Password Based Key Exchange with Hidden Elliptic Curve Public Parameters

Julien Bringer[1], Hervé Chabanne[1,2], and Thomas Icart[1,3]

[1] Sagem Sécurité
[2] Télécom ParisTech
[3] Université du Luxembourg

Abstract. We here describe a new Password-based Authenticated Key Exchange (PAKE) protocol based on elliptic curve cryptography. We prove it secure in the Bellare-Pointcheval-Rogaway (BPR) model. Our proposal is conceived in a such a way that it ensures that the elliptic curve public parameters remain private. This is important in the context of ID contactless devices as, in this case, it is easy to link these parameters with the nationality of the ID document owners.

1 Introduction

To enable secure communication over insecure channels, two-parties encrypt and authenticate their messages using a shared secret key which is usually obtained via a key exchange. A key exchange scheme enables the two parties to establish a common secret in an authenticated way (Authenticated Key Exchange, AKE). The goal is that the session key should only be known from the parties involved in the protocol: the session key should be indistinguishable from a random data. Password-based key exchange protocols are a convenient way to achieve this. The two parties rely on a shared low-entropy secret (e.g. a four-digit PIN) to derive a common high-entropy session key.

Password-Based Authenticated Key Exchange (PAKE) protocols are now considered in the context of identity documents to ensure the security of the communication between the chip and a reader [15]. With machine readable travel documents (MRTD), the data stored on the machine readable zone (MRZ) are seen as low-entropy shared information between the reader and the chip to establish a secure link. For efficiency constraints, the protocols usually rely on an elliptic curve setting. In such context, the parameters would almost surely depend on the nationality of the document owner. However, if one eavesdrops the communication of an AKE protocol based on elliptic curves, then he would learn some elliptic curve points and would be able to obtain the elliptic curve parameters. With this knowledge, he may retrieve the owner's nationality, thus leading to a privacy leakage (which itself can conduce to security issues). This is a great motivation to find PAKE protocols based on elliptic curves while enabling the elliptic curve parameters to remain hidden. To the best of our knowledge, our work is the first to provide a secure solution to this problem.

1.1 Related Works

AKE protocols Password-based Authenticated Key Exchange was considered first by Bellovin and Merritt [4]. The goal is to authenticate a key exchange between two parties based on simple passwords possibly from a small dictionary that an adversary may know. The basic idea behind the several schemes described in [4] was to manage the key exchange partially in an encrypted form (hence the acronym EKE for Encrypted Key Exchange). This work has been followed by many variants and later on several security analysis in different models, e.g. [2–5, 7–9].

The main issue is the design of a protocol which could resist offline dictionary attacks. Other security properties are also possible such as forward secrecy.

One of the most well-know variant of EKE is Diffie-Hellman EKE (DH-EKE) which is merely a DH key exchange where at least one message exchanged under DH is encrypted via the password. Bellare *et al.* introduced in [2] a formal security model to grasp the specificity of password-based key exchange which has been used later in [8, 9] to establish the security of DH-EKE under ideal assumptions (namely the Ideal Cipher Model and the Random Oracle Model, ROM). However Ideal Cipher Model is not easily applicable to elliptic curves and thus in its classical version, DH-EKE is not well adapted to elliptic curves. A naive application of the encryption step of DH-EKE to elliptic curves points would lead to an insecure scheme. Due to the redundancy in a point representation, partition attacks [13] are made possible to distinguish possible passwords from impossible ones and the security against offline dictionary attacks does not hold. In [7], a modification of the scheme is suggested by using either a point of the curve or a point over its twist. Although, this enables to withstand the security issue, the scheme is not proved in the model of [2] and becomes much less simple than DH-EKE.

Another well-known PAKE, and widely used, is SPEKE (Simple Password Exponential Key Exchange). It is part of the IEEE P1363.2 standard and is introduced in [12]. Likewise, it is based on Diffie-Hellman key exchange but the password is here used to select the generator of the DH key exchange, which is then operated in clear. To do so, a hash of the password is used to generate a group element, and the security is based on the ROM. Here again such operation is not straightforward to transpose into elliptic curves, although becoming feasible via the recent work of [10].

Finally, in both cases, DH-EKE and SPEKE do not enable to hide the curve parameters which are easily deduced from 2 eavesdropped points.

The BPR (Bellare-Pointcheval-Rogaway) model Several security models have been suggested to analyze the security of password-based AKE. The BPR model [2] is now considered as a standard model for PAKE protocols. It captures well the security requirements that a PAKE should satisfy. In particular even if protocols remain always subject to online guessing attacks, it should thwart offline dictionary attacks. Forward secrecy is another possible aspect. Different protocols have been shown secure thanks to this model. The model is based on the Find-Then-Guess principle where an adversary – mounting an active attack against several protocol instances running concurrently – should not be able to determine whether a session key is the actual one, i.e. that the key should be indistinguishable from a random string. This ensures an implicit authentication between the two parties involved in the protocol. The model is refined in [1] by requiring that all the concurrent session keys look random (Real-Or-Random principle).

Admissible Encodings The notion of admissible encoding has been introduced by Boneh-Franklin in order to hash into elliptic curves since it is required for their Identity-Based Encryption scheme [6]. Later, Coron and Icart [10] have introduced a more general notion of admissible encodings. These encodings are build from deterministic functions into elliptic curves such the ones of [11, 14].

