A Secure Channel for Attribute-Based Credentials

[Short Paper]

Gergely Alpár*
Radboud University Nijmegen, ICIS DS
Nijmegen, The Netherlands
gergely@cs.ru.nl

Jaap-Henk Hoepman
Radboud University Nijmegen, ICIS DS
Nijmegen, The Netherlands
jhh@cs.ru.nl

ABSTRACT

Attribute-based credentials (ABCs) are building blocks for user-centric identity management. They enable the disclosure of a minimum amount of information about their owner to a verifier, typically a service provider, to authorise the credential owner for some service, application, or resource.

By directly applying attribute-disclosure protocols, the data is revealed not only to the verifier, but anyone who has access to the communication channel. Moreover, as verifiers are not intrinsically authenticated, one can accidentally reveal attributes to the wrong party. Therefore, a secure channel has to be established between the prover and the verifier.

Although efficient ABC smart-card implementations exist, not always can they perform all prover features. An equality proof, for instance, is essential in creating pseudonyms that enable temporary identification and eventually establishing a channel. Without this feature, other techniques have to be developed. In this paper we apply a more general notion of authentication that does not require card identification or pseudonyms. Based on this concept, we propose a security model that includes mutual authentication and setting up a channel between a card and a verifier. We present two efficient and provably secure protocols under standard assumptions in the random oracle model.

Categories and Subject Descriptors

D.2.0 [**Computer-Communication Networks**]: General—*Security and Protection*; K.6.5 [**Management of Computing and Information Systems**]: Security and Protection—*Authentication*; K.4.1 [**Computers and Society**]: Public Policy Issues—*Privacy*

Keywords

identity management; cryptography; secure channel; mutual authentication; anonymous credential; attribute-based credential

1. INTRODUCTION

As individuals perform an increasing amount of transactions online, privacy is of primary importance in the digital world. For many of these transactions, for instance, the full identity of a user is not essential, only a few non-identifying attributes are sufficient to be revealed (*e.g.*, "over 18" for buying alcoholic drinks). Traditionally, authentication means the process of proving the ownership of one's identity. A more general notion of authentication is the process of proving that certain predicates, or attributes, hold for an entity.

Attribute-based credentials [5, 8, 7] make it possible to perform such proofs. An attribute-based credential (ABC), a cryptographic container of attributes related to an individual, is signed by an authoritative issuer that attests to these attributes. As attributes in an ABC can be shown independently of one another, its owner can adaptively use them in different applications. These proofs are called *selective disclosure* proofs. Moreover, each instance of showing an ABC looks unrelated to other instances and also to the issuing of this credential.

A tamper-resistant, personal smart card is a viable choice to carry ABCs. Several studies (*e.g.*, [3, 12, 15]) show that the most essential functionalities are feasible to implement on currently available smart cards. However, not all features can be implemented on these resource-limited devices. Due to their complexity, proofs of equality of attributes in separate credentials or property proofs about attributes (like, an attribute lies in an interval or is an element of a set) require more memory (RAM) than available on some—otherwise suitable—platforms. This work is motivated by the IRMA project[1], in which users' smart cards cannot run equality or property proofs.

A verifier is allowed to access only those attributes from a card that are necessary for the authorisation. This is described by an access policy that is included in a public-key certificate provided to each verifier. Therefore, when a verifier and a card start communicating, the verifier sends its

*Partly supported by the research program Sentinels as project 'Mobile IDM' (10522). Sentinels is being financed by Technology Foundation STW, the Netherlands Organisation for Scientific Research (NWO), and the Dutch Ministry of Economic Affairs.

[1] IRMA stands for 'I Reveal My Attributes'. The IRMA project is based on an efficient smart-card implementation of Idemix [15]. For more information, see `https://www.irmacard.org`.

