Evaluation of Privacy-ABC Technologies - A Study on the Computational Efficiency

Fatbardh Veseli and Jetzabel Serna

Chair for Mobile Business and Multilateral Security, Goethe University Frankfurt
Theodor-W.-Adorno-Platz 4, 60323 Frankfurt am Main, Germany
fatbardh.veseli@m-chair.de, jetzabel.serna@m-chair.de

Abstract. Privacy-enhancing attribute-based credential (Privacy-ABC) technologies use different cryptographic methods to enhance the privacy of the users. This results in important practical differences between Privacy-ABC technologies, especially with regard to efficiency, which have not been studied in depth, but is necessary for assessing their suitability for deployment on devices with limited computational power and for highly dynamic scenarios. In this paper, we compare the computational efficiency of two prominent Privacy-ABC technologies, IBM's Idemix and Microsoft's U-Prove, covering all known Privacy-ABC features. The results show that presentation overall in general is more efficient with Idemix, whereas U-Prove is more efficient for the User side (proving) operations during the presentation, and overall when there are more attributes in a credential. For both technologies, we confirmed that inspectability, non-revocation proofs, and inequality predicates are costly operations. Interestingly, the study showed that equality predicates, the number of attributes in a credential, and attribute disclosure have a very small influence on the efficiency for both technologies. Finally, we discuss important trust considerations specific to Privacy-ABCs.

1 Introduction

Nowadays, electronic service providers continuously collect, store, integrate and analyse huge amounts of personal data. This data is commonly used to provide authentication, personalized services, targeted advertisements, and to develop innovative applications that address critical societal challenges in sectors like transportation, eHealth, etc. However, each piece of information is a digital footprint of our identity, which may introduce risks to our privacy. In this regard, regulation, such as the forthcoming European Data Protection Regulation [1], can be a useful instrument for protecting the privacy of individuals, but needs to be enforced with technical privacy protection mechanisms.

Privacy-enhancing Attribute-Based Credentials (Privacy-ABCs) are innovative technologies that both provide authentication and access control for the digital services, and enhance user's privacy. Furthermore, they may relieve the Service Providers from the liability with respect to the personal data about the users in cases of security breaches by reducing the amount of the collected

data to a minimum. The main concepts behind Privacy-ABC technologies have initially been introduced by David Chaum's anonymous credential systems in the 80s [2]. In particular, they enable pseudonymous access to services, minimal disclosure of attributes, and unlinkability of users' transactions. However, despite their potential and their apparent technical maturity, their adoption is still low [3].

In this paper, we focus on the technical aspects that might influence the adoption of Privacy-ABCs. In this respect, we identify efficiency as an important factor that can influence their viability of deployment in a wide range of scenarios. Especially when deployed for devices with limited resources (e.g., smart phones or smart cards) and with high mobility (e.g., vehicular on-board units), it is important to understand which technology performs better and which privacy features are more costly in terms of computational time. Therefore, we compare the computational efficiency of two prominent implementations of Privacy-ABC technologies, namely Microsoft's U-Prove [4] and IBM's Identity Mixer (Idemix) [5]. Additionally, we evaluate the cost advanced features on the efficiency, such as inspectability, non-revocation proofs, predicates, number of attributes, and number of disclosed attributes.

The rest of the paper is organized as follows, Section 2 introduces the main concepts of Privacy-ABC technologies. Section 3 presents the related work. Section 4 introduces the study methodology, whereas the main results are presented in Section 5. These results are then discussed in Section 6, where also important challenges and important trust considerations are identified. Finally, Section 7 concludes the paper.

2 Background

A general architecture of Privacy-ABC technologies consists of three main entities, namely a *User*, an *Issuer*, and a *Verifier*. The Issuer issues *credential(s)* to the *User*, which can later be used for authentication with the Verifier. A credential contains attributes about the User, e.g. name, date of birth, etc. Such an architecture may optionally also include an entity that takes care of revocation of credentials, and another entity that can revoke the anonymity of otherwise anonymous users. The main interactions between Privacy-ABCs system entities [6] can be represented by the different stages of its lifecycle, that is, *issuance, presentation, inspection*, and *revocation*.

- **Issuance** of a credential is an interactive protocol between a User and an Issuer. By issuing a credential to the User, the Issuer vouches for the correctness of the attribute values contained in the credential.
- **Presentation** is an interactive protocol in which the User proves the possession of certain credential(s) or claims about the attributes. A *presentation token* is a cryptographic proof derived from the (credential of the) User as an evidence of possessing certain credential(s), optionally disclosing some attribute to the Verifier.

– **Inspection** provides conditional anonymity. It enables a trusted entity (i.e., an Inspector) to revoke the otherwise anonymous transaction (presentation).
– **Revocation** ends the validity of the credentials whenever necessary, such in case of service misuse, credential compromise, or loss of credential storage medium (e.g. smart card).