An admissible encoding from $\{0,1\}^*$ into a group is a function which enables to transform any bit string into a group element. To be such an encoding, it must exist a polynomial time inversion algorithm, which is able to compute a bit string from a group element. This algorithm ensures that for a random group element as input, the output is a random bit string. Thanks to these encodings, [10] describes a way to create a random oracle into elliptic curve. We here use the fact that a random point can be represented by a random bit string, whatever are the elliptic curve public parameters.

This property can be used within the DH-EKE protocol, as it is much safer to encrypt random bit strings, especially when the key is directly derived from a low entropy password.

1.2 Our Contribution

We provide the first PAKE protocol which ensures that the elliptic curve public parameters remain hidden, even against dictionary attacks. This is made possible thanks to the existence of admissible encodings. Since each elliptic curve point can be seen as a random bit string, it enables to process bit string inside a block cipher in order to encrypt a point. The password can here be seen as a seed for computing the secret key used in the block cipher. This has a direct application within DH-EKE, because this implies that an eavesdropper cannot verify which elliptic curve parameters are used. Since eavesdroppers only see random bit strings, whatever are the elliptic curve public parameters, they cannot distinguish which ones are used.

This is a direct application of admissible encodings on elliptic curves. Moreover, we here exploit the fact that the knowledge of the elliptic curve parameters can be interpreted as a shared secret. This enables to drop out the encryption of EKE while the protocol can be still proved secure. We introduce complexity assumptions based on the discrete logarithm problem ensuring that finding one bit string which represents 2 points with known discrete logarithm from 2 different curves, is hard. Using these assumptions enables us to prove the security of our Diffie-Hellman AKE where the password gives the elliptic curve parameters.

2 Definitions

2.1 Coron-Icart Admissible encoding

We here give the definition from [10] of admissible encoding.

Definition 1. *Given 2 random variables X and Y over a set S, we say that the distribution of X and Y are (ϵ)-statistically indistinguishable if:*

$$\sum_{s \in S} |\Pr(X = s) - \Pr(Y = s)| < \epsilon.$$

Moreover, given a security parameter, two distributions are statistically indistinguishable if they are (ϵ)-statistically indistinguishable for an ϵ negligible in the security parameter.

Definition 2 (Admissible Encoding). *A function $F : S \mapsto R$ is said to be an ε-admissible encoding if:*
1. *F is computable in deterministic polynomial time;*
2. *there exists a probabilistic polynomial time algorithm $\mathcal{I}_F$ such that given $r \in R$ as input, $\mathcal{I}_F$ outputs s such that either $F(s) = r$ or $s = \perp$, and the distribution of s is ε-statistically indistinguishable from the uniform distribution in S when r is uniformly distributed in R.*

When an admissible encoding from $\{0,1\}^L$ into a curve exists, its inversion algorithm $\mathcal{I}_F$ enables to transform uniformly distributed elliptic curve points into uniformly distributed bit strings. Encrypting an elliptic curve point thus becomes easier when it is represented as a bit string. In particular, no trivial partition attack is possible.

Icart's Mapping in Characteristic 2 The equation which defines an elliptic curve $E_{a,b}$ in characteristic 2 is of general form:

$$(E_{a,b}) \qquad Y^2 + XY = X^3 + aX^2 + b$$

where a and b are elements of $\mathbb{F}_{2^n}$. For an odd n, the map $x \mapsto x^3$ is a bijection. Let

$$\begin{aligned} f_{a,b} : \mathbb{F}_{2^n} &\mapsto (\mathbb{F}_{2^n})^2 \\ u &\mapsto (x, ux + v^2) \end{aligned}$$

where $v = a + u + u^2$ and $x = (v^4 + v^3 + b)^{1/3} + v$. It is clear that, whenever computing a cube root is an exponentiation, computing $f_{a,b}$ is a deterministic polynomial time algorithm.

Lemma 1 ([11]). *Let $\mathbb{F}_{2^n}$ be a field with n odd. For any $u \in \mathbb{F}_{2^n}$, $f_{a,b}(u)$ is a point of $E_{a,b}$.*

We here focus on characteristic 2 for two reasons: the computation is simpler than in the general case, and the inverting algorithm is deterministic and quite easy to implement (cf. Section 6). Note that the Icart's encoding is not the only known general encoding for elliptic curves in characteristic 2. In [14], Shallue and van de Woestijne proposed another encoding. This encoding is not as simple, but it seems clearly possible to adapt our work to any existing admissible encoding.

Let E be an elliptic curve over a field $\mathbb{F}_{2^n}$. From such a point encoding f and a generator G of the group of points, an admissible encoding F from $\{0,1\}^{2n}$ to E can be constructed. Let l be a $2n$-bit long string. This string is split in 2 substrings $u||\lambda$ where λ is seen as a n-bit integer. We thus consider:

$$F(l) = f(u) + \lambda \cdot G \tag{1}$$

The resulting function is proved to be an admissible encoding in [10] with a negligible ϵ. The main condition on f is to be an encoding which verifies a condition weaker than in Definition 2, where the inversion algorithm needs to work only a polynomial fraction of the inputs and where the statistically indistinguishability is measured only with respect to this fraction. Such encoding is denoted in [10] a weak encoding. In the sequel, we refer to this construction as the Coron-Icart admissible encoding.

Admissible Representation We introduce below the notion of admissible representations. These representations are the outputs of $\mathcal{I}_F$, when it is applied to an elliptic curve point.