DIM'13, November 8, 2013, Berlin, Germany.
Copyright 2013 ACM 978-1-4503-2493-9/13/11 ...$15.00.
http://dx.doi.org/10.1145/2517881.2517884.

certificate and the card checks its access rights and adheres only to allowed attribute queries later.

A secure channel is essential for ABC proofs since a selective disclosure protocol requires protection of a card-holder's privacy against an eavesdropping adversary for different reasons.

- Disclosed attributes: Obviously, confidentiality should be provided for personal data during the authentication process; *e.g.*, identifying attributes.
- Verifier's request: Attribute requests may reveal information about provided services; *e.g.*, if a service provider asks for "over 18", it may leak information about the sort of movie a user watches.
- Issuer's signature: Information about the credential issuer may give hints about the type of credentials and/or the values of attribute; *e.g.*, the signature of a credential issued by an employer reveals where an individual works.

Not only eavesdropping, but also active attacks can threaten system security. First, when proofs are not linked to each other, which is the case in smart-card protocols due to resource constraints, a verifier cannot be assured that they are originating from the same device. For instance, a membership attribute on one card could be combined with an "over 18" attribute on another card. A secure channel that is bound to one device on the prover's side can prevent this, so-called card-pooling attack. Second, without a secure channel, a classical man-in-the-middle attack can be set up. For instance, acting as a card and using proofs from a real card, a rogue verifier could access some service at another verifier. The same adversary could also inject proofs from other devices that affect the authorisation decision of the verifier.

A secure channel provides additionally the benefit of a session that can link verification of (potentially non-identifying) attributes from already existing credentials on a card and the issuance of new ones. This sort of issuing procedure, in which new credentials rely on non-identifying attributes, is also in line with the original notion of an (anonymous) credential [9].

To establish a channel with a verifier, a card cannot reveal a unique, card-specific identifier as it would destroy the privacy properties of the ABC technology. Hence, a new notion of validity is required to realise authentication. A card is regarded as valid (or authentic) if it holds a particular credential and thus it can perform a proof about it. For example, a national identity card would be considered as such if it can prove that it carries an attribute-based credential issued by the state authority responsible for electronic identity cards.

Throughout this paper individuals are assumed to hold personal cards used to disclose relevant attributes in order to access some service. Each card carries attribute-based credentials and a root credential [1] which, by an empty proof, shows validity of the card. This paper introduces a security model and proposes two protocols that realise items 1a, 1b, and optionally 3b in the following list of steps in the context of ABCs:

1. Authentication
   (a) *Establishing an authenticated secure channel*
   (b) *Selective disclosure within the channel*
2. Authorisation decision based on selectively disclosed information
3. Accessing service
   (a) Resource, application; or
   (b) [Optional] Credential issuance within the channel

1.1 Attribute-Based Credentials

An attribute-based credential is a cryptographic container for attributes represented as bit-strings. A credential is signed by an issuer using some special signature that provides the following functionalities.

- A zero-knowledge proof can be produced by the holder of the credential that reveals any desired subset of the attributes along with the issuer's signature and a proof that the disclosed attributes are actually in the credential. This mechanism is called a ***selective disclosure***, or simply a verification protocol.
- Verification instances carried out using a credential are unlinkable, unless attributes make them linkable.
- A verification instance is also unlinkable to the issuance of the credential.

As a result, individuals, using attribute-based credentials, can prove a minimum amount of information about themselves. The two most important technologies that realise attribute-based credentials are Microsoft's U-Prove [5, 4] and IBM's Idemix [8, 14], which are being put in a unified architecture by the European ABC4Trust project [7].