At the core of Privacy-ABCs untraceability and unlinkability of credentials are the two most important privacy-related features. *Untraceability*[1] refers to the privacy feature which guarantees that the presentation of credential(s) cannot be linked to their issuance, whereas *unlinkability*[2]refers to the privacy feature that guarantees that a Verifier cannot link different presentations of a given user. Additionally, Privacy-ABCs also support the following features:

– **Carry-over of attributes**, which enables users to carry over some attribute(s) from an existing credential into a new one without disclosing it to the Issuer.
– **Key binding**, which enables binding a credential to a secret key, or having one or more credentials bound to the same secret (protecting against credential pooling);
– **Selective disclosure** of attributes during the presentation;
– **Predicates over attributes**, which allow logical operators, such as greater or smaller than, to be applied on attributes without disclosing them;
– **Pseudonyms** enable users to create (one or more) pseudonyms for a service;
– **Inspectability** is an accountability feature that enables revocation of a user's anonymity if certain pre-defined conditions have been met.

3 Related work

Efficiency of Privacy-ABC systems has been identified as an important challenge and previously discussed in a number of studies [7–11]. On a theoretical level, Baldimtsi and Lysyanskaya [11] as well as Camenisch and Groß [9] have specially addressed the importance of computational efficiency in resource-constrained devices, such as smart phones and smart cards.

Later, Camenisch and Lysyanskaya [10] proposed a signature scheme with more efficient protocols based on the Strong RSA assumption. Following this direction, Chase and Zaverucha [12] have proposed an approach for providing (a subset) features of Privacy-ABCs based on the use of message authentication codes (MACs) instead of public keys for better efficiency. However, their proposal has an important limitation since the Issuer and Verifier share the same secret key, making them not be suitable in scenarios, such as ad-hoc networks, where a regional road authority will act as an Issuer, and a number of road side units for traffic management will act as Verifiers.

A number of other efforts to achieve efficient implementations of Privacy-ABC technologies have emerged, especially focused on smart cards [7, 8, 10, 13].

[1] Also known in the literature as "Issuer-unlinkability".
[2] Also known in the literature as "Verifier-unlinkability".

For instance Bichsel et al. [7] reported the first practical implementation based on Idemix on a JCOP card. Mostowski and Vullers [14] optimized the efficiency U-Prove, and later Vullers and Alpar [15] optimized the efficiency for Idemix on a MULTOS card and presented a comparison of both approaches. However, their practical evaluation of Privacy-ABC technologies covered only the basic presentation of a single credential.

De la Piedra *et al.* [16] provided additional optimizations for Idemix on smart cards by implementing an efficient extended Pseudo-Random Number Generator (PRNG). Contrary to Vullers and Alpar [15], De la Piedra *et. al* presented efficiency results considering a more advanced setup which included a combination of credentials and the use of predicates. However, the authors did not cover advanced features such as inspection or revocation, and furthermore, their experiments only considered Idemix.

In summary, existing efforts have so far either focused on a single technology or covered only a limited subset of the Privacy-ABC features. We fill this gap by evaluating the computational efficiency of the two Privacy-ABC technologies covering all of the known features of Privacy-ABCs. This complements previous published work focused on storage and communication efficiency [17]. To our knowledge this is the first attempt to compare both technologies, under a common architecture and evaluation framework. This is especially important considering that the publicly available description of U-Prove or Idemix define efficiency only in theoretical terms and do not provide practical benchmarks of the actual efficiency [5, 18].

4 Methodology

In this work, we have adopted the Privacy-ABC technologies evaluation framework proposed by Veseli *et al.* [19]. This framework defines a rich set of criteria for benchmarking Privacy-ABC technologies based on their efficiency, functionality, and security assurance. With regard to efficiency, the framework distinguishes between *computational efficiency*, measured in time units; *communication efficiency*, measuring the sizes of the dynamically generated data; and *storage efficiency*, measuring the sizes of the static data in permanent storage. We focus on the former, covering all Privacy-ABC features.

Evaluated technologies. The core building block of Privacy-ABC technologies is the signature scheme. Therefore, we compare Microsoft's U-Prove based on Brands' signatures [20], and IBM's Idemix, which is based on Camenisch-Lysyanskaya's signatures [10]. As such, both technologies support selective disclosure of attributes, pseudonyms, and untracebility.[3] Other advanced features, such as non-revocation proofs, inspectability, or predicates are supported by additional building blocks, which are shared for both Idemix and U-Prove. For

[3] Unlike Idemix, U-Prove tokens are untraceable, but linkable between different presentations.

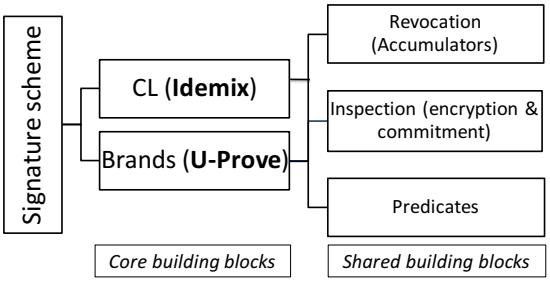

Fig. 1. Overview of the evaluated technologies

Table 1. Testbed for the experiment

Processor	1.8 GHz Intel Core i7
Number of processors	1
Number of Cores	2
L2 Cache (per Core)	256 KB
L3 Cache	4 MB
Running Memory	4 GB 1333 MHz DDR3
OS	Mac OS X

revocation, we have used the accumulator technology based on the Camenisch-Lysyanskaya [21], whereas inspectability is implemented using the verifiable encryption scheme introduced by Camenisch-Shoup [22]. Figure 1 shows an overview of the components evaluated in our study.

