

# Anonymous yet Accountable Access Control\*

Michael Backes  
IBM Zurich Research  
Laboratory<sup>†</sup>

mbc@zurich.ibm.com

Jan Camenisch  
IBM Zurich Research  
Laboratory<sup>†</sup>

jca@zurich.ibm.com

Dieter Sommer  
IBM Zurich Research  
Laboratory<sup>†</sup>

dso@zurich.ibm.com

## ABSTRACT

This paper introduces a novel approach for augmenting attribute-based access control systems in a way that allows them to offer fully anonymous access to resources while at the same time achieving strong accountability guarantees. We assume that users hold attribute certificates and we show how to exploit cryptographic zero-knowledge proofs to allow requesting users to prove that they hold suitable certificates for accessing a resource. In contrast to the commonly taken approach of sending all possibly relevant certificates to the access control system, our approach hence does not release any information to the access control system except for the presence of a set of certificates satisfying the access condition. This constitutes the minimal amount of information that has to be released for coming up with a correct access decision, and our approach is the first to achieve this. Additionally given a trusted third party for identity escrow, we furthermore show that a concise application of zero-knowledge proofs offers the access control system the capability to hold a requesting user accountable for her actions under specific, well-defined conditions. All the employed cryptographic techniques are highly efficient, and an architecture for exploiting our approach in practical scenarios is already in place.

## Categories and Subject Descriptors

H.4 [Information Systems Applications]: Communication Applications—*privacy*

## General Terms

Security

<sup>†</sup>Säumerstrasse 4, CH-8803 Rüschlikon, Switzerland

\*Part of the work reported in this paper is supported by the European Commission through the IST Programme under Contract IST-2002-507932 ECRYPT and by the IST Project PRIME. The PRIME project receives research funding from the European Community's Sixth Framework Programme and the Swiss Federal Office for Education and Science.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

WPES'05, November 7, 2005, Alexandria, Virginia, USA.  
Copyright 2005 ACM 1-59593-228-3/05/0011 ...\$5.00.

## Keywords

Privacy, Accountability, Access Control, Anonymous Credentials, Certificates, Anonymous Transactions

## 1. INTRODUCTION

Today's interactions over open communication networks typically require users to provide information such as their identities in order to get access to services. Access control systems can then be used to grant or deny access to resources based on the obtained information after a user has registered for a service. In terms of user privacy, this approach however suffers from significant drawbacks. Firstly, multiple interactions of the same users are linkable for the access control system. Given that information processing systems are becoming increasingly powerful, the released data can be used by service providers to build extensive profiles of users, often without the users being aware of this. Secondly, it is usually very difficult to guarantee that the information released to the service provider is not leaked to additional parties or used in some other form that the user would consider a violation of her privacy.

Research in access control has made substantial progress with regard to access control systems for open communication networks, and recent work has started addressing the aforementioned problems [2, 24, 30, 31, 32]. An important step in this direction is to base access decisions on attributes of the requesting user rather than solely on her identity. This attribute-based access control approach enjoys several advantages over pure identity-based access control, both since it offers a greater flexibility for a user in terms of which information she wants to release, and since unique identities typically have no or only a restricted common meaning to service providers in open networks, whereas attributes can often be interpreted in a uniform manner by different service providers. The common way to provide attributes to service providers in a trustworthy manner is to send attribute certificates for the required attributes. Collecting these certificates then in particular allows the service provider to hold the user accountable for her actions. However, despite the greater deal of flexibility over identity-based access control data, linkability between multiple transactions involving the same certificates can still be established. For instance using X.509-like attribute certificates [20] immediately leads to a release of the requesting user's identity as the user's public key certificate [23] has to be released as well.

The optimal solution to protect the privacy of the user would be to come up with a mechanism that only releases the minimal information that is necessary for coming up with a correct access decision while additionally offering unlinkability of different transactions. However, to prevent users from staying anonymous even if they use the accessed resource in a malicious way, it seems appropriate to complement this with a mechanism for holding users

accountable under specific, well-defined conditions.

The approach presented in this paper achieves such a solution. On the one hand, it offers strong user privacy in the sense of unlinkable transactions and of guaranteed minimal release of information; on the other hand, it achieves strong accountability guarantees for the access control system. So far, these two properties have always been traded off for each other. Essentially, our work is based on the idea that it is sufficient for a requesting user to provide a proof that she possesses a set of certificates that satisfy the access condition, and that such a proof can be conducted in zero-knowledge, i.e., the service provider should only learn that the user should be granted access, but no information beyond that. In particular, there is no need for releasing any other information, e.g., which certificates are present, or the detailed values that are certified, and hence two proofs involving the same certificates do not necessarily become linkable. In more detail, our approach allows a user to prove in zero-knowledge that she holds a set of certificates that satisfy an access condition expressed as a logical formula in propositional logics without negation and with predicates, where the predicates can consist of arithmetic expressions over the attributes of her certificates. Common use cases comprise showing that a user is at least 18 years old, that her income exceeds a specific threshold, that multiple certificates have been issued to the same party, or any logical formula over such statements. We realize the proofs by proving knowledge of a signature (used to compute the certificates) in zero-knowledge using a signature scheme with suitable verification protocol, e.g., [7, 8]. The approach in particular requires a user to own certificates that are bound to her such that her private key is required whenever she performs a proof over her certificates.