A selective disclosure is a proof protocol that starts with the verifier's request for certain attributes and it includes a fresh, random nonce n. As a response, the revealed attributes and the credential or its randomised version are transmitted together with a non-interactive zero-knowledge (NIZK) proof. This proof is also a signature on the nonce, which demonstrates *freshness*; *e.g.*, no replay attack is possible. The following simple notation is introduced for the NIZK proof that the card provides:

$$\mathsf{SD}\left((a_i)_{i\in\mathcal{D}}; n\right),$$

where $\mathcal{D}$ is the set of indices i corresponding to attribute i in a given credential and n is the message/nonce to be signed. We implicitly assume that in an implementation the message also contains a unique description of the context of the protocol run, like $n\|\mathtt{context}$; this will be omitted from now on. A selective disclosure is called an ***empty proof*** if no attribute is revealed (*i.e.*, $\mathcal{D} = \emptyset$), only the mere existence of a credential is proven. Note that during selective disclosure proofs the identity of the credential issuers are revealed just like that of a certificate authority in case of public-key certificates.

An optional ABC functionality is an ***equality proof*** to show the equality of two secret values. Roughly speaking, such a zero-knowledge proof requires two times as much working memory (*i.e.*, RAM). If equality proofs are not feasible in a particular card implementation, in order for a set of attributes residing in different credentials to be proven, one has to perform as many selective disclosure proofs as the number of distinct credentials involved. (Otherwise, proofs of equality of the non-disclosed master secret key are required.)

Since during verifications users do not necessarily reveal identifying attributes, *revocation* is especially challenging.

Lapon et al. [11] give a comprehensive theoretical and practical comparison of different techniques to revoke ABCs that achieve anonymity and revocability simultaneously.

1.2 Authenticated Secure Channel

Although many authentication and key-exchange protocols have been proposed at a very early stage of cryptography, Bellare and Rogaway [2] are the first who studied authenticated key exchange rigorously in a cryptographic sense.

In [2] participants are de-coupled and all communication is controlled by an active adversary. To show security of an authentication protocol, one has to prove that the probability for this adversary to make participants accept the other's authenticity is negligible unless all messages are conveyed according to the protocol. The main tool to capture this notion is the so-called *matching conversation*. By attaching some extra information to the mutual authentication protocol, the authors achieve efficient and provably secure key-exchange protocols. The adversary is so powerful that she can query all secret session information from any participant. The security of a key exchange protocol is defined as the indistinguishability of a fresh (not queried) session key from a random string.

Our security model builds on [2], and it incorporates asymmetric and attribute-based authentication.

1.3 Related Techniques

A pseudonym [9, 8] is bound to a user's credentials, but it does not reveal anything about credential keys or the user's identity. A pseudonym is similar to a public key within a proof session, but there are practically an infinite number of pseudonyms corresponding to a secret key, which allow for unlinkability among separate sessions or contexts. Applying pseudonyms however requires equality proofs, that is, zero-knowledge proofs that the same secret key α was used to generate the pseudonym and verification proofs. In our model this is not considered to be feasible.

The German e-identity project [13] provides privacy-friendly proofs for citizens based on attributes, such as "over 18" age proofs. However, it employs a different approach from ABCs to achieve anonymity for particular identity cards. Each card has a public key (so-called chip authentication key) that enables authentication or channel establishment. A public key is not assigned to a single card, but a batch of cards. Therefore, batches can be identified, but not cards (or card holders). In practice, an appropriate batch size has to be determined. To achieve a proper level of anonymity, batches should not be too small. On the other hand, too big batch sizes result in infrastructural problems. In case of a card with a corresponding private key gets compromised, not only this particular card has to be revoked, but all cards of its batch.

The CAID and CAKE (credential-authenticated identification and key exchange) protocols [6] are introduced for authentication and key exchange using credentials. These protocols are proven to be secure in the universal composability framework. Although the motivation and the results closely related to ours, the present study assumes minimum about the resources of users' devices (*e.g.*, very simple policy), and there is no equality proof of attributes, in particular.

1.4 Our Contributions

In this paper we elaborate on ABC authentication within a secure channel. We propose to adapt the security model of [2] to ABC systems in which the prover's resources are very limited and the user is identified only to the extent of the attribute proofs included. We put forward two protocols and briefly discuss their security in the introduced model. Both protocols are practical and efficient in the sense that the computation and working memory overhead are much less than (or, in the case of the second protocol, comparable to) the resources ABC proofs require.

