses the Euclidean algorithm and the Bezout's identity from chapter 2 to calculate integers $k$ and $l$ in $\mathbb{Z}$ that satisfy

$$ku + l\prod_{i=1}^{n} e_i = 1$$

In our particular attribute based signature scheme the revoked values $\mathcal{L} = \{e_1, \cdots, e_n\}$ are prime numbers, hence the value $u$ will satisfy the Bezout's identity, in any other case $u$ will correspond to a revoked value.

*Step 4:* The prover proves the knowledge of secret integers $k, l, R$ such that

$$c_u^k C^l = g h^R \bmod N \tag{1}$$

where $R = kr$.

A protocol for such a zero-knowledge proof of knowledge can be found below:

*Step 4:* The prover proves to the verifier that he knows secret integers $k, l, R$ such that

$$g = c_u^k C^l h^{-R} \bmod N$$

(1) In order to achieve soundness the following values $r_k, r_l, r_R$ are added and the equations $B_k = g^k h^{r_k}$, $A_k = g^{r_k}$, $B_l = g^l h^{r_l}$, $A_l = g^{r_l}$, $B_R = g^R h^{r_R}$, $A_R = g^{r_R}$ must be defined. Then the prover chooses $\alpha, \beta \in [0, 2^{(\gamma_1+\gamma_2+\kappa)}]$, $\rho \in [0, 2^{(\gamma_1+\gamma_2+2\lambda+\kappa)}]$, $\beta_k, \beta_l, \beta_R \in [0, 2^{(2\lambda+\kappa)}]$ at random and computes $Y = c_u^\alpha C^\beta h^{-\rho}$, $E_k = g^{\beta_k}$, $F_k = g^\alpha h^{\beta_k}$, $E_l = g^{\beta_l}$, $F_l = g^\beta h^{\beta_l}$, $E_R = g^{\beta_R}$ and $F_R = g^\rho h^{\beta_R}$. Finally the prover send to the verifier:

$$(A_k, B_k, A_l, R_l, A_R, B_R, Y, E_k, F_k, E_l, F_l, E_R, F_R)$$

(2) The verifier chooses random $s \in [0, 2^\kappa]$ and sends it to the prover.

(3) The prover sends $x = \alpha + sk$, $y = \beta + sl$, $z = \rho + sR$, $v_k = \beta_k + sr_k$, $v_l = \beta_l + sr_l$ and $v_R = \beta_R + sr_R$.

(4) The verifier checks that:

(a) $c_u^x C^y h^{-z} = Y \cdot g^s$
(b) $g^{v_k} = E_k \cdot (A_k)^s$
(c) $g^x \cdot h^{v_k} = F_k \cdot (B_k)^s$
(d) $g^{v_l} = E_l \cdot (A_l)^s$
(e) $g^y \cdot h^{v_l} = F_l \cdot (B_l)^s$
(f) $g^{v_R} = E_R \cdot (A_R)^s$
(g) $g^z \cdot h^{v_R} = F_R \cdot (B_R)^s$

**Theorem** 4. *The previous protocol is a zero-knowledge proof of knowledge of integer values $(k, l, r)$ such that $g = c^k C^l h^{-R} \bmod N$ and thus is a non-membership proof of the value $u$.*

The security properties of the non-membership proof derived from the zero-knowledge proof of knowledge are proved below

Proof. Completeness of the non-membership proof is straightforward.
In order to prove the zero-knowledge of the protocol, it will be shown that even a malicious verifier $V'$, would not obtain new information on $u$ from the execution of the protocol.

$$\mathsf{PK}\{(k, l, R) : g = c_u^k C^l h^{-R} \bmod N\}$$

A transcript $(Y, A_k, B_k, A_l, B_l, A_R, B_R, E_k, F_k, E_l, F_l, E_s, x, y, z, v_k, v_l, v_R)$ of the protocol between the prover and $V'$ can be simulated as follows:

(1) Choose at random $x, y \in [0, 2^{(\gamma_1\gamma_2+\kappa)}]$, $z \in [0, 2^{(\gamma_1+\gamma_2+2\lambda+\kappa)}]$, $v_k, v_l, v_R \in [0, 2^{(2\lambda+\kappa)}]$, and $A_k, B_k, A_l, B_l, A_R, B_R \in QR_N$.
(2) Choose at random $s \in [0, 2^\kappa]$ with the same distribution as $V'$ does and,
(3) compute $Y = c_u^x C^y h^{-z} \cdot g^{-s}$, $E_k = g^{v_k} \cdot (A_k)^{-s}$, $F_k = g^x \cdot h^{v_k} \cdot (B_k)^{-s}$, $E_l = g^{v_l} \cdot (A_l)^{-s}$, $F_l = g^y \cdot h^{v_l} \cdot (B_l)^{-s}$, $E_R = g^{v_R} \cdot (A_R)^{-s}$, $F_R = g^z \cdot h^{v_R} \cdot (B_R)^{-s}$,

