I2PA, U-prove, and Idemix: An Evaluation of Memory Usage and Computing Time Efficiency in an IoT Context

Ibou Sene[1,2](✉), Abdoul Aziz Ciss[1], and Oumar Niang[1]

[1] Laboratoire de Traitement de l'Information et des Systèmes Intelligents (LTISI),
Ecole Polytechnique de Thiès, P.O. Box A10, Thiès, Senegal
senei@ept.sn

[2] Ecole Doctorale Développement Durable et Société (ED2DS),
Université de Thiès, P.O. Box 967, Thiès, Senegal
{aaciss,oniang}@ept.sn

Abstract. The Internet of Things (IoT), in spite of its innumerable advantages, brings many challenges namely issues about users' privacy preservation and constraints about lightweight cryptography. Lightweight cryptography is of capital importance since IoT devices are qualified to be resource-constrained. To address these challenges, several Attribute-Based Credentials (ABC) schemes have been designed including I2PA, U-prove, and Idemix. Even though these schemes have very strong cryptographic bases, their performance in resource-constrained devices is a question that deserves special attention. Therefor, this paper aims to conduct a performance evaluation of these schemes on issuance and verification protocols regarding memory usage and computing time. Recorded results show that both I2PA and U-prove present very interesting results regarding memory usage and computing time while Idemix presents very low performance with regard to computing time compared to I2PA and U-prove.

Keywords: ABC · Anonymity · Credential · IoT · Performances · Privacy · Lightweight cryptography

1 Introduction

Out of several emerging technologies and concepts, the Internet of Things is a new paradigm that brings both challenges and opportunities [1]. According to Ashton Kevin, to whom we owe the term "Internet of Things", the IoT has the potential to change the world, as did the Internet, maybe even more [2]. The Internet of Things represents a vision in which the Internet extends into the real world embracing everyday objects [3]. However, as mentioned above, it brings many challenges including issues about users' privacy preservation and constraints about lightweight cryptography [4]. We are among those who

R. Zitouni et al. (Eds.): AFRICOMM 2019, LNICST 311, pp. 140–153, 2020.
https://doi.org/10.1007/978-3-030-41593-8_10

think that the protection of privacy is a fundamental right and its loss would lead to the restriction of freedom [5]. Lightweight cryptography is a strong constraint because IoT devices are qualified to be resource-constrained. Indeed, these devices have three major constraints namely low energy autonomy, very limited storage capacity and very low computing power [4]. From there, were designed several schemes and the most promising include I2PA [4], Idemix [6], and U-prove [7]. These schemes are based on recognized robust cryptosystems. However, the question of their applicability in an IoT context, therefore in resource-constrained devices, is of capital importance. Roughly results recorded in [4] on issuance and verification of credentials made up of 10 attributes show that I2PA and U-prove are more efficient than Idemix regarding computing time efficiency. However, what about memory usage and computing time efficiency on different number of attributes? In this paper, we provide a deeper evaluation by regarding memory usage and computing time while issuing and verifying credentials made up of 1, 5, and 10 attributes respectively.

The rest of this paper is organized as follows. Section 2 is related to background review while related works are presented in Sect. 3. Section 4 describes experimental set-up whereas recorded results and discussions are depicted in Sect. 5. This paper is ended by a conclusion and perspectives in Sect. 6.

2 Background Review

We now recall few notions about Attribute-Based Credentials (ABC), Elliptic Curves Cryptography (ECC), Binary Scalar Multiplication (BSM), and Extended Homogeneous Coordinates (EHC). We refer readers to [4,8–11] for more details about discussed concepts in this section.

2.1 Attribute-Based Credential

Attribute-Based Credentials are mechanisms of authentication that allow to flexibly and selectively authenticate different attributes about an entity without revealing additional information about that entity. As a result, they do not necessarily identify the user, as they only provide authentic assertions about the user [4,8,9]. They are building blocks that aim at protecting users' privacy preservation.

2.2 Elliptic Curve Cryptography

Elliptic Curve Cryptography (ECC) was presented independently by Koblitz [12] and Miller [13] in the 1980s. Their structure of group and performance in computing time they offer make them a new direction in cryptography [4]. They offer good level of security with smaller key size. They are also less Central Processing Unit (CPU) intensive so they are ideal for resource-constrained devices.