2. SECURE ABC CHANNELS

2.1 Security Model

In order to set up a secure channel for selective disclosure, a card and a verifier have to mutually authenticate each other. However, since being "valid" is different at an anonymous card and a legitimate verifier, authentication has to be defined on both sides separately. A card is considered to be valid if it can carry out a whole verification protocol that includes an empty (selective disclosure) proof of the root credential. A verifier on the other hand has to show a valid public-key certificate and prove knowledge of the corresponding secret key.

In this model, sessions of cards and verifiers are modelled as oracles that follow the protocol. The adversary is a polynomial-time algorithm that controls the whole communication among oracles. The adversary's goal is to win one of two games, *i.e.*, to break one of the following two security properties of the system. First, it can try to eavesdrop on revealed but encrypted attributes thus compromising confidentiality. This is captured by the notion of *indistinguishability* of a session key and a random string. Second, the adversary can forge fake authentication by convincing a party that it is "talking" to a valid counter-party while this is not the case. This is captured by the unfeasibility of counterfeiting *matching conversations*. If an adversary has only negligible advantage in both games, the protocol is a *secure ABC session protocol*. Furthermore, a revocation mechanism will be considered to be *ideal* if any card corruption results in immediate card revocation.[2]

2.2 Implicit Card Authentication

We introduce an efficient ABC session protocol in Figure 1, which is called the implicit card authentication protocol or ***ICA***. The verifier and a presumed card establish a key k that is used to provide a secure channel based on k for the selective disclosure proofs. Note that unlike most authenticated key exchange protocols, the card's validity can only be verified within the channel, which explains the name of this protocol.

We assume that the verifier's public key is an initial input value to the card. Moreover, the card is also privy to the description of which attributes this verifier is eligible to request. In practice, a public-key certificate is sent to the card, from which it can extract and verify pk_V and attribute access rights. The verifier's private initial input is its secret key sk_V.

[2] A complete description of the security model and the full proofs of the protocols will appear in an extended version of this paper.

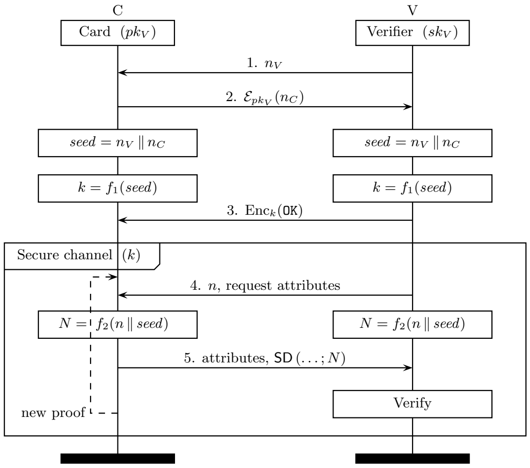
C
V
Card (pk_V)
Verifier (sk_V)
1. n_V
2. $\mathcal{E}_{pk_V}(n_C)$
$seed = n_V \parallel n_C$
$seed = n_V \parallel n_C$
$k = f_1(seed)$
$k = f_1(seed)$
3. $\mathrm{Enc}_k(\mathtt{OK})$
Secure channel (k)
4. n, request attributes
$N = f_2(n \parallel seed)$
$N = f_2(n \parallel seed)$
5. attributes, $\mathsf{SD}(\ldots; N)$
Verify
new proof

Figure 1: Implicit card authentication *ICA*: A key exchange for selective disclosure precedes card authentication.