In order to show soundness it is enough to proof soundness for Step 4 of the protocol. To demonstrate soundness, we assume that some prover $P^*$ can execute the protocol with a non-negligible success probability. An algorithm that uses $P^*$ as a subroutine and extracts integer values $k, l, R$ satisfying equation 1 is shown below.
By assumption on $P^*$, using standard rewinding techniques, there is a situation, for a given $\mathsf{PK}$, $P^*$ could answer for two different challenges $s$ and $s'$ the values $(x, y, z, v_k, v_l, v_R)$ and $(x', y', z', v'_k, v'_l, v'_R)$. Dividing the equations, hereunder is the resulting one.

$$g^{s-s'} = (c_u)^{x-x'} C^{y-y'} h^{z-z'} \tag{2}$$

Furthermore, since the values $A_k, B_k, E_k, F_k, A_l, B_l, E_l, F_l, A_R, B_R, E_R, F_R$ are defined in the first step of Step 4, they will be the same in both executions of the

protocol. Dividing the corresponding verification equations (b),(d) and (f), we obtain:

$$g^{v_k - v'_k} = A_k^{s-s'} \qquad g^{v_l - v'_l} = A_l^{s-s'} \qquad g^{v_k - v'_k} = A_k^{s-s'}$$

Since all the equations follow the same pattern, it will only be proved for the case of $A_k$. Applying Lemma 2 from Chapter 2, under the strong RSA assumption it can be concluded, that $s - s'$ must divide $v_k - v'k$, meaning that $\delta_k = \frac{v_k - v'k}{s-s'}$ is an integer.

Afterwards the same will happen for the verification equations (c),(e),(g). Again we consider one of the cases, (the other two are analogous): dividing the two instances of equation (c) we obtain.

$$g^{x-x'} \cdot h^{v-k-v'-k} = B_k^{s-s'}$$

From the previous case, $v_k - v'_k$ can be replaced by $\delta_k(s - s')$ and below it can be seen the resulting equation.

$$g^{x-x'} = (\frac{B_k}{h^{\delta_k}})^{s-s'}$$

Applying again Lemma 2 from Chapter 2, it can be concluded that $s - s'$ divides $x - x'$, hence $\delta_x = \frac{x-x'}{s-s'}$ is an integer number.

The same steps will be carried for the cases of $l$ and $R$, and finally it will be inferred that $x - x'$ as well as $y - y'$ and $z - z'$ can be divided by $s - s'$, which means that there are integers $k = \delta_x = \frac{x-x'}{s-s'}$, $l = \frac{y-y'}{s-s'}$, $R = \frac{z-z'}{s-s'}$, such that $g = (c_u)^k C^l h^R$, which completes the soundness proof. □

Let us now show that the probability that a prover can pass its verification is negligible if $u \in \mathcal{L}$.

Since this new proof has the soundness property, and the same happens with the zero-knowledge part of the Attribute-based signature scheme, a successful signature will mean that the signer knows integers $k, l, R, u, r$ such that:

$$c_u^k C^l = g h^R \qquad c_u = g^u h^r \bmod N$$

From the two previous equations,

$$g^{ku} h^{kr} g^{l \prod_{i=1}^n e_i} = g h^R \bmod N$$

That is to say the prover knows integers $ku + l \prod_{i=1}^n e_i - 1$ and $kr - R$ such that

$$g^{ku + l \prod_{i=1}^n e_i - 1} h^{kr - R} = 1 \bmod N$$

with an overwhelmingly large probability. In order to make the proof easier, $a, b$ are defined as $a = ku + l\prod_{i=1}^{n} e_i - 1$ and $b = kr - R$. Therefore, three different cases are considered:

- $b = 1$: If the exponent of $h$ equals 1, the discrete logarithm problem assumption would be broken, thus arriving to a contradiction.
- $b > 1$, the value $d$ is defined as $d = \gcd(a, b)$. If $d > 1$, then the following equation will hold,
$$(g^{a'} h^{b'})^d = 1 \bmod N$$
and the Strong RSA problem would be solved, thus breaking the strong RSA assumption for $x = 1$, hence arriving to a contradiction.
- From the previous cases, it can be deduced that $d = 1$, the Bezout's identity is applied to $a, b$, there are $\alpha, \beta$ such that $a\alpha + b\beta = 1$, applying Lemma 1 from Chapter 2, the following equation is obtained
$$g = (h^{-\alpha} g^{\beta})^b$$
Therefore breaking the strong RSA assumption for $x = g$ as a consequence another contradiction is found.