While relying on zero-knowledge protocols ensures minimal release of information and hence offers strong privacy for the user, no user accountability is achievable using only such an approach as there is no way of identifying the user in those cases where she should be held accountable. Thus there must additionally be a mechanism to escrow the identity of the user. A suitable way for doing this is to rely on a trusted third party for identity escrow and to require in the access control rules that the identity be escrowed under this third party key and specific conditions, e.g., if the user used the accessed resource in a law-violating way. Ideally, the third party is not involved in the escrow process, but it is passive in that it is only involved if a decryption of an identity is requested. We achieve this by allowing an access control rule to require the user to provide encryptions of attributes of certificates under the public key of the trusted third party together with a proof that the encryptions actually are encryptions over certified attributes. The agreed condition under which the identity should be escrowed has to be cryptographically bound to the encryption in a way that the third party will only decrypt the ciphertext if the condition is in fact fulfilled. This can be realized with a verifiable encryption scheme with labels [11] where the condition is encoded in the label. This approach additionally allows the access control system to require provision of commitments of attributes which is relevant for issuing new certificates with a committed attribute without the issuer learning the attribute value. The signature protocols of [7, 8] support this.

Finally, we stress that all employed cryptographic algorithms are highly efficient and that our approach is complementary to existing work in the access control area in that it serves as a sort of front-end to existing attribute-based access control schemes and that it should hence be easy to integrate into the access control process. In particular, we plan to integrate our approach into the PRIME architecture [26], which constitutes a comprehensive architecture for privacy-enhancing identity management in open networks.

**Paper Outline.** Section 2 contains a more comprehensive overview of the approach and its benefits. Section 3 contains the detailed construction for the approach and in particular shows how to combine the necessary building blocks such as the access control system and the cryptographic protocols. Section 4 discusses further related work, and Section 5 concludes.

## 2. OVERALL APPROACH

In this section we give present our approach of access control, in particular a description of the setting and the preliminaries, certificates and types of statements that can appear in access control rules and that can be proved with proof protocols. An example scenario for which our approach is well applicable and the benefits of the approach compared to existing work in the field are presented.

### 2.1 Setting and Preliminaries

The setting used for the approach in this paper comprises four types of players: *Requesters*  $R$  of resources, *service providers*  $S$  that provide resources (e.g., services), *certifiers*  $C$  that issue certificates to others, and *trusted third parties*  $T$  that provide encryption keys and typically act as passive identity escrow authorities. A party  $T$  must be trusted by  $R$  to only decrypt if an agreed condition is fulfilled, and by  $S$  to actually decrypt if the condition is fulfilled. Each party  $T$  can theoretically be realized by multiple parties using the idea of threshold cryptography as put forth by Desmedt and Frankel [18] in order to weaken the trust assumptions regarding a single party. We do not yet consider such a distribution of  $T$  in our approach. Each party  $S$  has an *access control system* in place whose *rules* govern access to his resources. A party  $R$  has a *portfolio* of certificates that can be used in proof protocols. When a requester  $R$  submits a *resource request* for a resource to  $S$ , a *proof request* equivalent to a composition of the rules that apply to the resource is generated and sent back to  $R$ . The requester executes a *proof protocol* over certificates with  $S$ . Through the proof protocol  $R$  shows to  $S$  that she possesses certificates with attribute values and conveys encryptions and commitments to  $S$  as required by the access control rules. In case the proof succeeds,  $R$  is eligible for access to the requested resource.

We assume that a requester already has all certificates she can use in an interaction with a resource provider; this simplifies the presentation, but is still open to the approach of obtaining certificates within an interaction of a requester with a provider. Certificates in our setting are generalizations of anonymous credentials; the proof protocols we employ can be used to prove knowledge of certificates, prove statements on attributes, release encryptions of attributes with proofs that the encryptions actually contain the attributes and release commitments of attributes with proofs that they are commitments of the attributes.

The access control approach presented in this paper is useful for controlling access to any kind of resource, in particular also if the resource is a new certificate to be issued to the requester; taking this into account requires that commitments of certain attributes of certificates can be requested by the access control system. New certificates can be issued on the committed attributes by the issuer without him learning the values themselves. This is for example useful for including the user's identity in a new certificate while not learning the identity, but being sure to sign the user's identity.

### 2.2 Certificates and User Portfolios

Let  $C_1, \dots, C_w$  be the certificate issuers in the system. A *certificate*  $c_{(i,j)}$  issued by  $C_i$  is a list of attributes  $c_{(i,j)}.a_1, \dots, c_{(i,j)}.a_{u_i}$  and a signature  $\sigma_{(i,j)}$  over the attributes. Let  $\mathcal{D}_{(i,k)}$  with  $1 \leq k \leq u_i$  be the value domain for the attribute  $a_k$  of the certificate. All

certificates  $c_{(i,\cdot)}$  issued by  $C_i$  are of the same type and structure defined by the number and domains of the attributes.