A verifier initiates the protocol ***ICA*** by sending a random nonce n_V chosen uniformly at random from the nonce space. Then, the card also generates a random nonce n_C from the same nonce space. It encrypts n_C using the public key pk_V and sends $\mathcal{E}_{pk_V}(n_C)$ to the verifier. After receiving $\mathcal{E}_{pk_V}(n_C)$, the verifier can decrypt it and compute n_C. Both participants can now compute and store *seed* and the channel key $k = f_1(seed)$. However, only the verifier has already authenticated by being required to know its private key. Applying explicit key confirmation, the verifier sends flow 3. $\mathrm{Enc}_k(\mathtt{OK})$, an OK message encrypted (by authenticated encryption) using the freshly established key k. The card checks it to get convinced that they share the same key (`KeyOK`) and that it indeed communicates with the intended verifier (`Accept`).

Within the established secure channel the two parties perform the selective disclosure proofs that eventually provide card authenticity. To initiate each selective disclosure, the verifier sends a fresh, random nonce n and requests some attributes to be disclosed. A verifier only requests a set of attributes that reside in *one* credential since cards are not assumed to be able to link proofs from different credentials. The card checks that the verifier is entitled to request these attributes. If not, the card rejects the request. Otherwise, it computes $N = f_2(n \| seed)$ and using it as a fresh nonce, the card generates a non-interactive selective disclosure that reveals the requested attributes and proves that they reside in the credential. The verifier checks the proof using N and stores the disclosed attributes. After all credential proofs with the required attributes were requested and performed successfully, the verifier terminates and outputs `Accept` and a set A of revealed and verified attributes.

THEOREM 1. *Assume that a CPA-secure public-key encryption $(\mathcal{G}_{pke}, \mathcal{E}, \mathcal{D})$ is given to encrypt confidential messages to verifiers, a secure ABC technology is given for selective attribute disclosure with ideal revocation, and an authenticated encryption $\mathrm{Enc}_k(\cdot)$ is given for secure channels with arbitrary keys from the key space. Moreover, let f_1, f_2 be random oracles with all participants (including the adversary) having access to them. Then the implicit card authentication protocol **ICA** is a secure ABC session protocol.*

PROOF SKETCH. Clearly, two parties following the protocol can both `Accept` under a benign adversary. Furthermore, the verifier receives all eligibly requested attributes that it can output at the end of the protocol.

1. The subprotocol ***ICA***^{key} that establishes a key by the first two flows is confidential; that is, seeing only the message flows, an adversary cannot distinguish the resulting key k from a random string of the same length. If flow 3. is omitted, ***ICA*** is also confidential. (Without the omission of the explicit key confirmation, an adversary would trivially win.)

2. The nonce transformation $f_2(n \| seed)$ in the secure channel cannot be existentially forged without knowing both the nonce n and the *seed*. This provides freshness and binds the key exchange to the selective disclosure proofs.

3. It is impossible to combine selective disclosure proofs coming from combined sources, that is, both from corrupted and uncorrupted cards.

4. Since without a valid credential it is impossible to forge a proof (relying on the ABC technology) and thus to produce a matching conversation, an adversary can either authenticate using exactly one uncorrupted card (benign adversary) or using only corrupted cards. Assuming that an ideal revocation is provided, the protocol is authentic.

Being confidential and authentic, ***ICA*** is a secure ABC session protocol. □

Note that we proposed to use a root credential that is only issued to valid cards after a rigorous verification procedure. The presence of such a credential, which can be demonstrated using an empty proof, shows that a card is valid. However, since ABC proofs are rather expensive in terms of time [3, 12, 15], it is often desirable to omit as many selective disclosure proofs as possible in practice. In a slightly modified trust model, verifiers may rely on issuers to verify properly the root credential before they issue a new ABC. In this case verifiers do not need to request a separate validity proof. This is a decision that the given system manager and/or a particular verifier can decide upon. Needless to say, if the verification aims at a service of issuing new credentials on a particular card, card validity should be verified and issuance should be carried out in the same secure channel as the verification.