Definition 3 (Admissible Representation). *Assume that F is an admissible encoding from S to R. For any $r \in R$, we define as an admissible representation of r any output of $\mathcal{I}_F(r)$.*

An element $r \in R$ may have many different admissible representations. Furthermore, an uniformly random $r \in R$ has an uniformly random admissible representation $s \in S$. For instance, following Eq. (1) a point P of an elliptic curve admits an admissible representation of the form (u, λ) where $u||\lambda$ is a random bit string of size $2n$.

2.2 BPR Security Model

This model defines the notion of partnership, session key freshness and the security against dictionary attacks. The model considers a set of honest players who do not deviate from the protocol. The adversary controls all the network communications. This is an active adversary modeled through queries. Users can have many protocol instances running concurrently. The adversary can create, modify, or forward messages and has oracle access to the user instances.

Let A and B be two users which can be part of the key exchange protocol P. Several concurrent instances may run in different executions of P: they are denoted by A_i and B_j. The server and the user share a low-entropy secret pw uniformly drawn from a dictionary of size N.

Oracles The protocol P consists of the execution of a key exchange algorithm. It is an interactive protocol between A_i and B_j that provides the instances of A and B with a session key sk. The adversary $\mathcal{A}$ has access to the following oracles for controlling the interactions.

– Execute(A_i, B_j) simulates a passive attack where $\mathcal{A}$ eavesdrops the communication. It causes an honest execution of P between fresh instances A_i and B_j.
– Send(U_i, m) models $\mathcal{A}$ sending a message m to instance U_i ($U = A$ or B). The output is the message generated by U in processing the message m according to the protocol and the state of the instance. It simulates an active attack.
– Reveal(U_i) returns the session key of the input instance. This query models the misuse of the session key by instance U_i. The query is only available to the adversary if the targeted instance actually holds a session key (i.e. if the protocol has correctly terminated).

Security Notions The **freshness** notion captures the fact that a session key is not already directly leaked. An instance is said to be fresh in the current protocol execution if the instance has terminated and neither a Reveal query has been called on the instance nor on a partnered instance. Here two instances are defined as **partnered** if both instances have terminated correctly with the same session key.

The Test(U_i) query models the semantic security of the session key. It is available to the adversary only if the aimed instance is fresh. When called, the oracle tosses a coin b and returns the session key sk if $b = 0$ or a random value (from the domain of keys) if $b = 1$.

The **AKE security** is then defined as follows. By controlling executions of the protocol P, the adversary $\mathcal{A}$ tries to learn information on the session keys. The game is initialized by drawing a password pw from the dictionary and by letting $\mathcal{A}$ asking a polynomial number of queries. At the end of the game, $\mathcal{A}$ outputs its guess b' for the bit outputted by the Test oracle.

The AKE advantage of the adversary $\mathcal{A}$ for the key exchange protocol P is denoted

$$\mathsf{Adv}_P^{\mathsf{ake}}(\mathcal{A}) = 2\Pr[b' = b] - 1$$

Definition 4. *The protocol P is said to be* **AKE-secure** *if the adversary's advantage is negligible in the security parameter, for any polynomially bounded adversary.*

A strategy of proof consists generally in the simulation of all the oracles to be able to show that there is no leakage.

Remark 1. An oracle Corrupt is also available in this model to analyze the **forward secrecy**. When called, the adversary will obtain the player's password. For AKE with forward secrecy, the Test query should not be related to a player corrupted before the Test query. Nevertheless, corruption after the query is allowed.

2.3 Classical Assumptions

We recall the classical Computational Diffie-Hellman (CDH) assumption.

Definition 5 (Computational Diffie-Hellman Assumption). *Let E be an elliptic curve and G be a generator of a subgroup of points of prime order. Let $\mathcal{A}$ be an algorithm that:*

- *inputs two random points $P = a \cdot G$ and $Q = b \cdot G$;*
- *and outputs $R = ab \cdot G$.*

The CDH assumption ensures that the best polynomial time adversary has a negligible probability of success, when the probability is taken over P and Q.

Throughout this work, we define $\mathsf{CDH}_G(P, Q)$ to be the correct value of R. We introduce later in the paper assumptions related to this problem when the elliptic curve parameters are unknown. In particular, given two elliptic curves E_1, E_2, we rely on the hardness of finding two admissible representations l and l' such that the points $\mathsf{CDH}(F_1(l), F_1(l')) \in E_1$ and $\mathsf{CDH}(F_2(l), F_2(l')) \in E_2$ are known (with F_i an admissible encoding into E_i).

3 A New Family of Complexity Assumptions

In order to prove the strength of our protocol, we need to introduce new complexity assumptions. These assumptions arise from the fact that we are using in the same scheme different elliptic curve parameters.

Throughout this section, we use the following definitions. Let k be a security parameter. Let S be a set of $N = poly(k)$ sets of elliptic curve parameters: $\{a_i, b_i, q_i, G_i\}_{i \in [1,N]}$ over a field $\mathbb{F}_{2^n}$ (i.e. elliptic curves $E_i := E_{a_i,b_i}$ over $\mathbb{F}_{2^n}$ with a point G_i, generator of a subgroup of points of order q_i) such that:

- for each i, an admissible encoding (cf. Definition 2) exists over E_{a_i,b_i};
- q_i is a prime integer and its cofactor is 1;
- for all $i \neq j$, we have $q_i \neq q_j$.

The last point ensures that there does not exist an isomorphism between the different curves. It is important since it ensures that the discrete logarithm of a point over E_i is not related to a discrete logarithm over another E_j.

Let F_i be the admissible encoding associated to E_i. In the sequel, we mainly focus on the Coron-Icart admissible encoding obtained via Equation. (1) (Section 2.1). It ensures that an admissible representation of size $2n$ exists for almost all points.