The set  $\mathcal{C}_i$  contains all valid certificates  $c_{(i,j)}$  with  $j \geq 1$  that can be issued by party  $C_i$  assuming a system with fixed parameters. The space  $\mathcal{C}$  of all certificates in the system is defined as  $\mathcal{C} = \bigcup_{i=1}^w \mathcal{C}_i$ . A certificate  $c_{(i,j)}$  corresponds to a list of certified attributes  $a_{(i,j,1)}, \dots, a_{(i,j,u_i)}$ .

A certificate  $c_{(i,j)}$  being issued by  $C_i$  to  $R$  is element of  $\mathcal{C}_i$ . A certificate issuance means that  $C_i$  vouches for the binding of the attributes  $a_{(i,j,1)}, \dots, a_{(i,j,u_i)}$  of the certificate to the certificate receiver  $R$ . The issuance of a certificate  $c_i$  can be seen as the issuance of certified attributes  $a_{(i,j,1)}, \dots, a_{(i,j,u_i)}$ .

The set  $\tilde{\mathcal{C}}_i$  is the set of all certificates having been issued by certifier  $C_i$  to any party. Clearly,  $\tilde{\mathcal{C}}_i \subseteq \mathcal{C}_i$  holds. Let  $\tilde{\mathcal{C}} = \bigcup_{i=1}^w \tilde{\mathcal{C}}_i$  denote the set of all certificates that have been issued by all certifiers to any party.

Each requester  $R$  has a *portfolio*  $\tilde{\mathcal{C}}_R \subseteq \tilde{\mathcal{C}}$  of certificates, equivalent to a portfolio  $\tilde{\mathcal{A}}_R$  of attributes. The issuance of certificates to players  $R_l$  partitions the set  $\tilde{\mathcal{C}}$  such that each equivalence class  $\tilde{\mathcal{C}}_{R_l}$  is the set of certificates having been issued to one party  $R_l$ .

A certificate is bound to a user by requiring a user to use an important private key she owns every time she involves a certificate in a proof protocol [6]. Protecting the key sufficiently guarantees that no party except for the legitimate holder of the certificate can claim to be owner of the certificate. This strongly binds the certificate to the user and accounts for *non-transferability* of certificates.<sup>1</sup>

## 2.3 Statements over Attributes of Certificates, Encryptions, and Commitments

In our approach access control rules are used to create requests for proofs of statements over attributes of certificates, provision of encryptions of attributes of certificates, and provision of commitments of attributes of certificates. Within proof protocols, these requested statements can be proved and the encryptions and commitments conveyed.

### 2.3.1 Certified Attributes

A party can prove in zero-knowledge in a proof protocol that she knows signatures of certificates, i.e., she can prove that those certificates have been issued to her. She can furthermore prove polynomial relations over attributes of the certificates. These proofs realize a mechanism to exactly provide the information that is required by the access control system in terms of certified assertions over attributes, but nothing more. Examples of relations that can be proved are that the sum of two attributes is greater than a constant or that two attributes of different certificates are equal.

Within our condition language certified attributes are referred to by  $c_i.a_k$  meaning attribute  $k$  of any certificate  $c_{(i,j)}$  issued by  $C_i$ . Certificates and attributes can equally well be referred to by a descriptive identifier when assuming a mapping between these two representations.

### 2.3.2 Verifiable Encryptions

A verifiable encryption is a ciphertext resulting from an encryption of attributes or encryption of the evaluation results of polynomials over attributes together with a proof that the encrypted values are the attributes' values or the results of the polynomial evaluations. The key feature is that the plaintext tuple to the encryption be certified data and that this is proved to the receiver.

<sup>1</sup>Another approach to non-transferability would be to take the approach of tying an important secret to a certificate such that everyone who gets hold of the certificate gets access to the secret [6].

Let  $a_1, \dots, a_n$  be attributes or results of polynomials evaluated over attributes,  $PKE$  a public key for encryption,  $cond$  a textually stated condition, and  $encrypt$  an appropriate encryption algorithm, e.g., the one put forth by Camenisch and Shoup [11]. Then  $e_i = encrypt_{PKE}(a_1, \dots, a_n, cond)$  is the encryption of the tuple  $a_1, \dots, a_n$  with the public key  $PKE$  and condition  $cond$ . The condition is cryptographically bound to the ciphertext. The decryption will be successful only if the same condition  $cond$  is provided to the decryption algorithm.

In our approach the encryption is typically performed under a third party  $T$ 's public key. The condition used for the encryption is agreed between service provider and user and  $T$  is trusted by the user to decrypt only if the condition is fulfilled and trusted by the service provider to actually decrypt if the condition is fulfilled. Due to the cryptographic binding of the condition to the ciphertext,  $T$  cannot be tricked by the service provider to decrypt under a different condition. The verifiable encryption can be considered a conditional release of data, or an escrow, passively involving a third party. Verifiable encryption is a powerful tool for achieving accountability of transactions while maintaining anonymity of the requester.

Let the set of all encryptions of results of polynomials evaluated over attributes of certificates, all possible conditions, all possible encryption keys  $PKE_i$  and fixed algorithm parameters be  $\mathcal{E}$ . Let  $\tilde{\mathcal{E}}_{R,p}$  be the encryptions provided by a prover  $R$  in a considered protocol run  $p$  to a verifier  $S$ . In the condition language introduced further below, we refer to the underlying plaintext  $a_j$  of an encryption with  $e_{(i,j)}$ .