2.3 An ABC Channel with DH Key Exchange

Although ***ICA*** is an efficient protocol to build a secure channel for ABC proofs, in some applications it might be desirable to make an explicit authentic key establishment. Furthermore, the roles of a verifier and a card is very different in ***ICA***, while in future scenarios participants may be provers and verifiers simultaneously. We propose a protocol that addresses these issues.

To construct a new protocol, we will employ two techniques. First, we define a new type of public-key certificate based on ABCs. Second, in order for the two parties to set up a session key, they use authenticated Diffie–Hellman

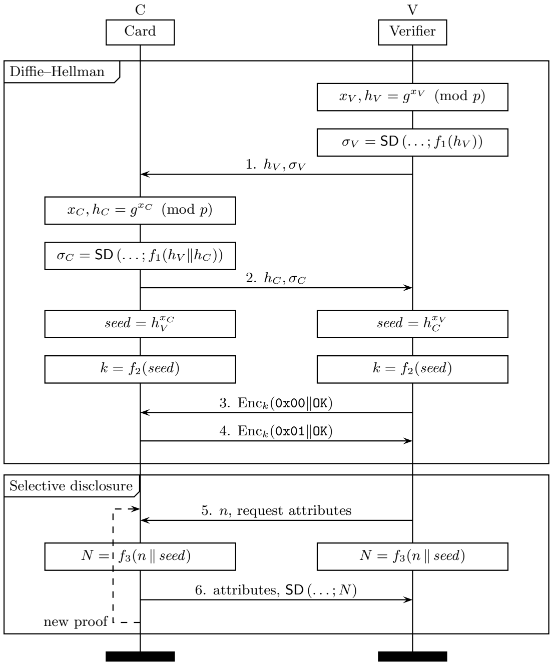
C
Card
V
Verifier
Diffie–Hellman
$x_V, h_V = g^{x_V} \pmod p$
$\sigma_V = \mathsf{SD}(\ldots; f_1(h_V))$
1. h_V, σ_V
$x_C, h_C = g^{x_C} \pmod p$
$\sigma_C = \mathsf{SD}(\ldots; f_1(h_V \| h_C))$
2. h_C, σ_C
$seed = h_V^{x_C}$
$seed = h_C^{x_V}$
$k = f_2(seed)$
$k = f_2(seed)$
3. $\mathrm{Enc}_k(\mathtt{0x00}\|\mathtt{OK})$
4. $\mathrm{Enc}_k(\mathtt{0x01}\|\mathtt{OK})$
Selective disclosure
5. n, request attributes
$N = f_3(n \,\|\, seed)$
$N = f_3(n \,\|\, seed)$
6. attributes, $\mathsf{SD}(\ldots; N)$
new proof

Figure 2: *ABCDH* protocol: Explicit ABC authentication and the Diffie–Hellman key exchange for selective disclosure.

(DH) key exchange. Both parties will authenticate using zero-knowledge proofs.

As mentioned above, we define an attribute-based public-key credential (ABPKC). A certificate authority issues ABPKCs that contain the verifier's identifier, public key and the access rights to particular attributes in ABCs that the verifier is authorised to request from smart-cards. As any other ABC, an ABPKC also has selective disclosure capacity. Therefore, a verifier can reveal its access rights adaptively depending on use cases. For instance, it is not necessary for a user to know that a service provider is eligible to request the "gender" attribute, when the current application needs information only about her age.

It is well known that the textbook Diffie–Hellman key-exchange protocol [10] is vulnerable to man-in-the-middle attacks. The reason for that is that there is nothing that binds the ephemeral public shares to the two parties who intend to establish a session key. While we do not identify cards and verifiers, we are still able to authenticate them and their messages. Each public share is signed by selective disclosure, that is, a non-interactive attribute-based zero-knowledge proof of knowledge; the ***ABCDH*** protocol is shown in Figure 2. Hereafter, we assume that a DH group is given (*e.g.*, $\mathbb{Z}_p^*$, a prime subgroup G of $\mathbb{Z}_p^*$ of order q, or points on an elliptic-curve over a finite field) and the participants know all system parameters and they verify the security of choices of the other party. Though we use multiplicative notation with a prime subgroup, the given protocol can easily be adapted to other groups.