Contrary to the Privacy-ABC technology introduced by Persiano [23], both U-Prove and Idemix provide a similar level of technology readiness [24] and have been integrated under a common architecture [25], which has a reference implementation openly available on Github [26]. We performed the practical benchmarks using this implementation, enabling the same measurement instrument to test both technologies, providing a fair comparison.

Experimental setup. The experimental setup has been done using Java, the experiments have been executed on a computer with the configuration shown in Table 1. All experiments have been evaluated using a key length of 1024 bits, based on the RSA group[4], which is an important element in the evaluation (see more on this in Section 6). All the experiments reflect the average performance time from 50 runs of each operation.

Limitations. Our results are based on the openly available versions of U-Prove and Idemix. However, contrary to Idemix, U-Prove could be instantiated over

[4] It is worth to note that the U-Prove's implementation was instantiated over standard subgroup, alternatively it could also be based on elliptic curves. However, this was not available for the reference architecture we used.

elliptic curves, which can be more efficient. Furthermore, our architecture is designed to allow flexible changes in the features that are used, but it also represents an overhead, which could be avoided in scenarios where flexibility is not needed.

5 Results

This section provides a comparison of the computational efficiency between U-Prove and Idemix, and an evaluation of the computational cost of advanced Privacy-ABC features.

5.1 Comparison of the Efficiency of Privacy-ABC Technologies

The comparison between Idemix and U-Prove will follow the lifecycle of the credentials, covering both issuance and presentation of Privacy-ABCs. Issuance efficiency is important for cases that require periodic issuance of new credentals, whereas presentation efficiency is assumed to be important for most of the practical scenarios. A non-efficient presentation may (negatively) influence users' experience and consequently their perception on the technology, which we assume to play a crucial role on their acceptance by users [27].

Comparing Issuance Efficiency. Privacy-ABC technologies can support *simple* and *advanced* forms of issuance. In the case of simple issuance, the Issuer does require the User to present any existing credential or pseudonym. This can be, for instance, when this is the first Privacy-ABC credential that the User gets. Advanced issuance follow a presentation of User's existing credential or pseudonym, bind a credential to the secret key of an existing credential or pseudonym (*key binding*), or even *carry over attributes* from the existing credential into the new one, without the Issuer learning their value(s).

Figure 2 compares the computational efficiency of Idemix and U-prove for different types of issuance, where all issued credentials contain 5 attributes. It includes efficiency results for the following issuance types:

Simple issuance is the simplest and therefore the most efficient form of issuance. It does not require the user to present any proof with Privacy-ABC technologies. For this type of issuance, both Idemix and U-Prove have a similar efficiency, where the credential is issued in less than 300 ms, with Idemix being only slightly more efficient (about 20 ms).

Show Pseudonym shows the efficiency for an advanced issuance that requires the User to present an existing pseudonym, after which the issuance of the new credential follows. The cost of showing a pseudonym is reflected in about 70% for Idemix and 80% overhead for U-Prove relative to simple issuance. Compared with U-Prove, Idemix is more efficient for about 50 ms.

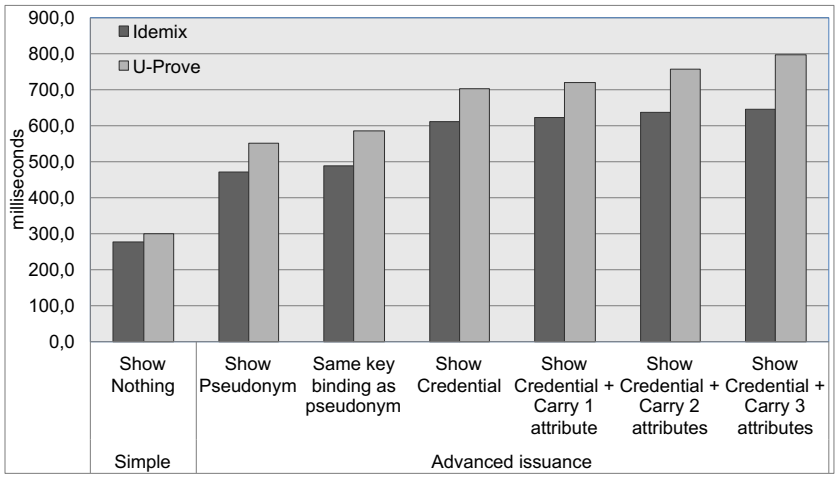

Fig. 2. Comparing the computational efficiency of Idemix and U-Prove for simple and advanced forms of issuance

Same key as pseudonym requires the new credentials to be bound to the same secret key as the pseudonym that the User presents. The effect of "key binding" has a small to small overhead for both Idemix (about 10 ms) and U-Prove (15 ms).

Show credential shows the time to get a credential issued when an existing credential is required (presented). The results show that, compared to simple issuance, this issuance has about 315 ms or 110% overhead for Idemix, and about 400 ms or 130% overhead for U-Prove. Idemix is in general more efficient than U-Prove for less than 100 ms.

