Anonymous Credentials with Practical Revocation using Elliptic Curves

Petr Dzurenda[1], Jan Hajny[1], Lukas Malina[1] and Sara Ricci[2]
[1]*Brno University of Technology, Technicka 12, Brno, Czech Republic*
[2]*UNESCO Chair in Data privacy, Department of Computer Science and Mathematics, Universitat Rovira i Virgili, Av. Països Catalans 26, Tarragona, Catalonia, Spain*

Keywords: Attribute-based Credentials, Anonymity, Efficient Revocation, Elliptic Curves, Privacy, Smart Cards.

Abstract: Anonymous Attribute-Based Credential (ABC) schemes allow users to anonymously prove the ownership of their attributes, such as age, citizenship, gender. The ABC schemes are part of a larger group of cryptographic constructions called Privacy Enhancing Technologies (PETs), aiming to increase user's privacy. In the article, we present a new ABC scheme based on elliptic curves and HM12 scheme. The scheme provides anonymity, untraceability, unlinkability, selective disclosure of attributes, non-transferability, revocation and malicious user identification. By involving elliptic curves, we achieved faster verification phase (by 30%) and smaller communication cost between user and verifier (by 85%) compared to the original HM12 scheme, with equivalent or greater security level.

1 INTRODUCTION

Current authentication schemes use identity-based authentication approach, i.e., a user reveals his identity to a verifier. In other words, a verifier asks a user the question ”Who are you?” at first. This question has a big impact on user's privacy. Nevertheless, user's identity does not need to be always disclosed. For instance, a hospital wants to give an access to the HIV discussion group to a user who presents this disease, but a patient could be reluctant to participate in the group if he has to reveal his identity. In many cases, it is enough to know some particular user's attributes only.

ABC schemes are modern cryptographic schemes which provide higher protection of users' privacy during a verification phase. Users can anonymously prove the possession of some personal attributes without disclosing their identity or any other sensitive information. ABC schemes change the question from ”Who are you?” to ”What can you do?”, which is more privacy-preserving for a user and sufficient for a verifier. A user in the system holds some personal attributes, and during the verification phase, he proves the possession of the attributes required by a verifier. The examples of attributes include age, citizenship, gender or nationality for eIDs (electronic ID cards), validity of ticket for public transportation, etc.

Elliptic curves cryptography (ECC) provides security level comparable to classic systems while using fewer bits and less computing power. For this reason, ECC is very suitable for ultra-low-power devices. Nowadays, there are only few well-known ABC schemes such as U-Prove (Christian Paquin, 2013), Idemix (Bichsel et al., 2010), Verheul (Verheul, 2001) and HM12 (Hajny and Malina, 2013; Hajny et al., 2014). Most of them can be easily constructed over ECC except the HM12 scheme.

In this article, we present a new ABC scheme based on elliptic curves (ECs) and the HM12 scheme. Our variant meets all requirements for an ABC scheme, in particular anonymity, untraceability, unlinkability, selective disclosure of attributes, non-transferability, revocation and malicious user identification. Furthermore, by involving ECs in the scheme, we achieve higher computational efficiency compared with the standard HM12 scheme, especially during the verification phase.

1.1 Related Work

There are only few practical ABC schemes, such as U-Prove (Christian Paquin, 2013), Idemix (Bichsel et al., 2010), Verheul (Verheul, 2001) and HM12 (Hajny and Malina, 2013; Hajny et al., 2014), which allow users to anonymously prove the possession of