As in Section 2.2, we assume some underlying secure cryptographic primitives, such as random oracles f_1, f_2, f_3 (implemented as a standard hash function), a symmetric-key authenticated encryption for the secure channel, and the ABC technology that supports selective disclosure of attributes.

The verifier generates its own share $x_V \in \mathbb{Z}_q$ and computes $h_V = g^{x_V} \pmod p$. Using its ABPKC, the verifier creates a selective disclosure proof $\sigma_V = \mathsf{SD}((id_V, \mathtt{Att\_Acc}); f_1(h_V))$ about its identifier id_V and its attribute access rights $\mathtt{Att\_Acc}$. The verifier uses $f_1(h_V)$ as the nonce to be signed by the non-interactive proof. Finally, it sends the proof σ_V and the DH public share h_V to the card.

Upon receiving the verifier's authentication and the public share, the card checks the proof and the values, and stores the access rights $\mathtt{Att\_Acc}$ and the identifier id_V. It generates its own private share $x_C \in \mathbb{Z}_q$ and computes $h_C = g^{x_C} \pmod p$. The card creates an empty proof of validity using $f_1(h_V \| h_C)$ as message to be signed by the NIZK: $\sigma_C = \mathsf{SD}(\emptyset; f_1(h_V \| h_C))$. Finally, the card sends its public share h_C and the proof σ_C to the verifier who can verify the proof.

According to the Diffie–Hellman key exchange, both parties can now calculate the *seed* $= g^{x_V x_C}$. They can derive their shared key $k = f_2(seed)$ and perform an explicit key confirmation. The verifier and the card carry out ABC selective disclosure proofs within the secure session protected by key k. Like in ***ICA***, nonces are transformed before they are signed by the NIZK proofs.

Theorem 2. *Assume that a Diffie–Hellmann group is set up and known to all participants, a secure ABC technology is given for selective disclosure with ideal revocation, and a symmetric-key authenticated encryption is given for the secure channel. Moreover, let f_1, f_2, f_3 be random oracles with all participants (including the adversary) having access to them. Then the* ***ABCDH*** *protocol is a secure ABC session protocol.*

Although the ***ABCDH*** protocol is not as efficient as the ***ICA*** protocol, it is worth discussing it for multiple reasons. First, it demonstrates that on an abstract level the verifier and the prover (the card) can be regarded in a symmetric manner. Both of them have an ABC to prove validity—potentially without identification—and to protect their privacy. This enables us to foresee applications that have not been considered yet. Some examples include machine-to-machine communication in the internet of things or ad-hoc communication between individuals who do not trust each other and thus wish to share as little information as possible in a given context. As devices become more powerful in terms of memory and computation, this protocol becomes efficient. Second, the use of ABCs as a new kind of public-key certificate provides more flexibility than the current technology and thus it extends the possible use of PKIs.

3. CONCLUSION

Selective disclosure and flexible credential issuance (*i.e.*, one based on already existing credentials) are important mechanisms of attribute-based credentials to provide security and privacy simultaneously. ABCs are building blocks of future privacy-friendly electronic identity systems. This paper shows how to build a secure channel between a verifier and a potentially anonymous smart-card carrying ABCs in the presence of a very powerful adversary.

First, we have described a security model to enable us to make security proofs with standard cryptographic primitives and assumptions. Second, we have shown two protocols that can be analysed in this framework. One of these protocols is more efficient, the other one can be generalised to new scenarios in which devices (a prover and a verifier) authenticate each other anonymously. Third, we have shown the security of these protocols relying on standard assumptions.

We assumed that proper revocation mechanisms exist that can handle abuses of ABCs. Although there exist cryptographic techniques for revocation, most of them are not efficient enough for smart-card implementations. Feasible and easily applicable privacy-preserving revocation techniques are crucial in the deployment of ABCs, but they are yet to be developed.