Show credential + carry 1/2/3 attributes show the overhead of "carrying" 1, 2, and 3 attributes respectively for Idemix and U-Prove. For Idemix, each carried attribute represents a computational overhead of less than 15 ms, whereas for U-Prove this costs about 25 ms. Also here, Idemix is more efficient than U-Prove for 100-150 ms.

Comparing Presentation Efficiency. In Figure 3, we provide a comparison of the computational efficiency of presentation for Idemix and U-Prove for four different presentations. We have distinguished between two steps during the presentation phase: *proving* includes the cryptographic operations performed by the User in order to generate the proof (presentation token), and *verification*, which is the step performed at the Verifier side upon receiving the presentation token of the User. This may be important, for instance, in order to distinguish between the effort distribution between the User and the Verifier for the two technologies, and adapt their computational power accordingly in order to achieve a better efficiency. The figure shows the following four presentation cases:

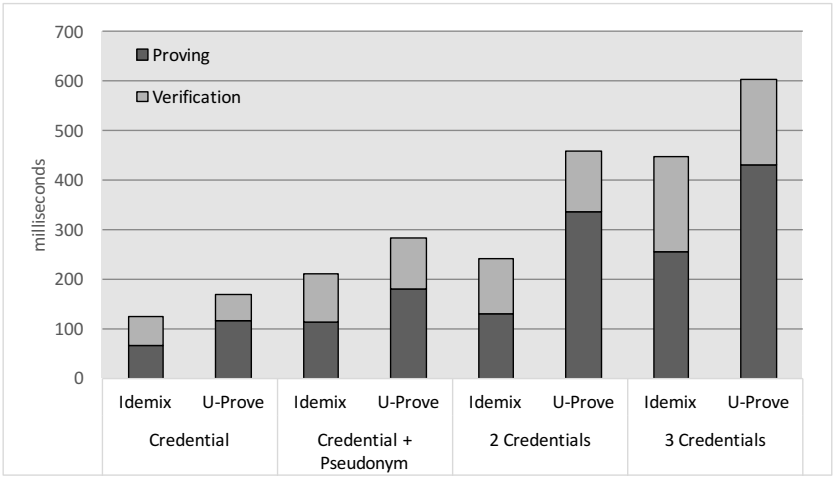

Fig. 3. Comparison of the computational efficiency of presentation of key-bound credentials between Idemix and U-Prove

Credential shows the efficiency of presenting a credential (zero knowledge) when no attributes are disclosed. The credential is key-bound, meaning that it underlies a secret key, and contains five different attributes. This type of presentation takes about 120 ms for Idemix and 180 ms for U-Prove, making Idemix slightly more efficient

Credential + Pseudonym shows the efficiency of presenting a credential and a pseudonym, both bound to the same secret key. This can be useful, for instance, when the Verifier wants to offer the possibility to the User to maintain a "reputation" under a certain pseudonym besides having a proof of a credential. Compared to "Credential", we can observe an overhead of showing a pseudonym, which is about 80 ms (about 60%) for Idemix, whereas for U-Prove it represents an overhead of about 110 ms (65%). Compared to showing a new credential, the overhead of showing a pseudonym is smaller for both technologies for 30 ms for Idemix and about 190 ms for U-Prove, which shows that presenting a pseudonym is more efficient with Idemix.

2/3 Credentials shows the overhead of additional credentials. Compared to "Credential", the presentation time grows linearly for each presented credential for both Idemix and U-Prove. Considering that Idemix is more efficient than U-Prove for about 75 ms per credential or about 35%, the same difference grows linearly with each new credential.

Effort distribution. Finally, we can observe that for both technologies, proving is more costly than verification. On average, proving takes about 55% of the total presentation time for Idemix, while for U-Prove it takes about 70%. This is another advantage for Idemix, since the goal is to make the computation at the User part more efficient.

Additional results. The *number of attributes* in a credential and *attribute disclosure* result in a small overhead on the presentation time for both technologies. For the former, U-Prove gets more efficient than Idemix as the number of attributes increases. Proving a credential with 5 more attributes showed better efficiency for U-Prove than for Idemix, where each new attribute cost about 4, respectively 6,5 additional ms. With regard to attribute disclosure, we noticed a small, but positive impact for both technologies, where the presentation time was about 2-5 ms more efficient for each disclosed attribute.

5.2 Evaluation of the Cost of Advanced Privacy-ABC Features

Besides the core features of Privacy-ABC technologies, such as unlinkability, selective disclosure, and pseudonyms, additional privacy features are also possible with extensions. These functionalities correspond to specific building blocks that need to be integrated with the respective Privacy-ABC technologies, most notably to support features such as *predicates, showing non-revocation, inspectability*, etc. The respective impact (overhead) of these features on the computational efficiency of presentation is shown in Figure 4 for Idemix.[5] The figure shows the computational efficiency for the following presentations (all of which require a credential of 5 attributes):

Credential is the basic benchmark, which simply requires presenting a credential, and serves as a reference for assessing the additional overhead of using other features.