2.3 Binary Scalar Multiplication

The fundamental operation of ECC is point scalar computation (also known as scalar multiplication) of the form [4]:

$$Q = k.P = \underbrace{P + P + \ldots + P}_{\text{k times}}$$

Security in ECC is based on Elliptic Curve Discrete Logarithm Problem (ECDLP) [4,10] that can be summarized as follows. Given an elliptic curve E defined over a finite field $\mathbb{F}_p$. Let $P, Q \in E(\mathbb{F}_p)$, find $k \in \mathbb{F}_q$, if it exists, such that $Q = k.P$ (q denotes the order of P). Scalar multiplication can be performed efficiently when tackling small numbers. However, when numbers hold in many bits (160 for instance), this might take lot of time. Several methods have been designed so far to speed up these operations including the double-and-add algorithm also known as binary algorithm. This algorithm is a very elegant technique to perform multiplication of big numbers. Two versions of this algorithm exist that either scan the scalar in a left to right or right to left direction [14]. Let k be an integer such that $k_{(10)} = (k_n k_{n-1} \ldots k_1 k_0)_{(2)}$, where $k_i \in \{0, 1\}, k_n = 1$ *and* $n \geq 1$. The left to right version is described in "Algorithm 1".

Algorithm 1. Left to right double-and-add

```
Input: $P \in E(\mathbb{F}_p), k \in \mathbb{F}_q$
Result: $k.P \in E(\mathbb{F}_p)$
R ←P
for $i \leftarrow n-1$ to 0 do
    R ←2.R
    if $k_i = 1$ then
        R ←R+P
    end
end
return $R$
```

The "Algorithm 1" is simple, efficient and has an average complexity of $nD + \frac{n}{2}A$ (D and A denote respectively the number of double and add operations). Implementations of I2PA and U-prove are based on this technique seeing its simplicity, its low complexity, and its easy implementation in place of other methods like Non-Adjacent Form (NAF) also known as Signed Binary Representation (SBR) which presents a more interesting complexity $(nD + \frac{n}{3}A)$ but requires a supplementary treatment on the representation of the scalar. Let us consider a device with a processor clocked at 1 GHz and $k = 2^{40}$. Computing $k.P$ with decimal representation of k would require around 19 min while with binary representation, this would take less than 1 ms. We remind that the number of bits required to represent a positive integer n in radix 2 is at most equal to $ceil(log_2(n)) + 1$, where $ceil(x)$ denotes the rounds of x up to the nearest integer. These results show how relevant it is to use this technique instead of decimal representation.

2.4 Extended Homogeneous Coordinates

According to our experimental parameters (see Sect. 4.3), Josefsson et al. [11] recommended to use extended homogeneous coordinates (EHCs). In the EHCs representation, (x, y) is represented as $(X : Y : Z : T)$ where $x = \frac{X}{Z}$, $y = \frac{Y}{Z}$ and $xy = \frac{T}{Z}$. The neutral point $(0, 1)$ is equivalent to $(0 : Z : Z : 0)$ for any nonzero Z. Coordinates $(X : Y : Z : T)$ and $(\lambda X : \lambda Y : \lambda Z : \lambda T)$ are equivalent for any nonzero λ. EHMs avoid inversion operations and, as a result, improve computing time efficiency. We refer readers to [11] for more details. Details of add and double formulas are presented respectively in "Algorithm 2" and "Algorithm 3".

Algorithm 2. Add formula

```
Input: $P_1, P_2 \in E(\mathbb{F}_p)$
Result: $P_1 + P_2 \in E(\mathbb{F}_p)$
A $\leftarrow (Y_1 - X_1)(Y_2 - X_2)$
B $\leftarrow (Y_1 + X_1)(Y_2 + X_2)$
C $\leftarrow 2dT_1T_2$
D $\leftarrow 2Z_1Z_2$
E $\leftarrow B - A$
F $\leftarrow D - C$
G $\leftarrow D + C$
H $\leftarrow B + A$
$(X, Y, Z, T) \leftarrow (EF, GH, FG, EH)$
return $(X : Y : Z : T)$
```

Algorithm 3. Double formula

```
Input: $P \in E(\mathbb{F}_p)$
Result: $2.P \in E(\mathbb{F}_p)$
A $\leftarrow X^2$
B $\leftarrow Y^2$
C $\leftarrow 2Z^2$
D $\leftarrow (X + Y)^2$
H $\leftarrow B + A$
E $\leftarrow H - D$
G $\leftarrow A - B$
F $\leftarrow C + G$
$(X', Y', Z', T') \leftarrow (EF, GH, FG, EH)$
return $(X' : Y' : Z' : T')$
```