### 2.3.3 Commitments

A commitment of an attribute or of the result of the evaluation of a polynomial over attributes is a cryptographic object that binds the committing party to the value committed to while not revealing any information on the value to the receiving party. Conveying a commitment of a value within a proof protocol is required if the receiving party is to issue a new certificate that includes the committed value, however this without the party learning any information on the value. In this case, the access control rule protects the service of issuing a new certificate of a certain type and requires a commitment of attributes of certificates to be released. If these and the other information being requested are provided, a new certificate can be issued including the attributes being backed by the commitments without the issuer obtaining any information on the attributes. Let  $PKC$  be the public parameters of a commitment scheme. Let  $comm_i = commit_{PKC}(a_1, \dots, a_n)$  be a commitment of a tuple of values  $a_i$ .

The set  $\mathcal{M}$  is the set of all valid commitments for fixed system parameters. The set  $\tilde{\mathcal{M}}_{R,p} \subseteq \mathcal{M}$  refer to the set of commitments being provided by a party  $R$  within a proof protocol instance  $p$ . In the condition language further below, we refer to the underlying value  $a_k$  of a commitment  $comm_i$  as  $comm_{(i,k)}$ .

## 2.4 Condition Language

The condition language introduced below is used for expressing access control rules, proof requests, and proof specifications. A well-formed sentence over the condition language is a logical formula with predicates.

A predicate in a logical formula used within the framework in this paper is a statement over a set of constants and variables where a variable refers either to the value of a certified attribute, a plaintext value of an encryption, or the underlying value of a commitment. All values are integers. A predicate for algebraic relations between objects can express polynomial expressions over the in-

tegers as long as the integers remain within an attribute domain depending on the chosen parameters.

Supported relational operators between constants, certified attributes, encryptions, and commitments are  $=, \neq, <, \leq, >, \geq$ . A reference to a certified attribute without a relational operator to another variable is an implicit operator meaning knowledge of a signature on the attribute. Any predicates are constructed using the operators described above. An algebraic relation involving certificates  $c_{i_1}, \dots, c_{i_k}$ , encryptions  $e_1, \dots, e_l \in \mathcal{E}$ , and commitments  $comm_1, \dots, comm_m \in \mathcal{M}$  means precisely the algebraic relation being expressed over the underlying values of attributes or results of polynomials evaluated over attributes.

See the example in (1) below for a predicate specifying possession of a certified attribute of type  $c_1.a_1$ . The example in (2) specifies that the sum of the certified attributes  $c_1.a_1$  and  $c_2.a_1$  be greater than or equal to the constant 2000. The example in (3) specifies that the encryption  $e_{(1,1)}$  be  $encrypt_{PKE}(c_1.a_1 + c_2.a_1, cond)$  where  $encrypt$  is an encryption algorithm,  $cond$  is a condition that is cryptographically bound to the ciphertext, and  $PKE$  is a public key for encryption. The latter two items are abstracted from the specification language. The example in (4) refers to a commitment of the sum of the two attributes.

$$c_1.a_1 \quad (1)$$

$$c_1.a_1 + c_2.a_1 \geq 2000 \quad (2)$$

$$e_{(1,1)} = c_1.a_1 + c_2.a_1 \quad (3)$$

$$comm_{(1,1)} = c_1.a_1 + c_2.a_1 \quad (4)$$

**Definition 1** *The condition language  $L$  is the set of correctly typed formulas using the assumed universe of predicates and in the syntax of propositional logics without negation and with predicates.*

Whenever a formula of the condition language is fulfilled by a proof, only the commitments and encryptions referred from the term in the disjunctive normal form of the formula that the prover has actually fulfilled with certificates, contain values as specified. Others contain random values, unless they also occur in the fulfilled term of the DNF. See [12] for details.

## 2.5 Example

Consider that one can apply for renting an apartment only if one either proves with a bank statement that her monthly income is above a certain threshold or one proves with a bank statement on her monthly income and a governmental statement on being subsidized for housing that both together meet the threshold. Many people would not want to release the information that they get supported by the state as this would immediately allow one to infer general statements about their social status, i.e., whether they are receiving governmental subsidy. The approach in the paper allows for hiding the information which of the two conditions holds for a user, by only proving the logical disjunction of the two to be true. The user can be held accountable for her actions in the context of the transaction by requiring her to encrypt her identity under the key of a third party and prove that the ciphertext indeed is an encryption of her identity as contained in a certificate.

The following example is a proof specification meaning that one proves that one has sufficient income using an income certificate issued by a bank or an income statement together with a subsidy certificate issued by an appropriate institution. An encryption of the user's full name as in the user's passport certificate is provided in  $e_{(1,1)}$ . To have accountability of the user's actions, we assume that the full name identifies the user in the context of the passport

issuer. We abstract from language elements for representing the encryption key being used and the revocation condition attached to the encryption.

$$\begin{aligned} & (Income.amount \geq 2000 \wedge e_{(1,1)} = Passport holder \wedge \\ & \quad Income holder = Passport holder) \\ \vee & (Income.amount + Subsidy.amount \geq 2000 \wedge \\ & \quad Income holder = Subsidy holder \wedge \\ & \quad e_{(1,1)} = Passport holder \wedge Income holder = Passport holder) \end{aligned}$$

The verifier of a proof following the above specification will not know whether the prover has used just an income statement that vouches for an income greater than or equal to 2000 or an income statement and a subsidy statement that together vouch for income greater than or equal to 2000. The verifier can be sure that the encryption  $e_{(1,1)}$  contains the full name of the prover as contained in her government-issued passport. The proof furthermore ensures that the certificates a proof is conducted over have been issued to the same party as the holder attributes are proved to be equal.