One question arises from this setting: *Given a bit string l, is the discrete logarithm of each $P_i = F_i(l)$ in basis G_i still hard to compute?*

3.1 Hard Problems around the Discrete Logarithm of the Points P_i

Since an admissible encoding has an inversion algorithm, over each curve E_i, given a point with an unknown discrete logarithm, we can almost always (except with a negligible probability) compute one of its admissible representations and thus we have:

Lemma 2. *Assume that F_i is an admissible encoding. Computing the discrete logarithm of any $P_i = F_i(l)$ with the knowledge of l is as hard as solving the discrete logarithm problem over the curves E_i.*

When an adversary computes an admissible representation l of a point P_i over E_i, we want that for all admissible representations he can choose, his advantage on the discrete logarithm of $P_j = F_j(l)$ in basis G_j over E_j remains negligible.

Definition 6 (Admissible Encoding Twin Discrete Logarithm Assumption). *Let $\mathcal{A}$ be an algorithm that:*

- *inputs S;*
- *outputs l and a couple $(r_i, r_j) \in (\mathbb{Z}/q_i\mathbb{Z} \times \mathbb{Z}/q_j\mathbb{Z})$ such that $P_i = F_i(l) = r_i \cdot G_i$ and $P_j = F_j(l) = r_j \cdot G_j$.*

The AETDL assumption holds if any polynomial algorithm succeeds with a negligible probability, when the probability is taken over S.

Remark 2. This assumption can be expressed differently for the Coron-Icart admissible encoding of Eq. (1). Indeed, for this encoding, l is a couple of values (u, λ) such that $F_i(l) = f_i(u) + \lambda \cdot G_i$. Clearly, finding u and a couple (r_i, r_j) such that $f_i(u) = r_i \cdot G_i$ and $f_j(u) = r_j \cdot G_j$ is equivalent at solving the AETDL problem. The problem is thus to find r_1, r_2 such that $f_1^{-1}(r_1 \cdot G_1) \cap f_2^{-1}(r_2 \cdot G_2)$ is a non empty set. For a random couple $(G, G') \in E_i \times E_j$ the probability to have a u such that $f_i(u) = G$ and $f_j(u) = G'$ is at most $4 \times \frac{4}{2^n} = 2^{-(n-4)}$ for the Icart mapping (any point has at most 4 preimages through this mapping). Since the scalar multiplication is a one-way map in each E_i, it is computationally hard to find such couples.

Definition 7 (Admissible Encoding Twin Computational Diffie-Hellman Assumption). *Let $\mathcal{A}$ be an algorithm that:*

- *inputs S and l;*
- *outputs l' and a couple of points (R_i, R_j) such that $\mathsf{CDH}_{G_i}(F_i(l), F_i(l')) = R_i$ and $\mathsf{CDH}_{G_j}(F_j(l), F_j(l')) = R_j$.*

The AETCDH assumption holds if any polynomial time adversary has a negligible probability of success, when the probability is taken over S and l.

This assumption is stronger than the AETDL assumption because the AETCDH problem can be solved using the l, r_i, r_j of the AETDL assumption.

Remark 3. Due to the special form of the Coron-Icart admissible encoding, F_i can be replaced by f_i into this assumption. For this reason, the AETCDH assumption ensures that an adversary, which receives a bit string u, cannot compute u' such that over E_i and E_j he knows both $\mathsf{CDH}_{G_i}(f_i(u'), f_i(u))$ and $\mathsf{CDH}_{G_j}(f_j(u'), f_j(u))$. It is easily seen that $\mathsf{CDH}_{G_i}(f_i(u'), f_i(u)) = R_i$ implies $\mathsf{CDH}_{f_i(u)}(G_i, R_i) = f_i(u')$. The AETCDH problem is thus to find $R_i \in E_i$ and $R_j \in E_j$ such that

$$f_i^{-1}(\mathsf{CDH}_{f_i(u)}(G_i, R_i)) \cap f_j^{-1}(\mathsf{CDH}_{f_j(u)}(G_j, R_j)) \neq \emptyset$$

As above, the probability for a random couple $(G, G') \in E_i \times E_j$ to have a u' such that $f_i(u') = G$ and $f_j(u') = G'$ is at most $4 \times \frac{4}{2^n} = 2^{-(n-4)}$ for the Icart mapping. Thanks to AETDL, choosing u such that the adversary knows both logarithms of G_i in basis $f_i(u)$ and G_j in basis $f_j(u)$ is hard. Thus either the map $R_i \mapsto \mathsf{CDH}_{f_i(u)}(G_i, R_i)$ or the map $R_j \mapsto \mathsf{CDH}_{f_j(u)}(G_j, R_j)$ is one way. Consequently, it is computationally hard to find a couple (R_i,R_j).

Remark 4. The AETDL assumption is stronger than the DL assumption. Likewise, AETCDH is a stronger assumption than the CDH assumption over any elliptic curve E_i. Indeed, AETCDH is trivial if for one curve E_j CDH is an easy problem. The following algorithm illustrates this.

1. Randomly select r_i, compute $P_i = r_i \cdot G_i, R_i = r_i \cdot F_i(l)$.
2. Compute $l' = \mathcal{I}_{F_i}(P_i)$.
3. Compute $R_j = \mathsf{CDH}_{G_j}(F_j(l), F_j(l'))$ and return l', R_i, R_j.

We finally introduce a last assumption, which is the password based variant of the AETDL assumption.