Dzurenda, P., Hajny, J., Malina, L. and Ricci, S.
Anonymous Credentials with Practical Revocation using Elliptic Curves.
DOI: 10.5220/0006467705340539
In *Proceedings of the 14th International Joint Conference on e-Business and Telecommunications (ICETE 2017) - Volume 4: SECRYPT*, pages 534-539
ISBN: 978-989-758-259-2

their attributes. These schemes also provide untraceability, which means that an issuer, who issued attributes to a user, is not able to track the user during the verification phase. U-prove is a cryptographic technology maintained by Microsoft Corporation. The main drawback of the scheme is the session linkability, i.e., all anonymous credentials of a single user are mutually linkable. Moreover, U-prove does not provide features for malicious user identification. Idemix technology (Identity Mixer) is an anonymous credential system developed at IBM Research in Zurich. Idemix provides session unlinkability. On the other hand, there is no universal efficient revocation mechanism, therefore it is not possible to directly revoke users' credentials and identify malicious users. Verheul scheme allow users to randomize their key pairs and the corresponding certificate. The drawbacks of the scheme are the unsupported selective disclosure of attributes and the inefficient revocation mechanism. At last, HM12 scheme solves all drawbacks of the previous schemes by providing anonymity, untraceability, unlinkability, selective disclosure of attributes and non-transferability. Revocation of anonymous credentials and identification of malicious users are made possible by using Okamoto-Uchiyama trapdoor (Okamoto and Uchiyama, 1998).

The most efficient implementation of U-prove protocol was done on a MultOS card and was described in (Mostowski and Vullers, 2011). The proof of attribute ownership is faster than 1 s. Idemix was also implemented on MultOS card and its proof of attribute ownership needs around 1.2 s. EC implementation of Verhoul's scheme on Java Card was described in (Batina et al., 2010), where the proof of attribute ownership implementation lasts around 1.5 s. Implementation results of HM12 on MultOS card platform were published in (Hajny and Malina, 2013), where the implementation of attribute ownership proof requires more than 2 s.

2 PRELIMINARIES

We use the notation introduced by Camenisch and Stadler (Camenisch and Stadler, 1997) to describe Proof of Knowledge (PK) protocols. Let c be a number in a finite group $\mathbb{K}$ and g a generator of the same group $\mathbb{K}$, the protocol proving the knowledge of discrete logarithm of c with respect to g is denoted as $PK\{w : c = g^w\}$. Equivalently, given C, G two points of an elliptic curve E over a finite field $\mathbb{F}$, where G is a base point of E, the protocol proving the knowledge of EC discrete logarithm of C with respect to G is denoted as $PK\{w : C = w \bullet G\}$. Furthermore, we use the proof of representation denoted as $PK\{w_0, w_1, \dots, w_i : c = g^{w_0} \cdot g^{w_1} \cdots g^{w_i}\}$ in the standard variant and as $PK\{w_0, w_1, \dots, w_i : C = w_0 \bullet G_0 + w_1 \bullet G_1 + \cdots + w_i \bullet G_i\}$ in the EC variant. The proof of discrete log equivalence with respect to different generators g_1, $g_2 \in \mathbb{K}$ is denoted as $PK\{w : c_1 = g_1^w \wedge c_2 = g_2^w\}$. A signature by a traditional scheme (e.g., RSA) of a user U on some data is denoted as $Sig_U(data)$. The symbol "$\cdot$" denotes multiplication, "$\bullet$" denotes scalar EC point multiplication, ":" means "such that", "$|$" means "divides", "$|x|$" is the bitlength of x, and "$x \in_R \{0,1\}^l$" is a randomly chosen bitstring of maximum length l. As in the original HM12 scheme, we also use the trapdoor one-way function of Okamoto-Uchiyama (OU) cryptosystem (Okamoto and Uchiyama, 1998) during the attribute issuance, nevertheless, the verification phase completely runs over ECs. OU cryptosystem relies on the assumption that the discrete logarithm problem is hard to compute in OU groups similarly as in RSA composite groups. However, if the factorization of the OU modulus $n = r^2 \cdot s$ is known, i.e., r, s are known, the discrete logarithms can be efficiently computed and, therefore, it is possible to recover w from $c = g^w \bmod n$ using the following equation,

$$w = \mathrm{dlog}_g c = \frac{[(c^{r-1} \bmod r^2) - 1]/r}{[(g^{r-1} \bmod r^2) - 1]/r} \bmod r, \quad (1)$$

where $g \in \mathbb{Z}_n$ is the generator of the OU group such that $g \bmod r^2$ is a primitive element of $\mathbb{Z}_{r^2}$ and the value r is the trapdoor in the OU scheme.