3 Related Works

The Internet of Things brings both challenges and opportunities [1]. Indeed, in an IoT context, performance, privacy preservation, and lightweight cryptography are key aspects that must be taken into account with special attention. To the best of our knowledge, the best way of protecting users' privacy preservation remains using Attributes-Based Credentials (ABC) also known as Privacy-ABC. Many ABC schemes have been designed so far including I2PA, U-prove, and Idemix. However, few works evaluate the efficiency of these schemes in an IoT context. On a theoretical level, authors of [15,16] have addressed the importance of computational efficiency in resource-constrained devices. Veseli et al. [17] have evaluated the computational efficiency of Idemix and U-prove. Their results shown that U-prove is more efficient than Idemix for the User operation (proving) and in general when a credential has more attributes. They have also stated that Idemix is more efficient in the rest of the cases, especially when advanced presentation features are used. Their simulation uses a computer with a processor of 1.8 GHz Intel Core i7 and both schemes are instantiated using the RSA cryptosystem. Veseli et al. [18] have addressed storage and communication efficiency of Idemix and U-prove. Their results suggest that for storage, Idemix is more efficient than U-prove, since a single credential provides multiple-show unlinkability. They have also pointed out that, in terms of communication efficiency, Idemix is more efficient for issuance, whereas U-prove is more efficient for presentation of credentials. They have developed a number of experiments in Java, which have been executed on a computer with a processor of 1.8 GHz Intel Core i7 and schemes are based on the RSA cryptosystem. Vullers et al. [19] have presented an efficient selective disclosure on smart cards using Idemix (using the MULTOS platform). Their implementation is based on a 1024 bits security level. They asserted that Idemix's selective disclosure can be efficiently implemented on a smart card. Mostowski et al. [20] provided an efficient U-prove implementation for Anonymous Credentials on smart cards (Using the MULTOS platform). Their implementation aims at making the smart card independent of any other resources, either computational or storage. Their performance results strongly support their idea to use a stand-alone U-prove smart card rather than the Microsoft device-protection approach, which seems to overlook the current capabilities of smart cards. SENE et al. [4] have conducted a comparison of I2PA, U-prove, and Idemix on issuance and verification regarding computing time for credentials made up of 10 attributes. They have instantiated U-prove using ECC and their results have shown that U-prove presented more interesting results than Idemix regarding computing time on issuance and verification protocols. I2PA and U-prove present nearby performance even though I2PA's results are more interesting. Although these works have presented interesting results, most of them have focused on the efficiency of a particular implementation of a particular technology, and on a particular platform. Some of them were interested in many schemes or many aspects of privacy preservation but used computer which does not give any idea on low-resource devices efficiency. To the best of our knowledge, this is the first contribution that evaluates efficiency of I2PA, U-prove and Idemix in an IoT context regarding computing time and memory usage. Furthermore, as far as we know, it is also the first one that considers ECC-based U-prove instantiation in low-resource devices.

4 Experimental Setup

This section describes both hardware and software setup. It also describes curve and parameters used to perform this evaluation.

4.1 Hardware Setup

The hardware setup consists of a smartphone and a Raspberry Pi. The Raspberry Pi ("Fig. 1") is used to deploy both issuer and verifier. We describe some of its characteristics below:

- Model Pi 3 B+
- 1 Go of SDRAM LPDDR2
- A 64-bit quad core processor clocked at 1.4 GHz
- Raspbian operating system

- Dual Band 2.4 GHz and 5GHz IEEE 802.11. b/g/n/AC Wireless LAN
- Enhanced Ethernet performance over USB 2.0 (maximum throughput of 300 Mbps)

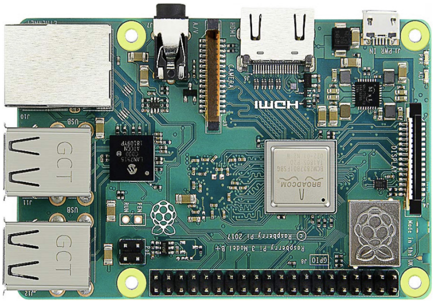
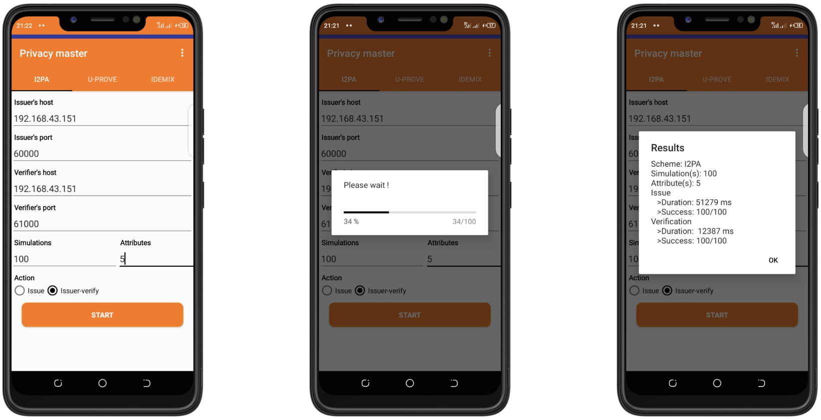