Definition 8 (n-Password Based Admissible Encoding Twin Computational Diffie-Hellman Assumption). *Let P_π be a point over E_{a_π, b_π}. Let l be an admissible representation of P_π ($P_\pi = F_\pi(l)$). Let $\mathcal{A}$ be a polynomial algorithm that:*

- *inputs S and l;*
- *outputs l', $K_1, \ldots, K_n$, where each K_i is a point of one of the curves in S.*

The n-PAETCDH assumption holds if any polynomial adversary $\mathcal{A}$ has a probability $1/N + \varepsilon$ to have returned one K_i amongst n such that $\mathsf{CDH}_{G_\pi}(F_\pi(l), F_\pi(l')) = K_i$, where ε is negligible.

In this assumption, ε is the advantage of the algorithm over the value of π. Indeed, a trivial way to solve the n-PAETCDH problem is, from S and l, to randomly choose a $j \in [1, N]$ and to assume that $j = \pi$. This has a probability at least $1/N$ to succeed. Further, this assumption implies the AETCDH assumption. Indeed an algorithm $\mathcal{A}^{\mathsf{aetcdh}}$, which solves the AETCDH problem, can be transformed into an adversary which solves the n-PAETCDH problem with $\varepsilon = 1/N$. The following lemma proves that the opposite is also true.

Lemma 3. *The AETCDH assumption implies the n-PAETCDH assumption, for any n.*

Proof. Let Succ^x be the probability of success of the best adversary against the problem x. Let Event_i be the event that an algorithm outputs $i \leq n$ points $K_{j_1}, \ldots, K_{j_i}$ such that

$$\mathsf{CDH}_{G_{j_i}}(F_{j_i}(l), F_{j_i}(l')) = K_{j_i}$$

We have:

$$\Pr\left[\mathsf{Succ}^{paetcdh}\right] \leq \sum_{i=1}^{\min(N,n)} \Pr\left[\mathsf{Event}_i\right] \frac{i}{N}$$

It is easily seen that for all $i \geq 2$, $\Pr\left[\mathsf{Event}_i\right] \leq \Pr\left[\mathsf{Succ}^{aetcdh}\right]$. This leads to:

$$\begin{aligned}\Pr\left[\mathsf{Succ}^{paetcdh}\right] &\leq \Pr\left[\mathsf{Event}_1\right] \frac{1}{N} + \Pr\left[\mathsf{Succ}^{aetcdh}\right] \frac{N-1}{2} \\ &\leq \frac{1}{N} + \Pr\left[\mathsf{Succ}^{aetcdh}\right] \frac{N-1}{2}\end{aligned}$$

If $\Pr\left[\mathsf{Succ}^{aetcdh}\right]$ is negligible, since N is polynomial, then: $\Pr\left[\mathsf{Succ}^{paetcdh}\right] = \frac{1}{N} + \varepsilon$ □

4 The EC-DH-EKE Protocol with an Admissible Encoding

The Diffie-Hellman Encrypted Key Exchange (DH-EKE) protocol is roughly a DH key exchange where each data sent is encrypted by a block cipher with a key derived from a shared secret. This protocol has been introduced in [4], extended in [2], and proved in the Ideal Cipher Model and Random Oracle Model under the CDH assumption in [8]. Its basic flows are presented in

Device	parameters : E, N, G	*Reader*
password π		password π
Compute $K_{pw} = H(\pi)$		Compute $K_{pw} = H(\pi)$
Pick α		Pick β
Compute $G_1 = \alpha \cdot G$	$\xrightarrow{z_1 = \mathcal{E}_{K_{pw}}(G_1)}$	Compute $G_2 = \beta \cdot G$
	$\xleftarrow{z_2 = \mathcal{E}_{K_{pw}}(G_2)}$	
Compute $K = \alpha \cdot \mathcal{D}_{K_{pw}}(z_2)$		Compute $K = \beta \cdot \mathcal{D}_{K_{pw}}(z_1)$

Fig. 1. Basic flows of the DH-EKE scheme

Figure 1 (a complete execution, with the final authentication checks, is given in Figure 2 in our elliptic curve instantiation).

Note however that it is assumed that the ideal cipher inputs group element. Consequently, a naive implementation of the DH-EKE over elliptic curves could be insecure. Indeed, the encryption of a point $P = (x, y)$ with a key $K_{pw} = H(\pi)$ leads to a ciphertext $z = \mathcal{E}_{K_{pw}}(x||y)$. However, for any password $\pi' \neq \pi$, the decryption of z is not a point over the elliptic curve with an overwhelming probability. This leads to an offline dictionary attack (see for instance [7]).

More generally, since there exists a redundancy in the representation of $P = (x, y)$, it is difficult to encrypt P without having a dictionary attack. The encryption over the elliptic curves points should in fact be a permutation. One possibility to struggle this problem is to represent P thanks to an admissible representation. Hence applying a classical cipher would become possible.

4.1 Parameters

Let k be a security parameter. Let $\mathcal{H}$ be a hash function with $\{0,1\}^l$ as output domain. Let N be the size of $\mathcal{D}$, the dictionary of the different passwords. Let $E_{a,b}$ be an elliptic curve over $\mathbb{F}_{2^{2k+1}}$ and G be a generator of its prime order subgroup of order q, with a cofactor 1.

We assume that the protocol takes place between different devices D and a reader R. Each device possesses a password $\pi \in \mathcal{D}$.

4.2 EC-DH-EKE

The DH-EKE scheme has been proved secure in [8] in the Ideal Cipher Model and the Random Oracle Model under the CDH assumption. However, the Ideal Cipher requires to manage group elements as inputs.