3 PROPOSED SCHEME

The entities involved in the ecHM12 scheme are following:

- **User (U)** – gets issued attributes from Issuer and anonymously proves their possession to Verifier.
- **Issuer (I)** – is responsible for issuing user attributes.
- **Revocation Authority (RA)** – validates user credentials (collection of attributes issued by Issuer), can revoke a (dishonest) user, and in collaboration with Issuer, can identify the (dishonest) user.
- **Verifier (V)** – verifies possession of required attributes provided by User.

Each entity communicates in the system through specific cryptographic protocols. All the protocols and involved entities are depicted in Figure. 1.

As well as the HM12 scheme, also ecHM12 scheme provides the properties required for ABC schemes:

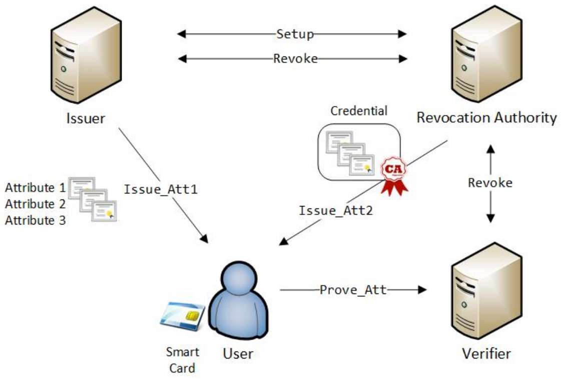

Figure 1: Architecture of proposed ecHM12 scheme.

- **Anonymity** – U anonymously proves possession of attributes. Therefore, his identity and his behaviour remains hidden in the system.
- **Untraceability** – all the credentials are randomized, i.e., I is not able to track U's movements and behaviour.
- **Unlinkability** – all sessions are mutually unlinkable. Therefore, V or an eavesdropper are not able to link individual sessions together and profile U.
- **Non-transferability** – U is equipped with unique private key, which is stored on a secure element, e.g. a smart card.
- **Selective disclosure of attributes** – U can choose the attributes which have to be disclosed, other attributes remain hidden.
- **Revocation** – RA is able to remove U from the system, to revoke U's credentials or to disclose U's identity and to revoke the unlinkability property.

3.1 Cryptography Specification

In this section we provide a description of each protocol which runs within the proposed scheme.

3.1.1 `Setup` Protocol

$(sysparam, K_{RA}, K_I) \leftarrow \mathtt{Setup}(k, l, m)$ – The `Setup` protocol mostly matches the original HM12 scheme, only in the final step the scheme is switched to the EC variant. The main purpose of this protocol is to establish *sysparam* and to generate K_I and K_{RA}. The input parameters k, l, m define the security level of the cryptographic scheme, where k presents the length of the hash function, l is related to the length of U's secrets and m is the verification error parameter. I defines a group $\mathbb{H}$ modulo big prime number p and generators of order q, with $q|p-1$. $\mathbb{H}$ is the subgroup of the group $\mathbb{Z}_p^*$ as in the DSA signature scheme. In addition, I generates the key pair sk_I and pk_I for signing purpose using a defined signature scheme, e.g. RSA. RA needs to:

- define the OU group $\mathbb{G}$ by specifying the modulus $n = r^2 \cdot s$, where r and s are big prime numbers ($|r| > 720$, $|r| > 2|q|$, $|n| \geq 2048$, $r = 2r' + 1$, $s = 2s' + 1$, where r' and s' are primes).
- generate $g_1 \in_R \mathbb{Z}_n^*$ of $ord(g_1 \bmod r^2) = r(r-1)$ in $\mathbb{Z}_{r^2}^*$ and $ord(g_1) = r \cdot r' \cdot s'$ in $\mathbb{Z}_n^*$.
- choose an EC over finite field $E(\mathbb{F}_p)$ with the domain parameters (a, b, p, q, G, h), where p is an big prime number specifying the field $\mathbb{F}_p$, $a, b \in \mathbb{F}_p$ are coefficients of the EC, G is an EC point generator $G = (x_G, y_G)$ of order q, and the h is the cofactor defined by $h = \#E(\mathbb{F}_p)/q$.
- randomly choose RA secrets $s_1, s_2, s_3 \in_R \mathbb{Z}_q$, such that $GCD(s_1, q) = GCD(s_2, q) = GCD(s_3, q) = 1$.
- compute $g_2 = g_1^{s_2} \bmod n$ in the OU group.
- compute $G_1 = G$, $ecA_{seed} = s_1 \bullet G_1$, $G_2 = s_2 \bullet G_1$ and $G_3 = s_3 \bullet G_1$ over $E(\mathbb{F}_p)$.