In a traditional attribute-based access control system, the information on whether the requester has used a subsidy statement in her proof would leak to the access control system although it is not at all required for computing the access decision. Furthermore, accountability as guaranteed in the approach of this paper would not be supported by those traditional systems. Thus, we considerably enhance the privacy for the user in this and conceptually similar scenarios.

## 2.6 Benefits of the Approach

The key benefits of the presented approach are that it offers anonymity in end-user's transactions over electronic media while preserving user accountability of her actions at the same time by requiring the requester to provably convey encryptions of certain certified attributes that allow to identify the user. A decryption of the attributes can only be done by the third parties  $T_i$  under whose keys the encryptions were computed and if the condition being bound to the ciphertext is used. These two properties of the access control approach constitute a significant improvement to today's approaches.

The approach guarantees the privacy of the requester, i.e., not more information than specified by the requester is released to the resource provider. Furthermore it guarantees the resource provider that if a request is granted, the requester knows (owns) the certificates that are required by his rules and has provided proper encryptions and commitments. In particular, a rule can express a disjunction of statements on certificates' attributes which can be particularly useful in decreasing the released information compared to today's approaches. Both guarantees are backed through correctness proofs of both the access control system and the certificate proof system and the interoperation of these systems.

Recapitulating, the approach in this paper is a further step towards implementing the purpose limitation and data quality principles as put forth by the data protection legislation, see, e.g., [19]. The former is achieved by using the proof protocols of [12] conveying the minimal required information to the resource provider, the latter by using certified data within these proof protocols.

## 3. DETAILED CONSTRUCTION

In this section we present the two key building blocks for our approach, an access control system and a proof system. We show how to combine those to derive our approach to privacy-enhanced access control.

### 3.1 Access Control System

The approach in this paper needs an arbitrary access control system fulfilling the following properties: The access control rules must be able to express formulas over attributes, encryptions, and commitments with an expressivity of our condition language to make full use of our approach. An access decision is based on proof specifications of successfully executed proofs of a requester. If a submitted request may not be fulfilled, the access control system returns an equivalent of the yet unfulfilled rules, a so called proof request, to the requester; the proof request produced by the access control system is a well-formed sentence in our condition language  $L$ . A proof request defines precisely what proof protocol over certificates has to be conducted by a prover in order to be eligible for accessing the requested resource. The system is an access control system of an attribute-based-like type, but using specifications of proof protocols over private certificates instead of just attribute values of certificates that are sent to the service provider to allow for the decision.

#### 3.1.1 Rules

Access control rules, or access conditions, are used to define requirements a requester must fulfill in order to obtain access to a resource. Multiple different conjunctions or disjunctions of predicates may each allow access to a resource; the access control rules are expressed with the condition language  $L$ . Encryptions and commitments require new language constructs analogous to the ones used for attributes that are not yet supported by current proposals of access control languages.

A resource can have multiple rules applying to it. In this case the rule to be applied is the one composed of all applicable rules. The appropriate composition of all applicable rules for a resource gives the proof request that is returned to a requester. The proof request can be thought of one dynamically generated aggregated rule that applies to the resource.

One access control system that is applicable to our framework has been put forth by Bonatti and Samarati [2]. This system fulfills the abovementioned requirements, but would still require that the rule evaluation engine be generalized such that it evaluates rules based on proof specifications of successful proof protocols instead of just attributes of certificates sent by the user. This extension can be accomplished for this access control framework [27]. In addition, extensions to the rule language to account for encryptions and commitments would be required.

#### 3.1.2 Satisfiability of a Rule

A rule is satisfiable by a requester  $R$  if there exists a set  $\tilde{C}_R$  of certified attributes having been issued to party  $R$  such that the rule is satisfiable in an interpretation over the universe  $\tilde{C}_R \cup \mathcal{E} \cup \mathcal{M}$  of the requester's portfolio of certificates and all possible encryptions and commitments that could be provided within a proof protocol.

A rule is satisfied by a proof protocol instance  $p$  by  $R$  if and only if the following two conditions are fulfilled:

1. The proof protocol with a proof specification over  $\tilde{C}_R$ ,  $\tilde{\mathcal{E}}_{R,p}$ , and  $\tilde{\mathcal{M}}_{R,p}$  has been successfully executed by  $R$ . The first of the sets refers to  $R$ 's portfolio, the latter two sets refer to the set of encryptions and commitments having been provided to the verifier in the proof protocol.
2. The proof specification of the proof protocol equals the rule.

Due to the interpretation of the proof specification over  $\tilde{C}_R$ ,  $\tilde{\mathcal{E}}_{R,p}$ , and  $\tilde{\mathcal{M}}_{R,p}$  this immediately implies that the prover of the proof knows signatures on the specified attributes and has provided encryptions and commitments as required by the specification.