Thanks to the admissible encoding, a group element can be seen as a bit-string. For this reason, a real implementation of the protocol is much more realistic because the Ideal Cipher can be replaced by a cipher such as AES-128, while an ideal cipher from elliptic curve points has still to be found. The resulting protocol is described by Figure 2.

This finally leads to an efficient and secure protocol. Additionally, the elliptic curves parameters remain hidden from an eavesdropper, since it only sees some encryption of statistically indistinguishable bit string.

Remark 5. In the masked DH-EKE variant, which is proved in [9] in the ROM only, the encryption primitive is a mask generation function instead of an ideal cipher, the Diffie-Hellman values sent are masked by addition with a full-domain hash of the password. Here a similar problem arises: the hash needs to be a ROM into elliptic curves. It is possible to use the [10]

Device		*Reader*
	parameters : $E_{a,b}, N, G$	
password π		password π
Compute $K_{pw} = H(\pi)$		Compute $K_{pw} = H(\pi)$
Pick α		Pick β
Compute $G_1 = \alpha \cdot G$		Compute $G_2 = \beta \cdot G$
Compute $l_1 = \mathcal{I}_F(G_1)$		Compute $l_2 = \mathcal{I}_F(G_2)$
	$\xrightarrow{z_1 = \mathcal{E}_{K_{pw}}(l_1)}$	
	$\xleftarrow{z_2 = \mathcal{E}_{K_{pw}}(l_2)}$	
Compute $l_2 = \mathcal{D}_{K_{pw}}(z_2)$		Compute $l_1 = \mathcal{D}_{K_{pw}}(z_1)$
Compute $K = \alpha \cdot G_2$		Compute $K = \beta \cdot G_1$
$= \alpha \cdot F(l_2)$		$= \beta \cdot F(l_1)$
Compute $\mathcal{K} = \mathcal{H}(K, z_1, z_2)$		Compute $\mathcal{K} = \mathcal{H}(K, z_1, z_2)$
Compute $K_{\mathsf{enc}} = \mathcal{H}(\mathcal{K}, 1)$		Compute $K_{\mathsf{enc}} = \mathcal{H}(\mathcal{K}, 1)$
Compute $K_{\mathsf{mac}} = \mathcal{H}(\mathcal{K}, 2)$		Compute $K_{\mathsf{mac}} = \mathcal{H}(\mathcal{K}, 2)$
Compute $T_D = \mathcal{H}(K_{\mathsf{mac}}, z_2)$		Compute $T_R = \mathcal{H}(K_{\mathsf{mac}}, z_1)$
	$\xrightarrow{T_D}$	
	$\xleftarrow{T_R}$	
Abort if T_R invalid		Abort if T_D invalid

Fig. 2. The EC-DH-EKE scheme with an Admissible Encoding

ROM construction, which is based on Admissible Encoding, to hash the password. But in that case the elliptic curves parameters will not be kept hidden as the resulting ciphertext are points on the curve.

Remark 6 (Eavesdroppers without the Elliptic Curve Parameters). The family of DH-EKE protocol is secure against offline dictionary attacks under the CDH assumption: an adversary has to compute $K = \mathsf{CDH}_G(G_1, G_2)$ to get some information on the password. Indeed, based on CDH and the ROM, the distribution of G_1, G_2, T_D, T_R is computationally indistinguishable from the uniform distribution over $E_{a,b}^2 \times \{0,1\}^{2l}$. Using this property and the property of the admissible encoding (cf. Definition 2), we know that l_1 and l_2 are bit strings computationally indistinguishable from random ones. In the Ideal Cipher Model, this implies that the z_1, z_2, T_D, T_R are indistinguishable as well. For this reason, an adversary who does not know the elliptic curve parameters, cannot compute them, even if he has a list of curves parameters.

5 Our Proposal of Password Based EC-DH Key Exchange without Encryption

In the EC-DH-EKE scheme (Figure 2), we use the admissible representation in order to encrypt properly over elliptic curves. As an additional benefit, this protocol also ensures the privacy of the elliptic curve parameters. Following this last idea, we modify further the EC-DH-EKE protocol in order to base the authentication directly on the knowledge of the elliptic curve parameters instead of the knowledge of an additional password.

Our proposal is similar to our EC-DH-EKE variant: points are represented by an admissible representation but we did not encrypt the representations anymore. Since the distribution of l_1, l_2, T_D, T_R is computationally indistinguishable from the uniform distribution, exchanging these values in clear makes no difference from an eavesdropper point of view. This enables to avoid the use of an ideal cipher in the security analysis. In the sequel, we denote our scheme EC-DH-ARKE which stands for Elliptic Curve Diffie-Hellman Admissibly Represented Key Exchange.

In our scheme, the dictionary of passwords becomes a set of different elliptic curves parameters indexed by a table.

5.1 Parameters

Let k be a security parameter and N a polynomial integer in k. Let $\mathcal{H}_0, \mathcal{H}_1, \mathcal{H}_2$ be 3 hash functions with $\{0,1\}^l$ as output domain. Let $\mathbb{F}_{2^n}$ be a field such that there exist efficient admissible encodings and such that $2^n = \mathcal{O}(2^{2k})$. Let $S = \{a_i, b_i, G_i, q_i\}_{i\in[1,N]}$ be a set of elliptic curve parameters such that G_i is a generator of the prime order group of $E_i = E_{a_i,b_i}$ of order q_i. It is assumed that the cofactor is 1 for each group of points. We also assume that the prime integers q_i are pair-wise distinct. This last condition is sufficient to ensure that no isomorphism exists between any couple (E_i, E_j).