As a result of the previous cases, the only option left is that $b = 0$, thus $a$ must be equal to 0 modulus the order of the exponent which is $pq$. If $a = ku + l\prod_{i=1}^{n} e_i - 1 = 0 \bmod pq$, and the value $u$ satisfies $u = e_I$ where $1 \leq I \leq n$, then

$$1 = ke_I + l\prod_{i=1}^{n} e_i = e_I(k + l\prod_{i=1}^{I-1} e_i \prod_{i=I+1}^{n} e_i) \bmod pq$$

Ergo two different numbers would be obtained, one being the inverse of the other, which would enable to find $pq$ with high probability, and from $pq$ and $N$, $N$ could be factorized, thus arriving to a contradiction, so $u \notin \mathcal{L}$ □

This new method requires fewer elements than the polynomial evaluation argument and the length of the proof is fixed, and there are no longer the inconveniences of accumulators protocols. This method seems at first sight more efficient than the previous one, which will be proved in the next chapter, where the two methods will be compared.

# Chapter 8
# Efficiency Analysis

Finally in this chapter, after explaining and adapting both revocation protocols in Chapters 5 and 7 respectively for the particular RSA attribute-based signature scheme explained in Chapter 4, an analysis of efficiency and comparison between both methods will be carried.

First of all, a description of how to exactly incorporate the revocation protocol into the signature scheme will be done for both protocols. Afterwards the corresponding efficiency analysis in terms of cost of communications will be performed for both revocation protocols. Our main goal was to design and choose the most efficient protocol to remove users that have misbehaved or that have lost their attributes. The major problem consists in maintaining privacy, a user who wants to prove that he is not in the revocation list, can not just show his serial number and its corresponding non-membership witness, because the user will be revealing his serial number. In both protocols, the user will prove in zero-knowledge that his serial number is not in the revocation list.

## 1. Incorporating Revocation based on the Polynomial Evaluation Argument into the Signature Scheme

With the aid of this protocol, the user will be able to prove that he is not in the revocation list without revealing his secret value $e$, the user will just need to prove that the result of the polynomial evaluation argument is different from 0, as he is not revoked. Hereunder a description of the additions to the original RSA attribute-based signature scheme can be observed.

**Setup:** In the RSA attribute-based signature, the master entity chooses $N = PQ$, $P = 2p+1$ and $Q = 2q+1$. The public parameters are $\mathsf{pms} = (\kappa, \gamma_1, \gamma_2, \epsilon, \Delta, \lambda, N, g, H_0, H_1, q', \tilde{\mathcal{P}})$ and the master secret key is $\mathsf{msk} = (P, Q)$. Additionally, since the polynomial evaluation argument is developed in a different group, the following values must be added to the public parameters: $\mathbb{G}$, a group of order $\tilde{p}$, where $\tilde{p}$ is a prime at least $\gamma_1 + \gamma_2 + 1$ bits, $\tilde{g}$ and $\tilde{h}$. Furthermore, the revocation list

$\mathcal{L} = \{e_1, \cdots, e_D\}$ will be required to be publicly available.

**KeyGen:** In our signature, we take as input an identity $\mathsf{id}$, a subset of attributes $S \subset \tilde{\mathcal{P}}$ satisfied by $\mathsf{id}$, the master secret key $\mathsf{msk}$ and the public parameters $\mathsf{pms}$. The master entity chooses at random a prime number $e$, and the global secret key $(\mathsf{sk}_{\mathsf{id},S} = (e, \{\mathsf{sk}_i\}_{\mathsf{at}_i \in S}))$ is computed. The value $e$, is the number that it is used to prove that the user is not revoked, and corresponds to the value $u$ in Chapter 5.

**Sign and Verify:** $(m, \mathcal{P}, \ell, \mathsf{sk}_{\mathsf{id},S}, \mathsf{pms})$.The signature consists of a zero-knowledge non-interactive proof of knowledge of an integer $e$ and at least $\ell$ $e$-th roots modulo $N$ of the values $H(\mathsf{at}_1), \ldots, H(\mathsf{at}_n)$. In addition, the prover proves to the verifier that $e$ is not in the revocation list. Following the steps described in Chapter 5, first the prover commits his value $e$ in $c_u = \tilde{g}^u \tilde{h}^r$, and then proves that it is the same as the value committed in the element $B$ of the signature, and in the second step proves that the polynomial is different from 0, hence the value is not in the revocation list $\mathcal{L}$.

**Revocation:** If we want to revoke a user, the master entity must add his value $u$ to the current revocation list $\mathcal{L} = \{e_1, \cdots, e_D\}$, the resulting revocation list will be $\mathcal{L} = \{e_1, \cdots, e_D, u\}$.