Credential + Predicate: Date equality represents the presentation of a credential and checking that one of the attribute values (in this case, the date of birth) equals a given constant value, both of which are of type *Date*. We can observe from the results of this diagram that date equality proof represents little to no overhead on the efficiency of presentation.

Credential + Predicate: String equality is similar to the previous one, except that the compared attributes are of different data type, namely they are of type *String* (as compared to the previous one, which is *Date*). Also in this case we observe little to no overhead on the presentation time.

Credential + Revocation shows the overhead of presenting a revocable credential. In this case, the presentation requires also proving that the credential is not revoked. We can observe that the overhead of proving non-revocation accounts for about additional 160 ms or about 130% more time (compared to "Credential").

Credential + Predicate: Date "greater than" shows the efficiency of both presenting a credential and showing that one of the attributes (the date of birth) is greater than a certain constant value. This is especially useful for scenarios where checking the age of a person is necessary, e.g. checking that a

[5] The features used in this section utilize the same "extensions", making the computational overhead the same for both technologies. For simplicity, we show the impact on Idemix.

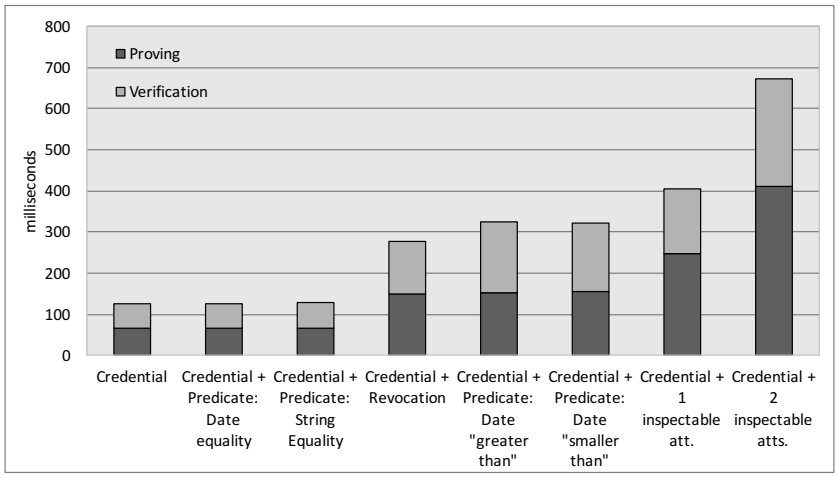

Fig. 4. Impact of the different features on the presentation efficiency (Idemix)

person is not older than a given age. We can observe that, compared to "Credential", showing that the date is greater than a given constant date costs about 200 additional ms or about 165% more (compared to "Credential").

Credential + Predicate: Date "smaller than" similar to the previous one, the efficiency of this predicate is comparable to checking "greater than", i.e., about 200 additional ms or 165% overhead. This is especially important for checking that a person is older than a given age, e.g. that a person is over 18 years old.

Credential + 1/2 Inspectable atts. shows the additional overhead of having one, respectively two attributes inspectable during the presentation. Typically, inspectability should suffice by having one attribute inspectable, in order to uncover the identity of the person in a given transaction (revoke anonymity), however there may be cases when more attributes are to be inspectable. From the figure, we can see that inspectability has the biggest overhead on the presentation among all of the presented features of Privacy-ABCs, and grows linearly by about 275 ms or by 210% for each inspectable attribute.

6 Discussion

This section discusses important implications of the results, and identifies open research challenges and specific trust issues for Privacy-ABC technologies.

6.1 Implications

Results show that both technologies (U-Prove and Idemix) present similar computational efficiency for simple issuance (i.e., less than 300 ms), Idemix outper-

forming U-Prove by 20 ms. However, for advanced issuance (e.g., carry over of attributes) differences could be of 150 ms, again Idemix being more efficient. Although, in many scenarios issuance efficiency will not play a major role, it may be relevant to those scenarios where users frequently need to get credentials issued interactively. For instance, in vehicular networks, the nodes are expected to send messages every 300 ms and have a communication range of 1000 m, making a time difference of 150 ms (e.g. the overhead of advanced issuance) an important decision factor for deciding the type of issuance. Nevertheless, in cases where issuance of credentials is assumed to be done off-line, the efficiency of both technologies can be considered acceptable in the aforementioned scenario.

In the following, we discuss some of the main implications related to the different features of the presentation.

Inspectability is the most computationally expensive feature that linearly grows with each new attribute made inspectable. The reason for this is the use of cryptographic *commitments* of the given attribute and *verifiable encryption* of that commitment with the public key of the Inspector. The Verifier is then able to check that the encrypted value corresponds to the value indicated and that is encrypted with the right key. However, one can assume that this process requires additionally administrative controls and therefore its efficiency is not critical in most scenarios.

Revocability is an important feature, but it is as costly as the presentation of a credential. Our revocation technology is based on the *accumulator* scheme of Camenisch-Lysyanskaya [21]. The overhead for the User comes from the fact that each revocable credential needs to be proven that it is part of the accumulator, making it not suitable for highly dynamic scenarios. A recommendation would be that, whenever more credentials are needed in a presentation, only one should be checked for revocation, and have the other credentials as non-revocable. In this way, the cost of proving non-revocation for the other credentials is avoided.