Fig. 1. Hardware environment

The smartphone ("Fig. 2") acts as a user. Some of its characteristics are depicted below:

- Model TECNO SPARK KB7j
- RAM 2 GB
- ROM 16 GB
- CPU 2.0 GH*4
- Battery 3500 mAh
- Memory 16 GB



Issuer's host
192.168.43.151
Issuer's port
60000
Verifier's host
192.168.43.151
Verifier's port
61000
Simulations 100
Attributes 5
Action
○ Issue ● Issuer-verify

Please wait !
34 % 34/100

Results
Scheme: I2PA
Simulation(s): 100
Attribute(s): 5
Issue
>Duration: 51279 ms
>Success: 100/100
Verification
>Duration: 12387 ms
>Success: 100/100
OK

(a) Initial view (b) Processing view (c) Result view

Fig. 2. Android application's screenshots.

4.2 Software Setup

The software environment is made up of three major components that are issuer, verifier, and user (“Fig. 3”). Issuer and verifier are Java Socket while user is an Android application. We are running both the issuer and the verifier on the same device (the Raspberry Pi) while the user is running on a smartphone. The “Fig. 3” is an overview of software components.

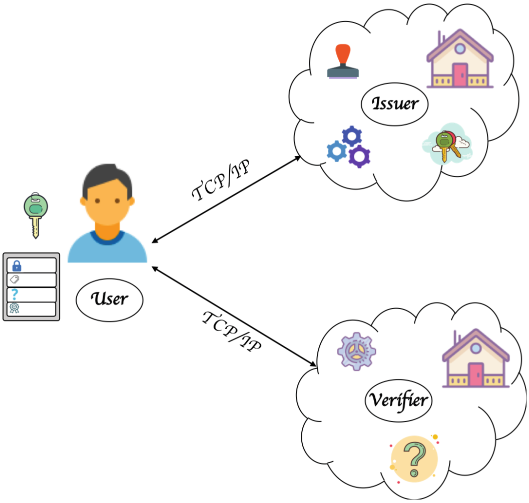


Fig. 3. Software environment

4.3 Parameters

Edwards’ curves are known to offer better performances among all Elliptic Curve (EC) families [21]. The $Curve25519$ was introduced as an ECDH (Elliptic Curve Diffie-Hellman) function but it is known today as the underlying elliptic curve designed for use with ECDH key agreement scheme ($X25519$) or with ECDSA (Elliptic Curve Digital Signature Algorithm) signature ($Ed25519$). It was first introduced in its Montgomery form $E : v^2 = u^3 + 486662u^2 + u$ over the prime field defined by $p = 2^{255} - 19$. This curve ensures a 128-bit security level as the fastest known attack on the discrete logarithm problem [22]. Nowadays, it is used in Protocols, Networks, Operating Systems, Software, SSH Software, TLS Libraries, etc. [23]. Below, we describe parameters used in our performance evaluation and they are adapted from [22]. The parameter k defines keys’ size for schemes I2PA and U-prove while k' defines Idemix’s keys size. The parameter p defines the field $\mathbb{Z}_p$, d defines the elliptic curve $E_d : x^2+y^2 = 1+dx^2y^2$. Values x_0 and y_0 define the coordinates of the base point P with order q. The component Z_0 defines the third component in extended homogeneous coordinates of the base point. We refer reader to [10,24] for keys’ size justification. Parameters’ values are depicted below:

- $k = 160$
- $k' = 1024$
- $p = 2^{255} - 19$
- $d = 3709570593466943934313808350875456518954211387984321901638878553308594028355 5$

- $x_0 = 15112221349535400772501151409588531511454012693041857206046113283949847762202$
- $y_0 = 46316835694926478169428394003475163141307993866256225615783033603165251855960$
- $Z_0 = 1$
- $q = 2^{252} + 27742317777372353535851937790883648493$

At the core of ABC schemes, we have attributes. An attribute is a characteristic or a qualification of a person [4]. It certifies that an entity has skill, knowledge, qualification, etc. An attribute certified by a third party is known as a claim. Whatever the nature of an attribute, it can be represented in a decimal format. Therefore, attributes' values used in this evaluation are described below:

- $a_0 = 3022871045856445402$
- $a_1 = 2303921356947$
- $a_2 = 63990592803$
- $a_3 = 63188281798077$
- $a_4 = 2334544185927680150715$
- $a_5 = 7247895906071689951 5$
- $a_6 = 1321084182402701079543 63$
- $a_7 = 5335947794968310 3$
- $a_8 = 39309000932222668473935279818668 3$
- $a_9 = 2930303348526267$

5 Results and Discussion

This section depicts and comments results of our performance evaluation. Unless explicitly stated, time will always be expressed in milliseconds (ms) and memory in Megabyte (MB). It should also be noted that, for memory metrics, all values are rounded to two decimal places. We remind that U-prove and I2PA are instantiated using ECC as mentioned before. The implemented versions of U-prove and Idemix are based on schemes presented by Gergely Alpár [9] while I2PA implementation is based on the scheme presented by SENE et al. [4]. We point out that every simulation is carried out with new random parameters except system's parameters and attributes' values.

5.1 Limitations

We note some limitations that should be taken into account while exploring results presented thereafter.

- Our results are based on the openly available versions of U-prove and Idemix.
- During the issuance phase and at user side, when the issuer takes lots of times to issue credential, recorded minima in terms of memory usage at user side may be biased. The user may remain idle for a while which considerably lowers used resources. This is the case with Idemix since its issuance requires lots of times (See “Fig. 7”).
- During the verification phase, in order not to impact memory usage, we first issue a credential and then verify it immediately instead of storing all credentials that should be verified. This may impact the recorded minima at verifier side if the issuance of a credential takes lots of times. The later may remain idle for a while what considerably lowers used resources. This is the case with Idemix that requires a lot of times to issue a credential (See “Fig. 7”).

5.2 Memory Usage Evaluation

This section describes results about memory usage. Figures presented below are recorded with VisualVM 1.3.9 [25] using “Tracer-Monitor Probes” plugin. Evaluations involve 100 simulations on issuance and verification of credentials made up of 1, 5, and 10 attributes respectively.

5.2.1 Issuance

In this section, we describe memory evaluation at issuer and user sides. At issuer side (respectively at user side), we evaluate the memory required to issue (respectively to get) a credential.

At Issuer Side: Recorded results from issuance of credentials made up of 1, 5, and 10 attributes respectively are presented in “Fig. 4”.

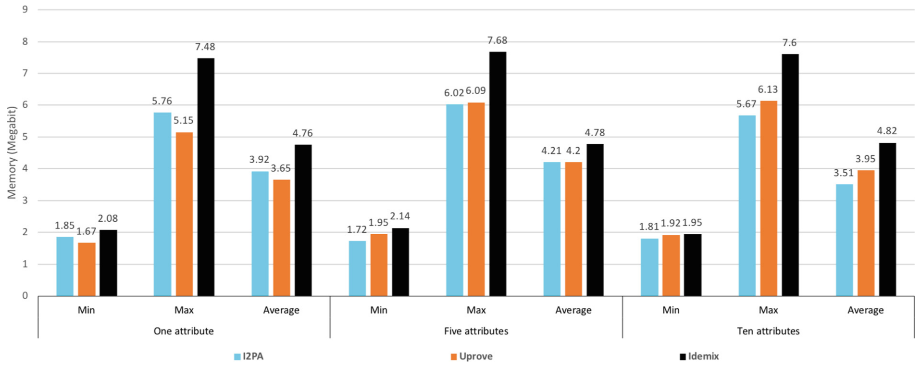


Fig. 4. Issuance memory usage at issuer side

As shown in “Fig. 4”, the three schemes present nearby performance regarding memory usage at issuer side while issuing credentials. At first glance, this may seem paradoxical seeing keys’ size (160 for I2PA and U-prove, 1024 for

Idemix). However, this can be explained by the usage of extended homogeneous coordinates while instantiating U-prove and I2PA. Nevertheless, in all cases, Idemix requires more resource in average and records highest maxima. U-prove and I2PA, in three cases, have nearby average consumptions; 3.92 against 3.65 (respectively 4.21 against 4.2, and 3.51 against 3.95) for issuance of 1 attribute (respectively 5 and 10 attributes).

At User Side: This section describes memory usage at user side while issuing credentials of 1, 5, and 10 attributes respectively. We shall not evaluate the memory usage in the verification phase since the user only presents her credential; she performs no operation. Due to limitations noted in the mobile application while recording memory usage at user side, we recorded these results with a user implemented using Java socket and running in a Raspberry PI. The "Fig. 5" is an illustration of recorded results.