## 2. Incorporating Revocation based on the protocol for special applications into the Signature Scheme

Incorporating revocation with this new protocol will be easier than the previous one since the proof is in the same group as the RSA attribute-based signature and less elements need to be added. This protocol is based on the concept that as all the revoked values are prime numbers, we will only need to proof in zero-knowledge that the secret value of the user $e$ satisfies the Bezout's identity with the product of the revoked numbers.

**Setup:** In the RSA attribute-based signature, the master entity chooses $N = PQ$, $P = 2p+1$ and $Q = 2q+1$. The public parameters are $\mathsf{pms} = (\kappa, \gamma_1, \gamma_2, \epsilon, \Delta, \lambda, N, g, H_0, H_1, q', \tilde{\mathcal{P}})$ and the master secret key is $\mathsf{msk} = (P, Q)$. In this case, only the revocation list $\mathcal{L} = \{e_1, \cdots, e_D\}$ must be added to the public parameters.

**KeyGen:** In our signature, we take as input an identity $\mathsf{id}$, a subset of attributes $S \subset \tilde{\mathcal{P}}$ satisfied by $\mathsf{id}$, the master secret key $\mathsf{msk}$ and the public parameters $\mathsf{pms}$. The master entity chooses at random a prime number $e$, and the global secret key $(\mathsf{sk}_{\mathsf{id},S} = (e, \{\mathsf{sk}_i\}_{\mathsf{at}_i \in S}))$ is computed. The value $e$, is the number that it is used to prove that the user is not revoked, also corresponds to the value designated as $u$ in Chapter 6.

**Sign and Verify:** $(m, \mathcal{P}, \ell, \mathsf{sk}_{\mathsf{id},S}, \mathsf{pms})$.The signature consists of a zero-knowledge non-interactive proof of knowledge of an integer $e$ and at least $\ell$ $e$-th roots modulo $N$ of the values $H(\mathsf{at}_1), \ldots, H(\mathsf{at}_n)$. In addition, the prover proves to the verifier that $e$ is not in the revocation list following the steps described in Chapter 6. Since

all the proof work in the same group $QR_N$, the commitment named $c_u$ in Chapter 6 corresponds to the commitment $B$ from the RSA attribute-based signature, and the proof can be done with this value.

**Revocation:** If we want to revoke a user, the master entity must add his value $u$ to the current revocation list $\mathcal{L} = \{e_1, \cdots, e_D\}$, the resulting revocation list will be $\mathcal{L} = \{e_1, \cdots, e_D, u\}$.

# 3. Efficiency

At last, it is essential to assess the efficiency of both resulting signatures schemes. In order to do that, with the choice of the parameters given in [**18**], the growth in the length of the signature in either cases will be determined. For a security level of $\lambda = 1024$, the parameters used are $\gamma_1 = 1080$, $\gamma_2 = 800$ and $\kappa = 160$. It must be noted that, although in the previous Chapters 5 and 7, the proofs in zero-knowledge were interactive, but when executing the protocol now, as part of the attribute-based signature scheme, a non-interactive method must be enlisted, thus a hash function will be used instead. Described below are the resulting non-interactive protocols of step 1 from Chapter 5, and step 4 from Chapter 7. Subsequently the communication cost of each revocation protocol will be assessed.

**3.1. Protocol 1: Polynomial evaluation argument.** Firstly, it is explained how the non-interactive version of step 1 and step 2 works.

*Step 1:* The prover proves to the verifier that the value committed in $B$ in bases $(g, h)$ is the same as the value committed in $c_u$ in bases $(\tilde{g}, \tilde{h})$:

$$\mathsf{PK}\{(u, s, r) : B = g^u h^s \wedge c_u = \tilde{g}^u \tilde{h}^r\}$$

(1) The prover chooses $y \in [0, 2^{(\gamma_1+\gamma_2+\kappa)}]$, $\rho_1, \rho_2 \in [0, 2^{(2\lambda+\kappa)}]$ at random and computes $Y_1 = g^y h^{\rho_1}$ and $Y_2 = \tilde{g}^y \tilde{h}^{\rho_2}$.

(2) The prover computes $z = H(Y_1, Y_2, B, c_u)$ where $z \in [0, 2^\kappa]$.

(3) The prover computes $x = y + zu$, $w_1 = \rho_1 + zs$ and $w_2 = \rho_2 + zr$ and sends $(z, x, w_1, w_2)$.