The system parameters $sysparam = (g_1,\ g_2, h_1, h_2, n = r \cdot s, \mathbb{H}, \mathbb{G}, E(\mathbb{F}_p), G_1, G_2, G_3, ecA_{seed}, pk_I)$ are made public, the values r, s representing the RA key K_{RA} are securely stored by RA, and the key $K_I = sk_I$ is securely stored by I.

3.1.2 `Issue_Att` Protocol

$(K_U) \leftarrow \mathtt{Issue\_Att}(sysparam, K_{RA}, K_I)$ – Following the HM12 idea, the protocol is split into two parts `Issuer_Att1` and `Issuer_Att2` protocols, see Figure 2. The goal is to compute U's key $K_U = \{w_1, w_2, w_r\}$. `Issuer_Att1` runs between U and I. U generates a cryptographic commitment $\bar{H} = h_1^{w_1} \cdot h_2^{w_2} \bmod p$ in $\mathbb{H}$, where U's keys w_1, w_2 are committed values. After, U signs the commitment with his private key sk_U and sends it and the signature with the proof of construction PK to I. I verifies the signature and signs U's commitments by his private key sk_I. Commitments are stored by I for identification and revocation purposes. Any secure signature scheme, e.g. RSA, DSA, can be used. `Issuer_Att2` runs between U and RA. U computes another commitment $\bar{A} = g_1^{w_1} \cdot g_2^{w_2} \bmod n$ in OU group $\mathbb{G}$ and sends $\bar{A}$, $\bar{H}$, the signature of $\bar{H}$ (generated by I) and the proof of discrete log equivalence PK to RA. Now, RA is able to compute the User's key w_r using the Equation 1 such that the following equations hold:

$$\begin{aligned} ecA_{seed} &= w_1 \bullet G_1 + w_2 \bullet G_2 + w_r \bullet G_3 \\ ecA_{seed} &= w_1 \bullet G_1 + w_2 \cdot s_2 \bullet G_1 + w_r \cdot s_3 \bullet G_1 \qquad (2) \\ ecA_{seed} &= (w_1 + w_2 \cdot s_2 + w_r \cdot s_3) \bullet G_1 \end{aligned}$$