A party  $R$  gets access to a resource if and only if  $R$  fulfills the rule with an instance of a proof protocol.

### 3.2 Proof System

The system for proving knowledge of certificates, proving polynomial relations between attributes of these certificates, and releasing verifiable encryptions and commitments used in the approach in this paper is the one of Camenisch et al. [12]. The proofs are performed over certificates of a party.

A holder of certificates can use the certificates in proofs between her as a prover and another party, the verifier. A proof can involve multiple certificates. In contrast to conventional certificates such as X.509-like ones, possession of certificates can be proved without sending the certificates. Polynomial relations between multiple attributes of certificates can be proved without releasing any other information on the attributes. Encryptions and commitments of attributes and results of polynomials evaluated over attributes can be released including proofs that actually specified values are encrypted or committed to. Overall, the prover can precisely specify in the proof specification what information to release in a particular proof protocol instance. Exactly this information is released in the protocol, and nothing more; this is cryptographically guaranteed due to the applied zero-knowledge proof techniques.

A signature over attributes in this proof system is not computed over a hash of attributes as in conventional X.509 certificates [23, 20], but the signature takes each attribute into account individually [7, 8] which is necessary to conduct proof protocols referring to individual attributes of certificates.

In contrast to conventional certificate systems, the system allows that certificates be issued on commitments, i.e., without the issuer learning the attribute to be certified. This allows for example that an issuer issue a certificate containing the name of the receiver where a commitment of the name attribute has been provided by the receiver in a previous proof protocol with another certificate. The issuer does this certification without learning any information on the name of the receiver.

#### 3.2.1 Basic Proof Protocol

The basic building block of the proof system is a signature scheme with protocols with the following two properties: A signature can be computed by the issuer on commitments of attributes without the issuer learning the committed values. It is possible to prove knowledge (possession) of the signature while not revealing it.

In the fundamental proof protocol knowledge of a signature is proved without revealing any other information on the signature or the attributes that the signature applies to. In particular, multiple show protocol instances involving the same signature are unlinkable. A proof protocol is unlinkable to its issue protocol. The proof protocol for a certificate requires the certificate to be input by the prover and the signature verification key and proof specification by both prover and verifier. The proof protocol for a signature involves randomizing the signature and proving knowledge of the randomized signature with a zero-knowledge proof protocol. See Camenisch and Lysyanskaya [7, 8] for details on the signature schemes and protocols for the schemes.

#### 3.2.2 General Proof Protocol

Based on the above-sketched basic protocol for proving knowledge of a signature, advanced protocols are constructed that accommodate—on top of the functionality of the basic protocol—the functionality of proving relations between certified attributes (of multiple certificates), performing verifiable encryptions, and making commitments and proving polynomial relations of those over

attributes. A proof specification defines what information a proof conveys. The proof specification is a well-formed formula over our condition language  $L$ .

The proof protocol requires the certificates to be input by the prover, the signature verification key, the proof specification, the encryption public keys and conditions for verifiable encryption, and the commitment public keys by both prover and verifier. In an instance of a proof protocol precisely the information as specified by the proof specification is conveyed to the verifier.

### 3.2.3 Cryptographic Construction

In the common parameters model, we build upon several known protocols for proving statements about discrete logarithms, such as proof of knowledge of a discrete logarithm modulo a prime [29] or a composite [21, 17], proof of knowledge of equality of representation modulo two (possibly different) prime [14] or composite [10] moduli, proof that a commitment opens to the product of two other committed values [9, 4], proof that a committed value lies in a given integer interval [13, 9, 3], and also proof of the disjunction or conjunction of any two of the previous [15, 28].

That is, we can use these protocols to efficiently prove the formulas we defined among attributes that are signed using the Camenisch-Lysyanskaya signature schemes [7, 8], encrypted using the Camenisch-Shoup encryption scheme, or committed using the Pedersen commitment scheme [25, 17], without revealing the attributes themselves. For obtaining concurrent zero-knowledge proofs, Damgård's construction [16] is applied.

## 3.3 Combining the Building Blocks

In this section, we state a theorem on the security of the approach for access control in this paper and give a proof sketch. We assume adversaries to be polynomially bounded.

**Theorem 1** *If a requesting user  $R$  is granted access to a resource of a service provider  $S$  after having performed a proof protocol  $p$ , then the following properties hold:*

- 1) Security for the service provider:  *$R$  has run the proof protocol  $p$  with  $S$  successfully also providing encryptions and commitments such that the rule for the resource is fulfilled in an interpretation over  $\tilde{C}_R \cup \tilde{E}_{R,p} \cup \tilde{M}_{R,p}$  with overwhelming probability.*
- 2) Privacy for the requester: *The requesting user  $R$  has released no information except that she holds a set of certificates that satisfy the rule provided that no encryptions she has released are decrypted by a trusted third party  $T_i$ .*

We next provide a very rough sketch of the proof of the above theorem. This very high-level proof sketch intends to point out the main properties of the mechanisms we employ that lead to Theorem 1 of the access control approach of this paper. For rigorous proofs of each of the involved mechanisms we refer the reader to the literature.