(4) The verifier checks whether:

$$z = H(g^x h^{w_1} B^{-z}, \tilde{g}^x \tilde{h}^{w_2} (c_u)^{-z}, B, c_u)$$

*Step 2:* Pick $t_1, t_2 \leftarrow \mathbb{Z}_{\tilde{p}}$, $v = P(u)$, $w = v^{-1}$ are computed, the commitments $c_v = \tilde{g}^v \tilde{h}^{t_1}$, $c_w = \tilde{g}^w \tilde{h}^{t_2}$ are calculated. The prover proves to the verifier that $v \neq 0$.

$$\mathsf{PK}\{(v, w, t_1, t_2) : c_v = \tilde{g}^v \tilde{h}^{t_1} \wedge c_w = \tilde{g}^w \tilde{h}^{t_2}\}$$

(1) The prover chooses $\alpha, \beta \in [0, 2^{(\gamma_1+\gamma_2+\kappa)}]$, $\rho_1, \rho_2 \in [0, 2^{(2\lambda+\kappa)}]$ and $\rho_3 \in [0, 2^{(\gamma_1+\gamma_2+2\lambda+\kappa)}]$ at random and computes $Y_1 = \tilde{g}^{\alpha}\tilde{h}^{\rho_1}$, $Y_2 = \tilde{g}^{\beta}\tilde{h}^{\rho_2}$ and $Y_3 = (c_w)^{\alpha}\tilde{h}^{\rho_3}$.

(2) The prover computes $z = H(Y_1, Y_2, Y_3, c_v, c_w)$ where $z \in [0, 2^{\kappa}]$.

(3) The prover computes $\psi_1 = \alpha + zv$, $\psi_2 = \beta + zw$, $\theta_1 = \rho_1 + zt_1$, $\theta_2 = \rho_2 + zt_2$, $\theta_3 = \rho_3 - z\alpha t_2$ and sends $(z, \psi_1, \psi_2, \theta_1, \theta_2, \theta_3)$ to the verifier.

(4) The verifier checks that:

$$z = H(\tilde{g}^{\psi_1}\tilde{h}^{\theta_1}(c_v)^{-z}, \tilde{g}^{\psi_2}\tilde{h}^{\theta_2}(c_w)^{-z}, (c_w)^{\psi_1}\tilde{h}^{\theta_3}g^{-z}, c_v, c_w)$$

The communication cost of step 1 is 6616 bits and four additional exponentiations will be needed for the prover to perform this step. In the case of step 2, the cost of communication is 12744 bits and five exponentiations are required in this step. These two steps determine the fixed cost of the protocol, as step 3 of the protocol (polynomial evaluation argument) depends on the number of revoked values. This fixed cost of communication sums 19360 bits and 9 exponentiations are needed.

Furthermore, the communication size of the polynomial evaluation argument for a degree $D = 2^{d+1}$ polynomial is approximately $4d$ elements from $\mathbb{G}$ and $3d$ elements from $\mathbb{Z}_{\tilde{p}}$ elements. The cost of representing in bits an element of $\mathbb{G}$ is the same one as an element of $\mathbb{Z}_{\tilde{p}}$. Since $\tilde{p}$ has to be greater than $2^{\gamma_1} + 2^{\gamma_2} + 1$, hence $\log \tilde{p}$, which is the length in bits, will be at least $\gamma_1 + \gamma_2 + 1 = 1881$ bits. The total cost of step 3 is determined by the formula $4 \log D\mathbb{G} + 3 \log D\mathbb{Z}_{\tilde{p}} = 13167 \log D$.

**3.2. Protocol 2: Special applications.** First of all, a non-interactive version of step 4 of the protocol can be seen below.

*Step 4:* The prover proves to the verifier that he knows secret integers $k, l, R$ such that

$$g = c_u^k C^l h^{-R} \bmod N$$

(1) In order to achieve soundness the following values $r_k, r_l, r_R$ are added and the equations $B_k = g^k h^{r_k}$, $A_k = g^{r_k}$, $B_l = g^l h^{r_l}$, $A_l = g^{r_l}$, $B_R = g^R h^{r_R}$, $A_R = g^{r_R}$ must be defined. Then the prover chooses $\alpha, \beta \in [0, 2^{(\gamma_1+\gamma_2+\kappa)}]$, $\rho \in [0, 2^{(\gamma_1+\gamma_2+2\lambda+\kappa)}]$, $\beta_k, \beta_l, \beta_R \in [0, 2^{(2\lambda+\kappa)}]$ at random and computes $Y = c_u^{\alpha} C^{\beta} h^{-\rho}$, $E_k = g^{\beta_k}$, $F_k = g^{\alpha} h^{\beta_k}$, $E_l = g^{\beta_l}$, $F_l = g^{\beta} h^{\beta_l}$, $E_R = g^{\beta_R}$ and $F_R = g^{\rho} h^{\beta_R}$.