Predicates are costly when non-equality has to be shown, e.g. showing that an attribute value is greater or smaller than another one without revealing the attribute value. However, equality predicates seem to be very efficient and add very little overhead on the presentation, but they should be carefully used so that privacy is still preserved.

The number of attributes in a credential, as well as the **number of disclosed attributes** have shown small overhead on the efficiency of presentation for both, but slightly bigger overhead for Idemix, making U-Prove favourable when a credential has more attributes. Besides this, another implication for both technologies would be that a more efficient system design could result from decreasing the number of credentials whilst increasing the number of attributes in a credential whenever suitable, and disclosing only a desired subset of those attributes.

6.2 Open Challenges

Despite their practical differences, both Privacy-ABC technologies face some open challenges, calling for additional research efforts related to the efficiency, especially with regard to their deployability in different platforms, effective revocation, and the impact of the security level on the efficiency.

Deployment Platforms . The current results are performed on a personal computer with average computational power. Deploying them on other platforms, such as smart cards, mobile devices, or in the cloud comes with specific challenges. *Smart cards* require native support the cryptographic operations, and optimisations to make them practically efficient. For *mobile devices*, there are more possibilities, but having a cross-platform solution is challenging. One way is to use Javascript or have native browser support for digital signature schemes, which are still considered challenging [28]. A *cloud* solution would ease the availability in different user devices, but creates new privacy risks, since the cloud service provider would need to be trusted. Alternatively, one should check the feasibility of integrating technologies, such as proxy re-encryption schemes [29], where the cloud service provider can act on behalf of the User without seeing the attributes in clear.

Revocation and non-revocation proof. There is currently a compromise between efficiency and effective revocation. Unlike in the X.509 case, with Privacy-ABCs the User should not disclose a credential identifier to check for revocation. Instead, the user must show non-revocation in zero knowledge, but this is still a costly operation for the User. This requires the User to be on-line, which is an additional limitation (makes the technology not usable on "off-line" scenarios, e.g. smart cards). One way to optimise this process could be by shifting part of the proving effort from the User to the Verifier, or to extend the periods between different revocation checks, e.g. daily or weekly, depending on the scenario.

Key length and efficiency. The results in the paper reflect the 1024 bit key size for both, which seems to provide comparable level of security. According to the ECRYPT report on key sizes [30] this corresponds to the symmetric key size of around 72 bits, providing "short-term protection against medium organizations, medium-term protection against small organizations". According to the same report, an RSA cryptographic key size of 2048 bits would provide a security level corresponding to a symmetric key of around 105 bits, which is between a "legacy standard level" and one that offers a "medium-term protection" (about 10 years). Based on our experiments, the computational efficiency drops on average by a factor of four with the doubling of the key size. Therefore, an additional challenge remains to provide higher level of security assurance with smaller impact on the efficiency.

6.3 Trust considerations for Privacy-ABC technologies

One of the benefits of Privacy-ABC technologies is the fact that the User need to reveal minimal amount of information to Service Providers. This assumes less trust in the service providers for proper handling of user's personal information in general. However, while the technology itself enables better privacy, one has to ensure that the entities use the technology in a trustworthy manner. In this regard, three important considerations need to be made, which are briefly discussed in the following section.

Trust in the Verifier. Privacy-ABC technologies *enable* minimal disclosure of attributes, so that Service Providers (Verifiers) only require disclosure of the attributes that are necessary for authentication in a given scenario. This is defined in a so-called presentation policy. However, the technology itself cannot define by default what a minimal set of attributes for a given scenario is.

Hence, a Verifier can define "unfair" malicious policy, asking the User to disclose excessive amount of attributes [31]. In such a case, even though Privacy-ABC technologies are used, this does not imply a guarantee that a system using these technologies will always respect the supported privacy features. Therefore, one either has to trust that the Verifier will define proper presentation policies, or implement some mechanism in the infrastructure to protect from such a misuse potential, such as requiring certification of presentation policies by an external, trusted entity, or providing domain-specific or case-specific "standard" presentation policies for typical use-cases [31].

Trust in the Inspector. An Inspector is a dedicated entity in a Privacy-ABC supported infrastructure, which enables the revocation of anonymity in special cases. While this is done to provide conditional accountability for all the cases that could "go wrong", e.g. when a user violates the code of conduct for a given service, or when there is a threat to the lives or properties by anonymous users, this feature is also controversial, as requires trust on the proper conduct by the Inspector. We discussed earlier about the implications of the inpsectability on the efficiency of presentation, but in practice, a system should also limit the potential for authority misuse by a malicious Inspector, who could, in the worst case, revoke the anonymity of any user without a proper ground to do so.

Such a limitation could in practice be achieved by either organisational processes or technical solutions, or by a combination of them. An organisational solution could be to define a body that needs to decide on when inspection is needed, thereby requiring an approval by a committee or a group of people within an organisation. A technical solution would be to split the "inspection" key into several parts, and requiring at least 2 of the members of such a committee to come together to join their key parts in order to be able to perform inspection.