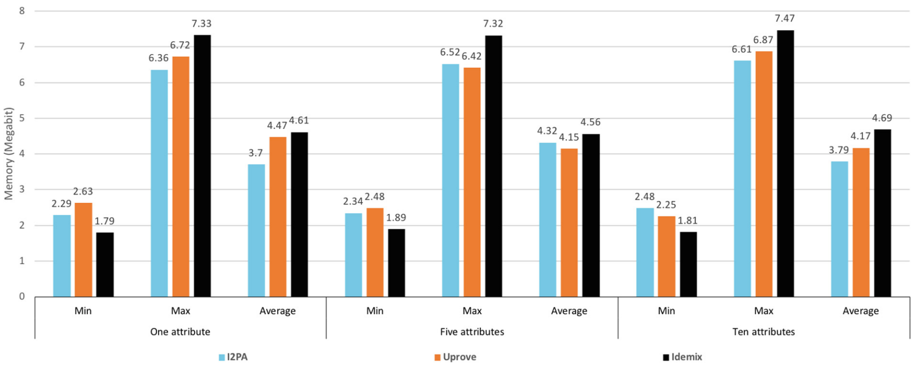


Fig. 5. Issuance memory usage at user side

"Figure 5" shows that, as we have already pointed out in the limitations (Sect. 5.1), Idemix has very low minima (1.79, 1.89, and 1.81) compared to other schemes (2.29, 2.34, and 2.48 for I2PA, 2.63, 2.48, and 2.25 for U-prove). Despite the fact that three schemes present nearby consumptions, Idemix has higher maxima and requires more resources on average.

5.2.2 Verification

If the credentials to verify are generated and stored beforehand, this may greatly affect memory usage and then influences recorded results. We verify a credential after generating it. This frees the memory once the credential is verified. "Figure 6" illustrates recorded results while verifying credentials of 1, 5, and 10 attributes respectively.

"Figure 6" shows that, globally, tendencies recorded here do not differ from those presented in previous sections. We can note that on average, Idemix, requires more memory than I2PA and U-prove. Except for the verification of

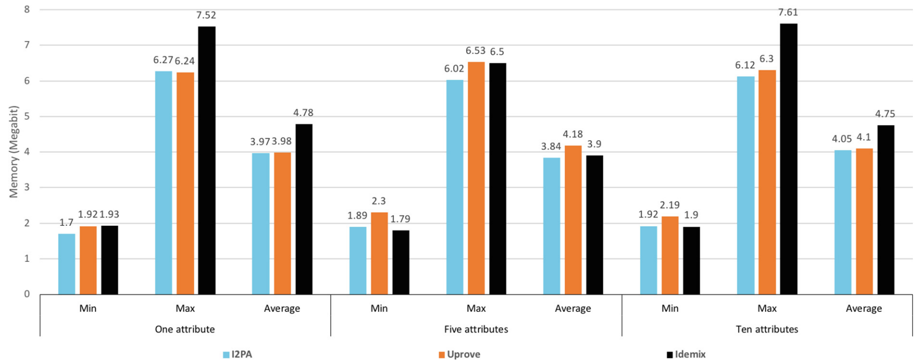


Fig. 6. Verification memory usage at verifier side

credentials made up of 5 attributes, Idemix presents the highest average values. I2PA, in all three cases, has an average value smaller than that present by U-prove and Idemix.

5.3 Time Evaluation

This section describes results recorded regarding computing time. These results concern 100 simulations involving credentials of 1, 5, and 10 attributes respectively. We shall consider the time required at user side to get a credential from an issuer as well as the one required to have a credential verified by a verifier.

5.3.1 Issuance

Results recorded from issuance of credentials made up of 1, 5, and 10 attributes are illustrated in “Fig. 7”. We illustrate minima, maxima, as well as average values.