5.2 The EC-DH-ARKE Protocol

During the initialization phase, each reader receives the set S as input and each device receives one element of S as parameters. It can further define its own public discrete logarithm based pair of public/secret keys with these parameters. The index i related to these parameters is given to the device owner. We stress that the set S does not need to remain secret. We use the index in order to enable a user to typeset data related to the parameter.

At the beginning of each authentication, the device holder has to typeset one index and then the reader verifies that the index corresponds to the elliptic curve parameters used by the device. The protocol is illustrated in Figure 3.

Device		*Reader*
password : $E_{a_\pi,b_\pi}, q_\pi, G_\pi$		password : $E_{a_\pi,b_\pi}, q_\pi, G_\pi$
Pick α		Pick β
Compute $G_1 = \alpha \cdot G_\pi$		Compute $G_2 = \beta \cdot G_\pi$
Compute $l_1 = \mathcal{I}_{F_\pi}(G_1)$		Compute $l_2 = \mathcal{I}_{F_\pi}(G_2)$
	$\xrightarrow{l_1}$	
	$\xleftarrow{l_2}$	
Compute $K = \alpha \cdot G_2$		Compute $K = \beta \cdot G_1$
$= \alpha \cdot F_\pi(l_2)$		$= \beta \cdot F_\pi(l_1)$
Compute $\mathcal{K} = \mathcal{H}_0(K, l_1, l_2)$		Compute $\mathcal{K} = \mathcal{H}_0(K, l_1, l_2)$
Compute $K_{\mathsf{enc}} = \mathcal{H}_1(\mathcal{K}, 1)$		Compute $K_{\mathsf{enc}} = \mathcal{H}_1(\mathcal{K}, 1)$
Compute $K_{\mathsf{mac}} = \mathcal{H}_1(\mathcal{K}, 2)$		Compute $K_{\mathsf{mac}} = \mathcal{H}_1(\mathcal{K}, 2)$
Compute $T_D = \mathcal{H}_2(K_{\mathsf{mac}}, l_2)$		Compute $T_R = \mathcal{H}_2(K_{\mathsf{mac}}, l_1)$
	$\xrightarrow{T_D}$	
	$\xleftarrow{T_R}$	
Abort if T_R invalid		Abort if T_D invalid

Fig. 3. Our proposal EC-DH-ARKE

5.3 Security Result

Our proposal is secure in the Random Oracle Model under the AETCDH assumption. More concretely:

Theorem 1 (AKE security). *Let S be a randomly chosen set of N elliptic curve parameters as above. Let π be an uniformly chosen index in $[1, N]$. Let $\mathcal{A}$ be an adversary in the BPR model against the AKE security of our scheme within a time T, with less than q_s interactions with the parties, q_p eavesdroppings and q_h hash queries. We have:*

$$\mathsf{Adv}^{\mathsf{ake}}_{\mathsf{EC-DH-ARKE}}(\mathcal{A}) \leq q_s \mathsf{Succ}^{q_h-\mathsf{paetcdh}}(T') + q_p \mathsf{Succ}^{\mathsf{cdh}}(T') + \varepsilon$$

where $T' = T + \mathcal{O}(Q^2)$, where $Q = q_s + q_h + q_p$ and ε is negligible if q_s, q_p and q_h are polynomial in k.

The security of this protocol relies on two ideas:

1. a passive eavesdropper does not get any information on the exchanged data whenever the CDH is a hard problem for any curve in S;
2. an active adversary can find the password by an online dictionary attack with a probability $1/N$. In fact, an adversary can always be turned into an algorithm, which solves the PAETCDH problem with almost the same probability.

5.4 Security Proof

We use a sequence of game in order to prove the security of the protocol. In the sequel $\Pr[G_i]$ denotes the probability in the game G_i that the adversary outputs the good guess $b' = b$.
Game G_0: This is the real security game. A set of N parameters is chosen, the device receives one element of S and the reader receives the same, while the set S is given to the adversary. The reader and the device act as described in Figure 3. We assume that $\mathcal{H}_0, \mathcal{H}_1, \mathcal{H}_2$ are random oracles into l bit strings.

Once the Test query is sent, following a randomly chosen bit b, the key K_{enc} or a random string is returned. Hence

$$\mathsf{Adv}^{\mathsf{ake}}_{\mathsf{EC-DH-ARKE}}(\mathcal{A}) = |\Pr[G_0] - 1/2|$$

Game G_1: We simulate the device and the reader for each query to the Send, Execute, Test and Reveal oracles, as the real players would do. We also simulate the random oracles $\mathcal{H}_0, \mathcal{H}_1, \mathcal{H}_2$. This does not change the adversary advantage but modifies the duration of the simulation because of the necessary table lookups. We thus have $T'[G_1] = T + \mathcal{O}(Q^2)$.

Game G_2: We abort the simulation if a collision occurs while simulating one of the random oracles. A collision occurs with a probability $Q^2/2^{l+1}$ for each random oracle. We thus have:

$$|\Pr[G_2] - \Pr[G_1]| \leq 5 \times \frac{Q^2}{2^{l+1}}$$

From this game, we are sure that the values of $\mathcal{K}$ are different. This property is also true for $K_{\mathsf{enc}}, K_{\mathsf{mac}}, T_D, T_R$.