**PROOF OF THEOREM 1 (SKETCH).** Informally, property 1) states that a service provider having granted a requester access to a resource has been convinced by the requester that she has certificates whose attributes fulfill the access conditions for the resource. In addition, encryptions and commitments of attributes have been provided as required in the rule with the verifier being convinced that they encrypt and commit the specified certified values. Being convinced means that the requester could have only cheated with negligible probability which is guaranteed by the cryptographic mechanisms underlying the proof. The provided encryptions give the service provider a guarantee of accountability of the user's actions.

When a requester has been granted access to the requested resource, this implies that the cryptographic proof protocol of possession of certificates that fulfill the access control rule including encryptions and commitments she has provided was successfully executed. It follows immediately from the unforgability property of the signature schemes [7, 8] and the soundness of the zero-knowledge proof protocols under the assumption of polynomially bounded adversaries that, as the proof convinced the verifier, the prover indeed has certificates fulfilling the proof and has provided correct encryptions and commitments, except for with negligible probability.

Property 2) follows in part from the zero-knowledge property of the proofs employed within the overall proof protocol. The security of the encryption scheme used for verifiable encryption [11] guarantees that the encrypted attributes be computationally hidden from any party that does not have access to the decryption keys. The information theoretic hiding property of the Pedersen commitment scheme [25] guarantees that the commitments do not leak any information on the committed values. The completeness of the zero-knowledge protocol guarantees that the verifier can be convinced.

The overall construction is secure in the common parameters model due to the properties of the mechanisms we combined.  $\square$

## 4. FURTHER RELATED WORK

We point to further related work in this section that is related to the approach of this paper.

Access control paradigms that base their decision on identities are less suitable for open networks than attribute-based systems. The attribute-based access control approach enjoys multiple advantages over pure identity-based access control, both since it offers a greater flexibility for a user in terms of which information she wants to release and since unique identities typically have no or only a restricted common meaning to service providers in open networks, whereas attributes can often be interpreted in a uniform manner by multiple different service providers. One candidate access control system for the approach presented in this paper is the one of Bonatti and Samarati [2]. We note that our extensions to their framework are orthogonal to the basic functionality of their framework. Other attribute-based access control approaches such as [24, 30, 31, 32] could benefit from our ideas as well.

The notion of zero-knowledge that is fundamental to the proof system we use has been introduced by Goldwasser, Micali, and Rackoff [22]. The signature schemes for issuing signatures on and proving knowledge of signatures with respect to committed values have been put forth by Camenisch and Lysyanskaya [7, 8]. The first scheme is a generalized approach to the anonymous credential system of Camenisch and Lysyanskaya [6]. The first of these schemes has been further optimized regarding parameter sizes and performance by Camenisch and Groth [5]. The proof framework that is based on the aforementioned mechanisms has been put forth by Camenisch et al. [12]. A more basic proof framework which is—mainly due to a lack of capabilities to proof disjunctions—not suitable for our general approach is due to Bangerter et al. [1].

## 5. CONCLUSION

We have presented an approach for augmenting attribute-based access control systems in a way that allows them to offer fully anonymous access to resources while at the same time achieving strong accountability guarantees. Essentially, our approach exploits cryptographic zero-knowledge proofs to allow requesting users to prove that they hold suitable certificates for accessing a resource without granting the access control system any additional informa-

tion beyond that. This constitutes the minimal amount of information that a user has to release so that the access control system can come up with a correct access decision, and our approach is the first to achieve this. In particular, the privacy guarantees that the approach offers to the users stand in blatant contrast to the commonly taken approach of sending all possibly relevant certificates to the access control system. Moreover, the approach can be seen as a complementary mechanism to existing work in the access control area in that it serves as a sort of front-end to existing attribute-based access control schemes and should hence be easy to integrate into the access control process. All employed cryptographic algorithms are furthermore highly efficient and are hence unlikely to become the performance bottleneck of the overall system.

**DISCLAIMER** The information in this document reflects only the author's views, is provided as is and no guarantee or warranty is given that the information is fit for any particular purpose. The user thereof uses the information at its sole risk and liability.