Trust implications in the Cloud. Efficiency could be improved if the User part of the computation could be outsourced to the cloud. This way, also mobility could be improved, since the user could access his credentials from any

device. However, as mentioned earlier, this implies additional trust is needed on the cloud service provider in proper handling of user credentials. A malicious cloud service provider could then spy on the user by tracking activities, or even worse, impersonate the User without his or her consent. While there are technical potentials for limiting such cases, including the use of special cryptographic tools such as proxy re-encryption schemes [29] or similar, there is a compromise between the level of risk the users gets exposed to and the convenience of the mobility provided by the cloud.

7 Conclusion

This paper presented an evaluation of two major Privacy-ABC technologies with regard to efficiency following the evaluation framework introduced by [19]. Our results have shown that U-Prove is more efficient than Idemix for the User operation (proving) and in general when a credential has more attributes. However, Idemix is more efficient in the rest of the cases, especially when advanced presentation features are used. Regardless of the efficiency, Idemix also provides unlinkability, which makes it more favourable in scenarios with stronger privacy requirements.

For both technologies, the efficiency drops linearly with the increased number of credentials being presented, and whenever inspectability, non-revocation proofs and inequality predicates are used. However, the number of disclosed attributes, number of attributes in a credential, and inequality predicates are relatively efficient. This knowledge is especially useful for system architects who need to understand the trade-off between using different features and the impact on the performance, e.g. defining some credentials as non-revocable in order to reduce the overhead on the presentation, whenever suitable. Finally, we identity important trust assumptions for Privacy-ABC systems, in particular regarding the trustworthiness of Verifiers and Inspectors, and trust implications for cloud-based deployments.

References

1. EU Commission. Proposal for a Rregulation of the European Parliament and of the Council - General Data Protection Regulation. Available online at `http://ec.europa.eu/justice/data-protection/document/review2012/com_2012_11_en.pdf`, 2012.
2. David Chaum. Security without identification: Transaction systems to make big brother obsolete. *Commun. ACM*, 28(10):1030–1044, October 1985.
3. Zinaida Benenson, Anna Girard, and Ioannis Krontiris. User acceptance factors for anonymous credentials: An empirical investigation. In *Proc. of the Workshop on the Economics of Information Security (WEIS)*, 2015.
4. Microsoft Research. U-Prove. `http://research.microsoft.com/en-us/projects/u-prove/`, 2013.
5. P. Bichsel, C. Binding, J. Camenisch, T. Gross, T. Heydt-Benjamin, D. Sommer, and G. Zaverucha. Cryptographic Protocols of the Identity Mixer Library. Technical Report RZ 3730 (99740), IBM Research GmbH, 2008.

6. Jan Camenisch, Maria Dubovitskaya, Anja Lehmann, Gregoy Neven, Christian Paquin, and Franz-Stefan Preiss. Concepts and Languages for Privacy-Preserving Attribute-Based Authentication. In *IDMAN*, volume 396, pages 34–52. Springer, 2013.
7. Patrik Bichsel, Jan Camenisch, Thomas Groß, and Victor Shoup. Anonymous credentials on a standard java card. In *Proceedings of the 16th ACM Conference on Computer and Communications Security*, CCS '09, pages 600–610, New York, NY, USA, 2009. ACM.
8. M. Sterckx, B. Gierlichs, B. Preneel, and I. Verbauwhede. Efficient implementation of anonymous credentials on java card smart cards. In *Information Forensics and Security, 2009. WIFS 2009. First IEEE International Workshop on*, pages 106–110, Dec 2009.
9. Jan Camenisch and Thomas Groß. Efficient attributes for anonymous credentials (extended version). *IACR Cryptology ePrint Archive*, 2010:496, 2010.
10. J. Camenisch and A. Lysyanskaya. A Signature Scheme with Efficient Protocols. In *Proceedings of the 3rd International Conference on Security in Communication Networks*, pages 268–289. Springer, 2003.
11. Foteini Baldimtsi and Anna Lysyanskaya. Anonymous credentials light. In *Proceedings of the 2013 ACM SIGSAC Conference on Computer & Communications Security*, CCS '13, pages 1087–1098, New York, NY, USA, 2013. ACM.
12. Melissa Chase and Greg Zaverucha. MAC Schemes with Efficient Protocols and Keyed-Verification Anonymous Credentials, 2013.
13. Lejla Batina, Jaap-Henk Hoepman, Bart Jacobs, Wojciech Mostowski, and Pim Vullers. Developing efficient blinded attribute certificates on smart cards via pairings. In Dieter Gollmann, Jean-Louis Lanet, and Julien Iguchi-Cartigny, editors, *Smart Card Research and Advanced Application, 9th IFIP WG 8.8/11.2 International Conference, CARDIS 2010, Passau, Germany, April 14-16, 2010. Proceedings*, volume 6035 of *Lecture Notes in Computer Science*, pages 209–222. Springer, 2010.
14. Wojciech Mostowski and Pim Vullers. Efficient u-prove implementation for anonymous credentials on smart cards. In Muttukrishnan Rajarajan, Fred Piper, Haining Wang, and George Kesidis, editors, *Security and Privacy in Communication Networks - 7th International ICST Conference, SecureComm 2011, London, UK, September 7-9, 2011, Revised Selected Papers*, volume 96 of *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, pages 243–260. Springer, 2011.
15. Pim Vullers and Gergely Alpar. Efficient Selective Disclosure on Smart Cards Using Idemix. In *IDMAN*, volume 396, pages 53–67. Springer, 2013.
16. Antonio de la Piedra, Jaap-Henk Hoepman, and Pim Vullers. Towards a full-featured implementation of attribute based credentials on smart cards. In Dimitris Gritzalis, Aggelos Kiayias, and Ioannis Askoxylakis, editors, *Cryptology and Network Security*, volume 8813 of *Lecture Notes in Computer Science*, pages 270–289. Springer International Publishing, 2014.
17. Fatbardh Veseli and Jetzabel Serna-Olvera. Benchmarking Privacy-ABC technologies - an evaluation of storage and communication efficiency. In *2015 IEEE World Congress on Services, SERVICES 2015, New York City, NY, USA, June 27 - July 2, 2015*, pages 198–205, 2015.
18. Christian Paquin and Greg Zaverucha. U-Prove Cryptographic Specification v1.1 (Revision 2). Technical report, Microsoft Corporation, 2013.