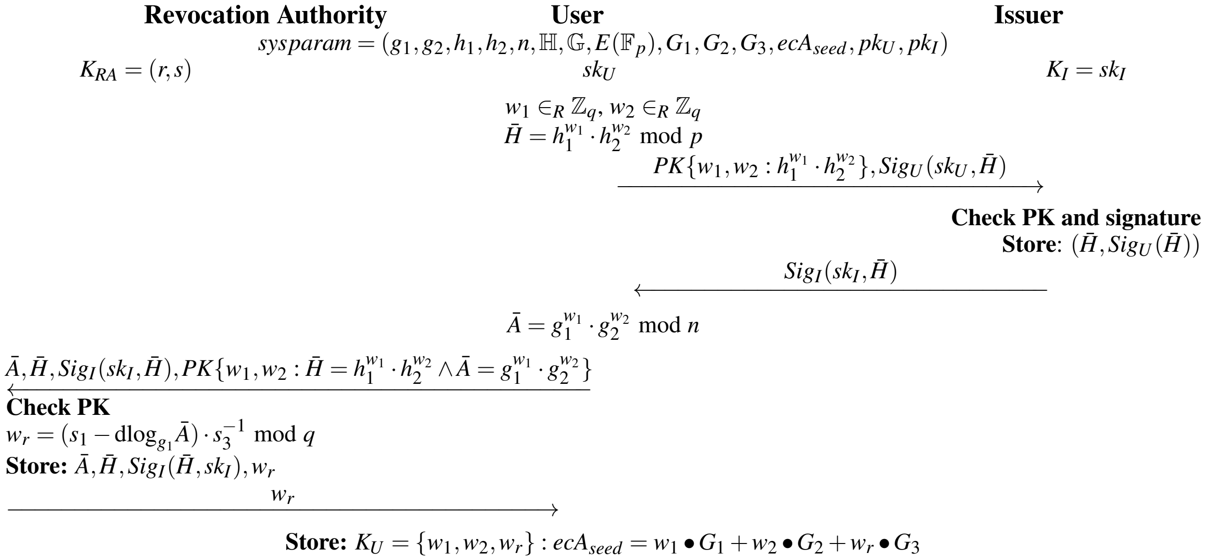
Revocation Authority
User
Issuer
$sysparam = (g_1, g_2, h_1, h_2, n, \mathbb{H}, \mathbb{G}, E(\mathbb{F}_p), G_1, G_2, G_3, ecA_{seed}, pk_U, pk_I)$
$K_{RA} = (r, s)$
sk_U
$K_I = sk_I$
$w_1 \in_R \mathbb{Z}_q$, $w_2 \in_R \mathbb{Z}_q$
$\bar{H} = h_1^{w_1} \cdot h_2^{w_2} \bmod p$
$PK\{w_1, w_2 : h_1^{w_1} \cdot h_2^{w_2}\}, Sig_U(sk_U, \bar{H})$
Check PK and signature
Store: $(\bar{H}, Sig_U(\bar{H}))$
$Sig_I(sk_I, \bar{H})$
$\bar{A} = g_1^{w_1} \cdot g_2^{w_2} \bmod n$
$\bar{A}, \bar{H}, Sig_I(sk_I, \bar{H}), PK\{w_1, w_2 : \bar{H} = h_1^{w_1} \cdot h_2^{w_2} \wedge \bar{A} = g_1^{w_1} \cdot g_2^{w_2}\}$
Check PK
$w_r = (s_1 - \mathrm{dlog}_{g_1}\bar{A}) \cdot s_3^{-1} \bmod q$
Store: $\bar{A}, \bar{H}, Sig_I(\bar{H}, sk_I), w_r$
w_r
Store: $K_U = \{w_1, w_2, w_r\} : ecA_{seed} = w_1 \bullet G_1 + w_2 \bullet G_2 + w_r \bullet G_3$

Figure 2: `Issue_Att` protocol of the ecHM12 scheme.

$\bar{A}, \bar{H}$ and w_r are stored in RA's database and w_r is sent to U. U securely stores $K_U = (w_1, w_2, w_r)$, eg. on a smart card.

3.1.3 Prove_Att Protocol

$(proof) \leftarrow$ `Prove_Att`$(sysparam, K_U)$ – This protocol runs fully over $E(\mathbb{F}_p)$. The protocol is depicted in Figure 3. U proves the ownership of attributes (w_1, w_2, w_r) to V using PK protocols. The unlinkability is provided by using the random number K_S, which is re-generated in every session. Moreover, the protocol provides revocation features by committing the value K_S in the commitment C_2 and the committed value w_r (revocable key part of the User's key) in the commitment C_1. C_1 and C_2 permit to check if the U is in the black list or not, and to remove him from the system by involving RA in the revocation process. The verification time depends on the amount of disclosed attributes by U and on the amount of all revoked Us.