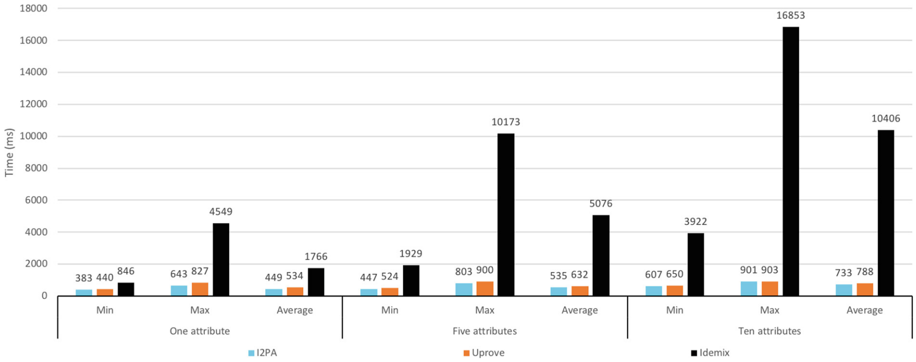


Fig. 7. Time issuance comparison

“Figure 7” shows that I2PA and U-prove have similar performance regarding computing time efficiency. However, I2PA presents more interesting result than U-prove. Idemix, meanwhile, has very low performance compared to I2PA and U-prove. The time it requires for issuance is on average at least 3 times (respectively 8 and 13) more important than that required by I2PA and U-prove for issuance of credential made up of 1 attribute (respectively 5 and 10 attributes). Regarding distribution of time for credential containing 1 attribute (respectively 5 and 10 attributes), 33% (respectively 45% and 50%) of simulations have duration higher or equal to the average for I2PA, 38% (respectively 45% and 49%) for U-prove, and 35% (respectively 39% and 51%) for Idemix. Finally, regarding computing time efficiency, I2PA and U-prove present more interesting result than Idemix. What should be the number of attributes (1, 5, or 10), I2PA and U-prove require less than 1 s for credential issuance, what can be considered as relevant.

5.3.2 Verification

As for the issuance, this section describes recorded results while verifying 100 credentials of 1, 5, and 10 attributes respectively. “Figure 8” illustrates registered results.

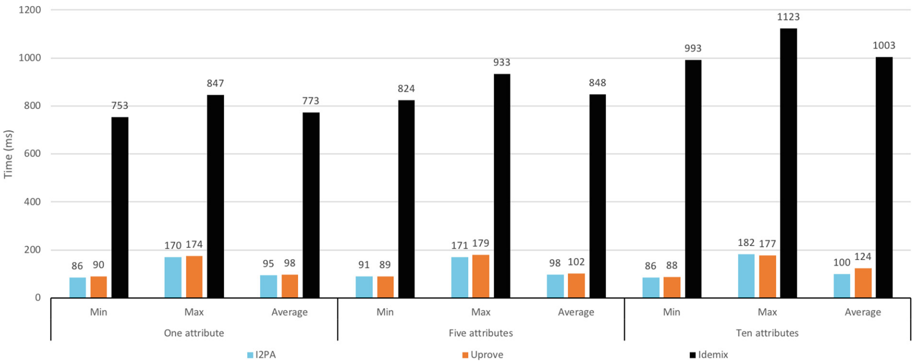


Fig. 8. Time verification comparison

“Figure 8” shows that, as for the issuance, I2PA and U-prove present similar performances on the verification protocol regarding computing time efficiency. However, except the maximum recorded during the issuance of credentials made up of 10 attributes, I2PA requires less computing time compared to U-prove. Idemix, meanwhile, has very low performance compared to I2PA and U-prove. The time it requires for verification is on average at least 7 times (respectively 8 and 8) more important than that required by I2PA and U-prove for verification of credentials made up of 1 attribute (respectively 5 and 10 attributes). Regarding distribution of time for credentials of 1 attribute (respectively 5 and 10 attributes), 22% (respectively 10% and 21%) of simulations have duration higher or equal to the average for I2PA, 13% (respectively 13% and 50%) for U-prove and 39% (respectively 54% and 12%) for Idemix. Finally, regarding computing time efficiency, we can safely assert that I2PA and U-prove present more interesting result than Idemix on verification protocol. They can thus be envisaged in a context of resource-constrained devices.

6 Conclusion and Future Works

In this paper, the performance evaluation of I2PA, U-prove, and Idemix we conducted in low-resource devices, was focused in evaluating computing time and memory usage efficiency. Three types of conclusions can be drawn:

- In terms of memory usage at issuer, user or verifier sides, I2PA, U-prove, and Idemix present nearby consumptions if I2PA and U-prove are instantiated using ECC and ECH representation. However, in average, Idemix requires more memory than I2PA and U-prove.
- In terms of computing time efficiency, Idemix has very low performances compared to I2PA and U-prove. The time it requires for issuance (respectively verification) is at least 3 times (respectively 7 times) more important than that requires by I2PA and U-prove.
- Even though EHC representation speeds up operations over the curve, it increases memory usage.

Finally, for computing time and memory usage efficiency criteria, I2PA and U-prove are two schemes that can be envisaged in an IoT context. However, for an effective choice, other criteria must be taken into account including issuance unlinkability, multi-show unlinkability, selective disclosure, randomization, etc. This evaluation, for reasons of completeness, could be extended by studying randomization and selective disclosure protocols efficiency as well as bandwidth usage.