(2) The prover computes

$$s = H(Y, E_k, F_k, E_l, F_l, E_R, F_R, A_k, B_k, A_l, B_l, A_R, B_R)$$

where $s \in [0, 2^{\kappa}]$.

(3) The prover computes $x = \alpha + sk$, $y = \beta + sl$, $z = \rho + sR$, $v_k = \beta_k + sr_k$, $v_l = \beta_l + sr_l$ and $v_R = \beta_R + sr_R$ and sends

$$(s, x, y, z, v_k, v_l, v_R, A_k, B_k, A_l, B_l, A_R, B_R)$$

(4) The verifier checks that

$$s = H(c_u^x C^y h^{-z} g^{-s}, g^{v_k} \cdot (A_k)^{-s}, g^x \cdot h^{v_k} \cdot (B_k)^{-s}, g^{v_l} \cdot (A_l)^{-s}, \\ g^y \cdot h^{v_l} \cdot (B_l)^{-s}, g^{v_R} \cdot (A_R)^{-s}, g^z \cdot h^{v_R} \cdot (B_R)^{-s})$$

The communication cost of this particular step is 21096 bits and 9 additional exponentiations will be needed for the prover to perform this step. Since the other steps do not entail any additional cost of communication, the total cost of communication of this second protocol is 21096 bits and the number of exponentiations only increases to 10, because we can use the commitment $B = g^e h^r$ from the original ABS as the value $c_u$.

**3.3. Comparison between both revocation protocols.** After determining the cost of communication for both protocols, hereunder in Table 1 can be seen the results. Moreover, it must be noted that protocol 1 depends on the size of the revocation list, contrary to protocol 2.

| Protocol 1 | Protocol 2 |
|---|---|
| $19360 + 13167 \log D$ bits | 21096 bits |

TABLE 1. Cost of revocation.

From the table above 1 it is clear that protocol 2 is more efficient than protocol 1, as the fixed cost of communication from protocol 1 is almost the whole cost of revocation of protocol 2. Already with a single revoked user, $D = 1$, the communication cost of Protocol 1 would be 32.527 bits, bigger than that of Protocol 2.

# 4. Analysis of efficiency of Protocol 2

As seen in the previous section, the most efficient revocation method was protocol 2. Right away, this protocol will be assessed against the original RSA attribute-based signature scheme. The original attribute-based signature scheme for a $(\ell, n)$-threshold signing policy with $n = |\mathcal{P}|$ attributes has an approximate length of $6800n + 3200 - 160\ell$ bits. The cost of each execution of the signing protocol is dominated by $10n$ exponentiations modulo $N$. Since the length of the signature depends on the number of attributes and the revocation of users is independent of it, implementing revocation of users might not be efficient for a small number of attributes, but it could be efficient for larger signing policies, with a larger number of attributes. In Table 2 below, the length of the attribute-based signatures is included, with and without the revocation extension, for different values of the

number $n$ of attributes of the signing policy, as well as the percentage increase of length of the signatures. The results in Table 2 give an idea about how much does it cost, at least in terms of communication cost, to add revocation to this particular attribute-based signature.

| $n$ | Approx. Kbit of ABS | Protocol 2 + ABS | Increasing % |
|---|---|---|---|
| 5 | 34000 | 55096 | 62% |
| 10 | 68000 | 89096 | 31% |
| 15 | 102000 | 123096 | 20.7% |
| 20 | 136000 | 157096 | 15.5% |
| 30 | 204000 | 225096 | 10.3% |
| 50 | 340000 | 361096 | 6.2% |
| 75 | 510000 | 531096 | 4.1% |
| 100 | 680000 | 701096 | 3.1% |

TABLE 2. Length of each ABS [**18**] with and without revocation for Protocol 2.

Comparing this results with the results obtained in [**15**], where other protocol originally developed by Li et al. [**19**] was considered to achieve a non-membership proof, the cost of communication has been reduced by 58% as it can be seen in Table 3, with an additional analysis of the reduction of the length of the resulting attribute-based signature scheme.
Furthermore, in the case of [**15**] the revoked list was private on the contrary to Protocol 2, this fact entails the necessity of a trusted third party online that would update secretly every user, every time there is a new update of the revoked list $\mathcal{L}$, which is eliminated in the case of Protocol 2.

| Cost of Procotol 1 | Cost of Protocol 2 | Cost of [**15**] Protocol | Best reduction % |
|---|---|---|---|
| $19360+13167 \log D$ | 21096 bits | 50000 bits | 58% |

TABLE 3. Comparing the communication cost of different revocation protocols

As it can be seen in Table 4, the effect of the new protocol is considerable in small and medium size policies.

| $n$ | ABS + Protocol 2 | ABS + [**15**] | Total ABS Reduction% |
|---|---|---|---|
| 5 | 55096 | 84000 | 34.4% |
| 10 | 89096 | 118000 | 24.5% |
| 15 | 123096 | 152000 | 19% |
| 20 | 157096 | 186000 | 15.5% |
| 30 | 225096 | 254000 | 11.4% |
| 50 | 361096 | 390000 | 7.4% |
| 75 | 531096 | 560000 | 5.5% |
| 100 | 701096 | 730000 | 4% |

TABLE 4. Comparing the length of attribute-based signatures with revocation.