3.1.4 Revoke Protocol

$(rev) \leftarrow$ `Revoke`$(sysparam, proof, K_{RA})$ – The original HM12 scheme uses OU trapdoor to solve discrete logarithm problem. In ecHM12 scheme, this trapdoor cannot be used. However, revocation of a dishonest user is still possible. The protocol input parameters are system parameters *sysparam* and *proof* generated by the User within `Prove_Att` protocol. The revocation part of the proof consists of commitments C_1 and C_2. RA computes Equation 3 for all user keys $w_{r_{DATABASE}}$ in RA's database until a match is found.

$$w_{r_{DATABASE}} \bullet C_2 \stackrel{?}{=} C_1 \qquad (3)$$

If a match is found, the commitment that belongs to this particular U is revoked by publishing w_r on a black list. The revocation complexity is linear in the number of Us instead of constant as in the HM12 scheme. Yet revocation remains practical, see Section 5 for implementation details. On the other hand, the protocol `Prove_Att` is faster than in HM12 scheme.

4 SECURITY ANALYSIS

The `ProveAtt` protocol is a standard proof of knowledge protocol that can be denoted as $PK\{(K_S w_1, K_S w_2, K_S w_r, K_S) : A = K_S \bullet ecA_{seed} \wedge A = K_S w_1 \bullet G_1 + K_S w_2 \bullet G_2 + K_S w_r \bullet G_3\}$.

Completeness. (i.e., honest users are always accepted by the protocol) is given by the design of the protocol and can be proven by expanding verifier's equations.

Soundness. (i.e., dishonest users are always rejected by the protocol) is proven by employing the standard PK knowledge extractor that can extract $(K_S w_1, K_S w_2, K_S w_r, K_S)$ and thus obtain valid user keys (w_1, w_2, w_r). Thus, the `ProveAtt` protocol never accepts a user that does not know correct keys.

Zero-Knowledge. (i.e., the protocol does not release any information about user keys) is proven by creat-

User **Verifier**

$$sysparam = (g_1, g_2, h_1, h_2, n = r \cdot s, \mathbb{H}, \mathbb{G}, E(\mathbb{F}_p), G_1, G_2, G_3, ecA_{seed}, pk_I)$$

$K_U = \{w_1, w_2, w_r\}$

$K_S \in_R \mathbb{Z}_q$
$A = K_S \bullet ecA_{seed}$
$C_1 = (K_S \cdot w_r) \bullet G_3,\ C_2 = K_S \bullet G_3$
$r_1, r_2, r_2, r_S \in_R \mathbb{Z}_q$
$\bar{ecA}_{seed} = r_1 \bullet G_1 + r_2 \bullet G_2 + r_3 \bullet G_3$
$\bar{A} = r_S \bullet ecA_{seed}$
$\bar{C}_1 = r_3 \bullet G_3$
$\bar{C}_2 = r_S \bullet G_3$ $\xrightarrow{A, \bar{ecA}_{seed}, \bar{A}, C_1, C_2, \bar{C}_1, \bar{C}_2,}$

$\xleftarrow{e}$ $e \in_R \mathbb{Z}q$

$z_1 = (r_1 - e \cdot K_S \cdot w_1) \bmod q$
$z_2 = (r_2 - e \cdot K_S \cdot w_2) \bmod q$
$z_3 = (r_3 - e \cdot K_S \cdot w_r) \bmod q$
$z_S = (r_S - e \cdot K_S) \bmod q$ $\xrightarrow{z_1, z_2, z_3, z_S}$ **Check BL:** $C_2 \bullet w_{rblacklisted} \overset{?}{=} C_1$

$$\bar{ecA}_{seed} \equiv e \bullet A + z_1 \bullet G_1 + z_2 \bullet G_2 + z_3 \bullet G_3$$
$$\bar{A} \equiv e \bullet A + z_S \bullet ecA_{seed},\ \bar{C}_1 \overset{?}{\equiv} e \bullet C_1 + z_3 \bullet G_3,\ \bar{C}_2 \overset{?}{\equiv} e \bullet C_2 + z_S \bullet G_3,$$

Figure 3: `Prove_Att` protocol of the ecHM12 scheme.