Authentication has been considered as a general notion. Rather than simply a proof of identity, authentication is a proof that certain predicates hold for an entity. When a secure channel is built on this notion of mutual authentication, participants can be convinced that the entities at the other end meet some requirements in terms of these predicates. Privacy-respecting applications will need security analyses in a similar model as the one shown in this paper. As we mentioned, ***ABCDH*** can be the first step towards many new such protocols.

Acknowledgement. We would like to thank the anonymous referees for their fruitful comments. Furthermore, we also want to thank Pim Vullers for the thorough proofreading and helpful comments and the whole IRMA team for several valuable discussions.

4. REFERENCES

[1] G. Alpár and B. Jacobs. Credential Design in Attribute-Based Identity Management. In R. Leenes, editor, *TILTing Perspectives*, 2013.

[2] M. Bellare and P. Rogaway. Entity authentication and key distribution. In *Advances in Cryptology—CRYPTO'93*, pages 232–249. Springer, 1994.

[3] P. Bichsel, J. Camenisch, T. Groß, and V. Shoup. Anonymous credentials on a standard Java Card. In *Computer and Communications Security – CCS 2009*, pages 600–610. ACM, November 2009.

[4] S. Brands. U-Prove technology overview. Technical report, Microsoft Corporation, March 2010.

[5] S. A. Brands. *Rethinking Public Key Infrastructures and Digital Certificates: Building in Privacy.* MIT Press, Cambridge, MA, USA, 2000.

[6] J. Camenisch, N. Casati, T. Gross, and V. Shoup. Credential authenticated identification and key exchange. In *Advances in Cryptology–CRYPTO 2010*, pages 255–276. Springer, 2010.

[7] J. Camenisch, I. Krontiris, A. Lehmann, G. Neven, C. Paquin, K. Rannenberg, and H. Zwingelberg. D2.1 Architecture for Attribute-based Credential Technologies. Technical report, ABC4Trust, 2011.

[8] J. Camenisch and A. Lysyanskaya. An Efficient System for Non-transferable Anonymous Credentials with Optional Anonymity Revocation. In B. Pfitzmann, editor, *Advances in Cryptology — EUROCRYPT 2001*, volume 2045 of *LNCS*, pages 93–118. Springer Berlin / Heidelberg, 2001.

[9] D. Chaum. Security without identification: transaction systems to make big brother obsolete. *Communications of the ACM*, 28:1030–1044, October 1985.

[10] W. Diffie and M. Hellman. New directions in cryptography. *Information Theory, IEEE Transactions on*, 22(6):644–654, 1976.

[11] J. Lapon, M. Kohlweiss, B. De Decker, and V. Naessens. Analysis of revocation strategies for anonymous Idemix credentials. In *Communications and Multimedia Security*, pages 3–17. Springer, 2011.

[12] W. Mostowski and P. Vullers. Efficient U-Prove implementation for anonymous credentials on smart cards. In G. Kesidis and H. Wang, editors, *Security and Privacy in Communication Networks – SecureComm 2011*, volume 96 of *LNICST*, pages 243–260. Springer-Verlag, 2011.

[13] A. Poller, U. Waldmann, S. Vowé, and S. Türpe. Electronic identity cards for user authentication—promise and practice. *IEEE Security & Privacy*, 10(1):46–54, 2012.

[14] Security Team, IBM Research. Specification of the Identity Mixer Cryptographic Library, version 2.3.4. Technical report, IBM Research, Zürich, Feb. 2012.

[15] P. Vullers and G. Alpár. Efficient Selective Disclosure on Smart Cards Using Idemix. In S. Fischer-Hübner, E. de Leeuw, and C. Mitchell, editors, *Policies and Research in Identity Management (IDMAN)*, IFIP AICT 396, pages 53–67. Springer, 2013.