# Chapter 9
# Conclusions

In this project, the main goal was to assess different revocation methods and find the most suitable for incorporating revocation of users into an existing RSA attribute-based signature scheme.

Having already developed a revocation method in [**15**], the concepts of the attribute-based signature and a number of mathematical concepts and assumptions which are essential in attribute-based cryptography were already known, but the previous revocation method relied on the concept of accumulators, which might sustain a number of disadvantages. For example, the necessity of a master entity that updates the accumulator and provides the user with a non-membership witness, which had to be updated every time a new member was revoked.

Hence, so as to find a more appropriate revocation protocol, a substantial number of papers were read and analyzed in order to find the most efficient ones, which are the two methods fully described in this project. The main mathematical objects, cryptographic assumptions and concepts used in this project are described in Chapter 2, Chapter 3 and Chapter 4.

Since we did not want to rely on accumulators and a trusted third party, the method developed in Chapter 5 was technically the most efficient one that could be found in the literature, as it has a communication complexity of $\mathcal{O}(\log D)$, where $D$ is the size of the revocation list; other methods such as the ones developed by Brands, Demuynck and De Decker [**5**], Peng [**24**] developed non-membership proofs with communication complexity to $\mathcal{O}(\sqrt{D})$ group elements. This method consisted of a polynomial evaluation protocol developed by Bayer and Groth [**4**], which was adapted to satisfy the needs of the attribute-based signature scheme.

After incorporating the previous revocation method, it was compared with the one developed in [**15**], and it was noted that in our particular attribute-based signature scheme the security of the signature would enable us to make public the revoked list $\mathcal{L} = \{e_1, \cdots, e_n\}$. Thus a trusted third party would not be required anymore as the accumulated value $\mathcal{C} = g^{e_1 e_2 \cdots e_n} \bmod N$ can be calculated by the user, facilitating the revocation method, and eliminating the need of a master entity that updates the accumulator or the witnesses . A new protocol described by Peng and Bao [**23**] was found that works in the case where the revoked elements satisfied the loosened

coprime condition, in our case, all the revoked elements are prime numbers, so a number of modifications of the protocol where made to adapt it to our attribute based signature scheme. Later on, it was found that the Peng and Bao protocol [**23**] was not sound and the protocol was not secure, even more an explicit attack against the soundness property was described in Chapter 6. In Chapter 7, the Peng and Bao protocol was modified in order to be secure, and also additional modifications were build to simplify the method because all the revoked elements are prime numbers in our attribute based-signature scheme, thus resulting in an even more efficient revocation scheme. Summing up, a more efficient and simple revocation mechanism was found that can be applied to our attribute-based signature scheme.

Finally, once both protocols were developed and explained, the efficiency of both these protocols was analyzed, in particular for the case of the communication cost, or in other words the length of the resulting complete attribute-based signatures. From the table at the end of Chapter 8, it can be observed that the second protocol, the one developed in Chapter 7 is more efficient, as the cost of communication is lesser than the cost of the polynomial evaluation protocol even in the case of only one revoked user. It must be noted that the revocation scheme of Chapter 7 can be applied to our particular attribute-based signature, but in other signature schemes (for instance, those based on pairings in groups of public and prime order) the situation could be different, and other non-membership protocols should be considered. An analysis of the computational complexity, that is, the time required to verify a signature, could be done essentially in the same way, and the conclusions would be almost identical.

At last, I would like to say that this has been an excellent learning experience and I am satisfied with the results from studying different revocation protocols and finding and adapting the most efficient one for the necessities of the existing attribute-based signature scheme. Although one of our initial goals was to implement the revocation protocol of Chapter 5, we were able to find a more efficient and simple one, even with the additional work that supposed the security problem of the original protocol developed by Peng and Bao [**23**] and the necessity to design a new secure protocol. As a possibility for future work, it would be very interesting to implement and test the resulting attribute-based signature scheme, including the revocation features provided in this work.