ing the zero-knowledge simulator that can simulate the `ProveAtt` protocol. The simulator is constructed in the standard way, that is by choosing the answers z' in random and reconstructing the remaining values using Verifier's equations:

$$(A', C'_1, C'_2) \in_R E(\mathbb{F}_p), (z'_1, z'_2, z'_3, z'_S, e') \in_R \mathbb{Z}_q \tag{4}$$

$$\begin{aligned} \bar{ecA'}_{seed} &= e \bullet A' + z'_1 \bullet G_1 + z'_2 \bullet G_2 + z'_3 \bullet G_3 \\ \bar{A}' &= e \bullet A' + z'_S \bullet ecA_{seed} \\ \bar{C'}_1 &= e \bullet C'_1 + z'_3 \bullet G_3 \\ \bar{C'}_2 &= e \bullet C'_2 + z'_S \bullet G_3 \end{aligned} \tag{5}$$

The simulator's output $(A', C'_1, C'_2, z'_1, z'_2, z'_3, z'_S, e', \bar{ecA'}_{seed}, \bar{A}', \bar{C'}_1, \bar{C'}_2)$ is indistinguishable from the real transcript of the `ProveAtt` protocol $(A, C_1, C_2, z_1, z_2, z_3, z_S, e, \bar{ecA}_{seed}, \bar{A}, \bar{C}_1, \bar{C}_2)$.

5 EXPERIMENTAL RESULTS

ABC schemes are usually implemented using Java Card and MultOS smart card platforms. The Java Card (JC) platform lacks modular operations support as well as EC primitives support. Basic operations over ECs are available on JC-3.0.5, but there is still no smart card with this operation system *available*. Therefore, we cannot consider JC in our tests. On the other hand, MultOS cards support EC point addition and EC scalar multiplication over ECs. We use Multos 4.2.1 card to compare the HM12 standard scheme with the proposed ecHM12 scheme. We measured both schemes with comparable security level defined by NIST (Barker, 2016), i.e., 1392 bit version of HM12 and 160 bit version of ecHM12, and we also provided a comparison with 224 bit version of ecHM12 (higher level of security with respect to the previous values). On the smart card side, the comparison of the `Prove_Att` protocol is shown in Table 1. The ecHM12 scheme is faster than HM12 scheme in the verification phase, even if a much higher security level of the ecHM12 scheme is used. Note that the efficiency of the verification phase is crucial for the scheme's speed, thus user friendliness. The EC scalar multiplication (ecPointMul) over $E(\mathbb{F}_{160})$ takes only 43 ms and 52 ms over $E(\mathbb{F}_{224})$ instead of 94 ms, that is the time required by modular exponentiation with 1392 bit base length and 560 bit exponent length in $\mathbb{Z}_n^*$. Data transmission is also improved: we need to transfer only 220 B in case of $E(\mathbb{F}_{160})$ or 308 B in case of $E(\mathbb{F}_{224})$ instead of 1,558 B in original scheme (1392 bit version) in the `Prove_Att` protocol. On the V side, the time needed for checking blacklist is also more efficient in ecHM12 scheme than in HM12 scheme because of the involved operations: ecHM12 uses scalar multiplication and HM12 uses the slower modular exponentiation.

For ecHM12 scheme, the revocation mechanism complexity is linear instead of constant as in HM12 scheme. However, we expect RA to be computationally strong and, consequently, the slow-down does not really affect the protocol complexity. We use old-

Table 1: Comparison results in milliseconds for 1392 bit version of the HM12 scheme and equivalent variant 160 bit and 224 bit versions of the proposed ecHM12 scheme.