References

1. Chen, Y.-K.: Challenges and opportunities of internet of things, pp. 383–388, January 2012. https://doi.org/10.1109/ASPDAC.2012.6164978
2. Ashton, K.: That "Internet of Things" Thing (2009). https://tools.ietf.org/html/draft-josefsson-eddsa-ed25519-03. Accessed 28 June 2019
3. Mattern, F., Floerkemeier, C.: From the internet of computers to the internet of things. In: Sachs, K., Petrov, I., Guerrero, P. (eds.) From Active Data Management to Event-Based Systems and More. LNCS, vol. 6462, pp. 242–259. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-17226-7_15
4. Sene, I., Ciss, A.A., Niang, O.: I2PA: an efficient ABC for IoT. Cryptography **3**(2), 16 (2019). https://doi.org/10.3390/cryptography3020016
5. Toumia, A., Szoniecky, S.: Prétopologie et protection de la vie privée dans l'Internet des Objets. Open Science-Internet des objets **2**(1) (2018)
6. Camenisch, J., Van Herreweghen, E.: Design and implementation of the idemix anonymous credential system. In: Proceedings of the 9th ACM Conference on Computer and Communications Security, pp. 21–30. ACM (2002)

7. Paquin, C., Zaverucha, G.: U-prove cryptographic specification v1. 1. Technical report, Microsoft Corporation (2011)
8. Alpár, G., Jacobs, B.: Credential design in attribute-based identity management (2013)
9. Alpár, G.: Attribute-based identity management: [bridging the cryptographic design of ABCs with the real world]. [Sl: sn] (2015)
10. Ciss, A.A.: Trends in elliptic curves cryptography. IMHOTEP: Afr. J. Pure Appl. Math. **2**(1), 1–12 (2015)
11. Josefsson, S., Liusvaara, I.: Edwards-curve digital signature algorithm (EDDSA). Technical report (2017)
12. Koblitz, N.: Elliptic curve cryptosystems. Math. Comput. **48**(177), 203–209 (1987)
13. Miller, V.S.: Use of elliptic curves in cryptography. In: Williams, H.C. (ed.) CRYPTO 1985. LNCS, vol. 218, pp. 417–426. Springer, Heidelberg (1986). https://doi.org/10.1007/3-540-39799-X_31
14. Rivain, M.: Fast and regular algorithms for scalar multiplication over elliptic curves. IACR Cryptology ePrint Archive, p. 338 (2011)
15. Baldimtsi, F., Lysyanskaya, A.: Anonymous credentials light. In: Proceedings of the 2013 ACM SIGSAC Conference on Computer & Communications Security, pp. 1087–1098. ACM (2013)
16. Camenisch, J., Groß, T.: Efficient attributes for anonymous credentials. ACM Trans. Inf. Syst. Secur. (TISSEC) **15**(1), 4 (2012)
17. Veseli, F., Serna, J.: Evaluation of privacy-ABC technologies - a study on the computational efficiency. In: Habib, S.M.M., Vassileva, J., Mauw, S., Mühlhäuser, M. (eds.) IFIPTM 2016. IAICT, vol. 473, pp. 63–78. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-41354-9_5
18. Veseli, F., Olvera, J.S.: Benchmarking privacy-ABC technologies - an evaluation of storage and communication efficiency, pp. 198–205, June 2015. https://doi.org/10.1109/SERVICES.2015.37
19. Vullers, P., Alpár, G.: Efficient selective disclosure on smart cards using idemix. In: Fischer-Hübner, S., de Leeuw, E., Mitchell, C. (eds.) IDMAN 2013. IAICT, vol. 396, pp. 53–67. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-37282-7_5
20. Mostowski, W., Vullers, P.: Efficient U-prove implementation for anonymous credentials on smart cards. In: Rajarajan, M., Piper, F., Wang, H., Kesidis, G. (eds.) SecureComm 2011. LNICST, vol. 96, pp. 243–260. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-31909-9_14
21. Liu, Z., Seo, H., Xu, Q.: Performance evaluation of twisted Edwards-form elliptic curve cryptography for wireless sensor nodes. Secur. Commun. Netw. **8**(18), 3301–3310 (2015)
22. El Housni, Y.: Edwards curves. Working Paper or Preprint, December 2018. https://hal.archives-ouvertes.fr/hal-01942759
23. IANIX. Things that use Ed25519 (2019). https://ianix.com/pub/ed25519-deployment.html. Accessed 25 Jan 2019
24. Sinha, R., Srivastava, H.K., Gupta, S.: Performance based comparison study of RSA and elliptic curve cryptography. Int. J. Sci. Eng. Res. **4**(5), 720–725 (2013)
25. Sedlacek, J., Hurka, T.: VisualVM, All-in-One Java Troubleshooting Tool (2017). https://visualvm.github.io/. Accessed 05 Mar 2019