## 6. REFERENCES

- [1] BANGERTER, E., CAMENISCH, J., AND LYSYANSKAYA, A. A cryptographic framework for the controlled release of certified data. In *Security Protocols 2004*, LNCS.
- [2] BONATTI, P. A., AND SAMARATI, P. A uniform framework for regulating service access and information release on the web. *J. Comput. Secur.* 10, 3 (2002), 241–271.
- [3] BOUDOT, F. Efficient proofs that a committed number lies in an interval. In *EUROCRYPT 2000*, vol. 1807 of LNCS, Springer Verlag, pp. 431–444.
- [4] BRANDS, S. Rapid demonstration of linear relations connected by boolean operators. In *EUROCRYPT '97*, vol. 1233 of LNCS, Springer Verlag, pp. 318–333.
- [5] CAMENISCH, J., AND GROTH, J. Group signatures: Better efficiency and new theoretical aspects. In *SCN 2004*, pp. 120–133. LNCS, Springer Verlag.
- [6] CAMENISCH, J., AND LYSYANSKAYA, A. Efficient non-transferable anonymous multi-show credential system with optional anonymity revocation. In *EUROCRYPT 2001*, vol. 2045 of LNCS, Springer Verlag, pp. 93–118.
- [7] CAMENISCH, J., AND LYSYANSKAYA, A. A signature scheme with efficient protocols. In *SCN 2002*, vol. 2576 of LNCS, Springer Verlag, pp. 274–295.
- [8] CAMENISCH, J., AND LYSYANSKAYA, A. Signature schemes and anonymous credentials from bilinear maps. In *CRYPTO 2004*, LNCS, Springer Verlag.
- [9] CAMENISCH, J., AND MICHELS, M. Proving in zero-knowledge that a number  $n$  is the product of two safe primes. In *EUROCRYPT '99*, vol. 1592 of LNCS.
- [10] CAMENISCH, J., AND MICHELS, M. Separability and efficiency for generic group signature schemes. In *CRYPTO '99*, vol. 1666 of LNCS, Springer Verlag, pp. 413–430.
- [11] CAMENISCH, J., AND SHOUP, V. Practical verifiable encryption and decryption of discrete logarithms. In *CRYPTO 2003*, LNCS. Springer Verlag.
- [12] CAMENISCH, J., SOMMER, D., AND ZIMMERMANN, R. A general certification framework with applications to privacy-enhancing certificate infrastructures. Tech. Rep. RZ 3629, IBM Research, July 2005.
- [13] CHAN, A., FRANKEL, Y., AND TSIOUNIS, Y. Easy come – easy go divisible cash. In *EUROCRYPT '98*, vol. 1403 of LNCS, Springer Verlag, pp. 561–575.
- [14] CHAUM, D., AND PEDERSEN, T. P. Wallet databases with observers. In *CRYPTO '92*, vol. 740 of LNCS, Springer-Verlag, pp. 89–105.
- [15] CRAMER, R., DAMGÅRD, I., AND SCHOENMAKERS, B. Proofs of partial knowledge and simplified design of witness hiding protocols. In *CRYPTO '94*, vol. 839 of LNCS, Springer Verlag, pp. 174–187.
- [16] DAMGÅRD, I. Efficient concurrent zero-knowledge in the auxiliary string model. In *EUROCRYPT 2000*, vol. 1807 of LNCS, Springer Verlag, pp. 431–444.
- [17] DAMGÅRD, I., AND FUJISAKI, E. An integer commitment scheme based on groups with hidden order. In *ASIACRYPT 2002*, vol. 2501 of LNCS.
- [18] DESMEDT, Y., AND FRANKEL, Y. Threshold cryptography. In *CRYPTO '89*, vol. 435 of LNCS, pp. 307–315.
- [19] EUROPEAN PARLIAMENT. Directive 95/46/ec on the protection of individuals with regard to the processing of personal data and on the free movement of such data. *Official Journal of the European Communities* 31 (1995).
- [20] FARRELL, S., AND HOUSLEY, R. An Internet Attribute Certificate Profile for Authorization. RFC 3281 (Proposed Standard), Apr. 2002.
- [21] FUJISAKI, E., AND OKAMOTO, T. Statistical zero knowledge protocols to prove modular polynomial relations. In *CRYPTO '97*, vol. 1294 of LNCS, pp. 16–30.
- [22] GOLDWASSER, S., MICALI, S., AND RACKOFF, C. The knowledge complexity of interactive proof systems. In *FOCS 1985*, pp. 291–304.
- [23] HOUSLEY, R., POLK, W., FORD, W., AND SOLO, D. Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile. RFC 3280 (Proposed Standard), Apr. 2002.
- [24] LI, N., MITCHELL, J. C., AND WINSBOROUGH, W. H. Design of a role-based trust management framework. In *Proc. IEEE Symposium on Security and Privacy* (2002).
- [25] PEDERSEN, T. P. Non-interactive and information-theoretic secure verifiable secret sharing. In *CRYPTO '91*, vol. 576 of LNCS, Springer Verlag, pp. 129–140.
- [26] Prime whitepaper. Whitepaper, 18 July 2005. <http://www.prime-project.eu.org/prime/public/press-room/whitepaper/PRIME-Whitepaper-V1.pdf>.
- [27] SAMARATI, P., June 2005. Personal communication.
- [28] SANTIS, A. D., CRESCENZO, G. D., PERSIANO, G., AND YUNG, M. On monotone formula closure of SZK. In *FOCS 1994*, IEEE, pp. 454–465.
- [29] SCHNORR, C. P. Efficient signature generation for smart cards. *Journal of Cryptology* 4, 3 (1991), 239–252.
- [30] YU, T., MA, X., AND WINSLETT, M. Prunes: an efficient and complete strategy for automated trust negotiation over the internet. In *ACM CCS '00*, ACM Press, pp. 210–219.
- [31] YU, T., WINSLETT, M., AND SEAMONS, K. E. Interoperable strategies in automated trust negotiation. In *ACM CCS '01*, pp. 146–155.
- [32] YU, T., WINSLETT, M., AND SEAMONS, K. E. Supporting structured credentials and sensitive policies through interoperable strategies for automated trust negotiation. *ACM Trans. Inf. Syst. Secur.* 6, 1 (2003), 1–42.