		HM12 1392 bit		ecHM12 160 bit		ecHM12 224 bit	
Operation	tpo	n.	tt	n.	tt	n.	tt
mExp(160)	46	3	138	0	-	0	-
mExp(400)	72	2	150	0	-	0	-
mExp(560)	94	1	94	0	-	0	-
mExp(720)	112	2	224	0	-	0	-
mExp(880)	131	2	262	0	-	0	-
mMul	100	9	900	6	600	6	600
Sub	50	3	150	3	150	3	150
RNG	49	5	245	5	245	5	245
ecMul	52/48	0	-	10	480	10	520
ecAdd	25/23	0	-	2	46	2	50
Total	-	-	**2163**	-	**1521**	-	**1565**

Note: tpo - time per operation, n. - number of operations, tt - total time per operation.

ish mid-range server, namely the 2009 IBM x3550 M2 with two Intel Xeon 2.27 GHz processors with 8 cores each and 32 GB RAM, to represent RA. The EC scalar multiplication over $E(\mathbb{F}_{224})$ took negligible 0.0189 ms, i.e. with $100,000$ users in the system, the revocation time will be 1.9 s at maximum.

6 CONCLUSIONS

We presented a new ABC scheme based on ECs and HM12 scheme. This variant meets all standard requirements on ABC schemes, i.e. anonymity, untraceability, unlinkability, selective disclosure of attributes, non-transferability, revocation and malicious user identification. By involving elliptic curves, the ecHM12 is faster in the Prove_att protocol, which makes the scheme more applicable in current access control systems. Prove_att protocol (on card) is about 30% faster than in the HM12 scheme. The efficiency advantage of our scheme grows with a higher security level of schemes. Our solution has also good impact on bandwidth, in fact, lower amount of data is transferred. Data communication is 85% smaller compared to HM12 protocol and considering comparable security level (1392 bit / 160 bit).

The revocation process requires linear time in the number of Us instead of constant time of the HM12 scheme, but, considering that the current servers have high computing power, the slow-down does not really affect the protocol usability. Our next steps are the MultOS smart card optimisation and black list check optimization on V's side. Further, we would like to improve the complexity of the Revoke protocol.

ACKNOWLEDGEMENTS

Research was sponsored by the Technology Agency of the Czech Republic project TA04010476 ”Secure Systems for Electronic Services User Verification”, the National Sustainability Program LO1401, European Commission (project H2020 644024 CLARUS) and Spanish Government (Sec-MCloud TIN2016-80250-R). For the research, infrastructure of the SIX Center was used.

REFERENCES

Barker, E. (2016). Recommendation for key management part 1: General (revision 4). *NIST Special Publication Part 1*, 800(57):1–147.

Batina, L., Hoepman, J.-H., Jacobs, B., Mostowski, W., and Vullers, P. (2010). Developing efficient blinded attribute certificates on smart cards via pairings. In *CARDIS*, pages 209–222. Springer.

Bichsel, P., Binding, C., Camenisch, J., Groß, T., Heydt-Benjamin, T., Sommer, D., and Zaverucha, G. (2010). Specification of the identity mixer cryptographic library version 2.3.0*. Technical report, IBM.

Camenisch, J. and Stadler, M. (1997). Efficient group signature schemes for large groups. *Advances in Cryptology—CRYPTO'97*, pages 410–424.

Christian Paquin, G. Z. (2013). U-prove cryptographic specification v1.1. In *Microsoft*, pages 1–23.

Hajny, J., Dzurenda, P., and Malina, L. (2014). Privacy-pac: Privacy-enhanced physical access control. In *Proceedings of the ACM CCS*, WPES '14, pages 93–96, New York, NY, USA. ACM.

Hajny, J. and Malina, L. (2013). Unlinkable attribute-based credentials with practical revocation on smart-cards. In *Smart Card Research and Advanced Applications: 11th International Conference, CARDIS 2012, Graz, Austria, November 28-30, 2012, Revised Selected Papers*, pages 62–76, Berlin, Heidelberg. Springer Berlin Heidelberg.

Mostowski, W. and Vullers, P. (2011). Efficient u-prove implementation for anonymous credentials on smart cards. In *International Conference on Security and Privacy in Communication Systems*, pages 243–260. Springer.

Okamoto, T. and Uchiyama, S. (1998). A new public-key cryptosystem as secure as factoring. In *International Conference on the Theory and Applications of Cryptographic Techniques*, pages 308–318. Springer.

Verheul, E. R. (2001). Self-blindable credential certificates from the weil pairing. pages 533–551.