19. Fatbardh Veseli, Tsvetoslava Vateva-Gurova, Ioannis Krontiris, Kai Rannenberg, and Neeraj Suri. Towards a Framework for Benchmarking Privacy-ABC Technologies. In Nora Cuppens-Boulahia, Frédéric Cuppens, Sushil Jajodia, Anas Abou El Kalam, and Thierry Sans, editors, *ICT Systems Security and Privacy Protection*, volume 428 of *IFIP Advances in Information and Communication Technology*, pages 197–204. Springer Berlin Heidelberg, 2014.
20. Stefan A. Brands. *Rethinking Public Key Infrastructures and Digital Certificates: Building in Privacy*. MIT Press, Cambridge, MA, USA, 2000.
21. Jan Camenisch and Anna Lysyanskaya. Dynamic accumulators and application to efficient revocation of anonymous credentials. In Moti Yung, editor, *Advances in Cryptology, CRYPTO 2002*, volume 2442 of *Lecture Notes in Computer Science*, pages 61–76. Springer Berlin Heidelberg, 2002.
22. Jan Camenisch and Victor Shoup. Practical verifiable encryption and decryption of discrete logarithms. In Dan Boneh, editor, *Advances in Cryptology - CRYPTO 2003*, volume 2729 of *Lecture Notes in Computer Science*, pages 126–144. Springer Berlin Heidelberg, 2003.
23. Giuseppe Persiano and Ivan Visconti. *Financial Cryptography: 8th International Conference, FC 2004, Key West, FL, USA, February 9-12, 2004. Revised Papers*, chapter An Efficient and Usable Multi-show Non-transferable Anonymous Credential System, pages 196–211. Springer Berlin Heidelberg, Berlin, Heidelberg, 2004.
24. European Commission. Horizon 2020 work programme 2014-2015 - g. technology readiness levels (trl). `https://ec.europa.eu/research/participants/data/ref/h2020/wp/2014_2015/annexes/h2020-wp1415-annex-g-trl_en.pdf`, 2014.
25. Patrick Bichsel, Jan Camenisch, Maria Dubovitskaya, Robert Enderlein, Stefan Krenn, Ioannis Krontiris, Anja Lehmann, Gregory Neven, Janus Dam Nielsen, Christian Paquin, Franz-Stefan Preiss, Kai Rannenberg, Ahmad Sabouri, and Michael Stausholm. D2.2 Architecture for Attribute-based Credential Technologies - Final Version. *ABC4TRUST* project deliverable, 2014. Available online at `https://abc4trust.eu/index.php/pub`.
26. ABC4Trust pilots. Abc4trust pilots. `https://abc4trust.eu/index.php/home/pilots/`. Last accessed on 14.12.2015.
27. Fred D Davis. User acceptance of information technology: system characteristics, user perceptions and behavioral impacts. *International Journal of ManMachine Studies*, 38:475–487, 1993.
28. Jonas Lindstrom Jensen. D4.4 smartphone feasibility analysis. `https://abc4trust.eu/download/Deliverable_D4.4.pdf`, 2014.
29. Matt Blaze, Gerrit Bleumer, and Martin Strauss. Divertible protocols and atomic proxy cryptography. In Kaisa Nyberg, editor, *Advances in Cryptology - EUROCRYPT'98*, volume 1403 of *Lecture Notes in Computer Science*, pages 127–144. Springer Berlin Heidelberg, 1998.
30. Nigel Smart *ed.* D.SPA.20 ECRYPT II yearly report on algorithms and keysizes (2011-2012). Technical report, European Network of Excellence in Cryptology II, September 2012.
31. Fatbardh Veseli, Dieter M. Sommer, Jan Schallaboeck, and Krontiris Ioannis. D8.12 Architecture for Standardization V2. *ABC4TRUST* project deliverable, 2014. Available online at `https://abc4trust.eu/index.php/pub`.