# References


[1] G. Ateniese, J. Camenisch, M. Joye, and G. Tsudik. A practical and provably secure coalition-resistant group signature scheme. *Advances in Cryptology - CRYPTO '00*, pg. 255-270 (2000)

[2] G. Ateniese, D. Song and G. Tsudik. Quasi-efficient revocation of group signatures. *Proceedings of Financial Cryptography*, pg. 183-197 (2001)

[3] N. Barić and B. Pfitzmann. Collision-free accumulators and fail-stop signature schemes without trees. *Advances in Cryptology - EUROCRYPTO '97*, pg. 480-494 (1997)

[4] S. Bayer and J. Groth. Zero-knowledge argument for polynomial evaluation with application to blacklists. *Advances in Cryptology - EUROCRYPT 2013 , LNCS 7881*, pg. 646-663 (2013)

[5] S. Brands, L. Demuynck, and B. De Decker. A practical system for globally revoking the unlinkable pseudonyms of unknown users. *ACISP, LNCS vol. 4586*, pg. 400-415 (2007)

[6] E. Bresson and J. Stern. Efficient revocation in group signatures. *Public Key Cryptography, LNCS vol. 1992*, pg. 190-206 (2001)

[7] F.Boudot. Efficient proofs that a committed number lies in an interval. *Advances in Cryptology - EUROCRYPT '00*, pg. 431-444 (2000)

[8] J. Camenisch and M. Michels. Proving in zero-knowledge that a number is the product of two safe primes. *Advances in Cryptology - EUROCRYPT '99*, pg. 106-121 (1999)

[9] J. Camenisch and M. Michels. Separability and efficiency for generic group signature schemes. *Advances in Cryptology - CRYPTO '99*, pg. 413-430 (1999)

[10] J. Camenisch and A. Lysynskaya. Efficient Revocation of Anonymous Group Membership Certificates and Anonymous Credentials (2001)

[11] J. Camenisch and A. Lysyanskaya. A signature scheme with efficient protocols. *Proceedings of the 3rd Conference on Security in Communication Networks*, pg. 268-289 (2002)

[12] J. Camenisch and A. Lysyanskaya. Dynamic accumulators and application to efficient revocation of anonymous credentials. *Advances in Cryptology - CRYPTO '02*, pg. 61-76 (2002)

[13] D. Chaum and T. Pedersen. Wallet databases with observers. *CRYPTO, LNCS vol. 740*, pg. 89-105 (1992)

[14] I. Damgård and E. Fujisaki. A statistically-hiding integer commitment scheme based on groups with hidden order. *Advances in Cryptology - ASIACRYPT '02*, pg. 125-142 (2002)

[15] M. Fueyo. Revocation of Users in RSA Attribute-Based Signatures (2014)

[16] E. Fujisaki and T. Okamoto. Statistical zero knowledge protocols to prove modular polynomial relations. *Advances in Cryptology - CRYPTO '97*, pg. 16-30 (1997)

[17] J. Groth and Y. Ishai. Sub-linear zero-knowledge argument for correctness of a shuffle. *Advances in Cryptology - EUROCRYPT 2008, LNCS 4965*, pg. 379-396 (2008)

[18] J. Herranz. Attribute-Based Signatures from RSA. *Theoretical Computer Science, Volume 527*, pag. 73-82 (2014)

[19] J. Li, N. Li and R. Xue. Universal accumulators with efficient nonmberbership proofs. *Proc. of ACNS'07, LNCS 4521*, Springer-Verlag, pg. 253-269 (2007)

[20] Y. Lindell. Parallel coin-tossing and constant-round secure two-party computation. *J. Cryptology 16(3)*, pg. 143-184 (2003)

[21] H.K. Maji, M. Prabhakaran and M. Rosulek. Attribute-based signatures. *Proc. of CT-RSA'11, LNCS 6558*, Springer-Verlag, pg. 376-392 (2011)

[22] T.B. Pedersen. Non-interactive and information-theoretic secure verifiable secret sharing. *Advances in Cryptology - CRYPTO'91*, pg. 129-140 (1991)

[23] K.Peng and F. Bao. Improving applicability, efficiency and security of non-membership proof. *International Symposium on Data, Privacy and E-Commerce*, pg. 39-44 (2010)
[24] K. Peng. A general, flexible and efficient proof of inclusion and exclusion. *CT-RSA, LNCS vol. 6558*, pg. 33-48 (2011)
[25] C. Schnorr. Efficient generation by smart cards. *Journal of Cryptology,4*, pg. 161-174 (1991)
[26] K. Yu, T. Yuen, S. Chow, S. Yiu and L. Hui. Pe(ar)2: Privacy-enhanced anonymous authentication with reputation and revocation. *ESORICS, LNCS vol. 7459*, pg. 679-696 (2012)