Game G_3: We simulate the Execute oracle using random values. To distinguish this game from the previous one, the adversary needs to solve the CDH problem over a curve for at least one couple (l_1, l_2) exchanged during one of the Execute queries. For this reason, we have:

$$|\Pr[G_3] - \Pr[G_2]| \leq q_p \mathsf{Succ}^{cdh}(T')$$

Game G_4: We abort the simulation when we get a collision on elliptic curve points chosen at the beginning. Since there are q_π points in the curve, we have that:

$$|\Pr[G_4] - \Pr[G_3]| \leq \frac{Q^2}{q_\pi}$$

From this game, we know that the admissible representations returned by the simulation are pair-wise distinct.

Game G_5: We abort when one triplet $(\mathsf{CDH}_{G_\pi}(F_\pi(l_1), F_\pi(l_2)), l_1, l_2)$ is queried a second time to $\mathcal{H}_0$, while l_1, l_2 are values exchanged during one instance I initiated by the query Send. When this second query to $\mathcal{H}_0$ occurs, we know that the adversary knows the value $\mathcal{K}$ of the instance I. We have assumed that the adversary does not make two identical queries to any random oracle.

Before this abortion, the adversary does not have any advantage over the password π since he has observed random values (due to the admissible representations property, see Section 2.1) that he could not verify, without querying $\mathcal{H}_0$ with a correct query. Once this event arises, we determine l, the value amongst l_1 or l_2, which is the value that we have sent while we simulated the Send query. We then get all the triplets queried to the random oracle $\mathcal{H}_0$ by the adversary, which contains l. These triplets form an answer to the PAETCDH problem. For this reason we have:

$$|\Pr[G_5] - \Pr[G_4]| \leq q_s \mathsf{Succ}^{q_h-\mathsf{paetcdh}}(T')$$

Game G_6: We do not use neither $\mathcal{H}_1$ nor $\mathcal{H}_2$ anymore when we simulate the execution of the protocol. For this reason, the Reveal query does not give any information such as the Send query concerning T_R and T_D. We thus have:

$$\Pr[G_6] = 1/2$$

Furthermore, since the adversary does not have computed $\mathcal{K}$ for any instance, there is no difference from the adversary point of view between G_5 and G_6. □

This result holds in the forward secrecy setting (cf. Remark 1) as well.

6 Implementation

We here describe how to implement the inversion map $\mathcal{I}_F$ for F defined by Eq. (1) thanks to the Icart mapping. We focus on this point, since the other computations are straightforward. We recall the complete algorithm to compute $\mathcal{I}_F$, this algorithm is described by [10] and it uses the algorithm Inv which is explained in the next section. q is the group order of the elliptic curve group of points.

Algorithm $\mathcal{I}_f$
Input: $P \in E_{a,b}$
Output: $u \in \mathbb{F}_{2^n}$ such that $f_{a,b}(u) = P$ or $u = \perp$

1. Compute the set $U = f_{a,b}^{-1}(P)$ using algorithm Inv
2. Let $\delta_P = |U|/4$
3. With probability $1 - \delta_P$ return $\perp$
4. Return a random element u in U.

Algorithm $\mathcal{I}_F$
Input: $P \in E_{a,b}$
Output: $(u, \lambda) \in \mathbb{F}_{2^n} \times \mathbb{Z}_q$ such that $P = F(u, \lambda) = f_{a,b}(u) + \lambda.G$, or $\perp$

1. For $i = 1$ to $T = -k/\log(1 - 2^{n-2}/q)$:
   (a) Randomly chooses $\lambda \in \mathbb{Z}_N$ and computes $Z = \lambda.G$
   (b) Let $X = P - Z \in \mathbb{G}$
   (c) Compute $a = \mathcal{I}_f(X)$
   (d) If $a \neq \perp$, return (a, λ)
2. Return $\perp$.

6.1 Computing a Preimage with the Algorithm Inv

Inverting the Icart's mapping [11] in characteristic 2 is possible by computing the roots of a degree 4 polynomial. Given a point $P = (x, y)$ on an elliptic curve $E_{a,b}$ of equation $(X^3 + aX^2 + b = Y(X+Y))$, we know that u is a preimage of P if and only if $y + a^2 + ux + u^2 + u^4 = 0$. One can remark that this equation is $\mathbb{F}_2$ linear. For this reason, finding its roots requires to solve a linear system over $\mathbb{F}_2$. The matrix related to the linear function $u \mapsto u^4 + u^2 + ux$ is easy to compute. Solving a linear system can be done thanks to Gaussian elimination. This operation requires $\mathcal{O}(n^3)$ binary operations. Furthermore, over a platform with registers of size w, the running time is $\mathcal{O}(n^3/w)$. Moreover, the inverting algorithm Inv is thus deterministic.

To compute the inverse of the admissible encoding, it is not necessary to compute the set U of solutions but only its cardinality, which is the number of roots of the equation $y + a^2 + ux + u^2 + u^4 = 0$. One possibility is to compute each time the matrix and its row echelon. However, a more clever way is to compute the greatest common divisor of $P(U) = y + a^2 + Ux + U^2 + U^4$ with $U^{2^n} - U$. This does not change the complexity but it reduces the memory requirement for the algorithm, from n^2 bits to $4n$ bits.

Remark 7. One can remark that the polynomial $U^{2^n} \mod P$ is necessarily a $\mathbb{F}_2$ linear polynomial. Furthermore, it can be expressed thanks to a polynomial expression in x, y and a^2.

6.2 The Overall Running Time

In the algorithm $\mathcal{I}_F$, the process has to be repeated at most T. If one wants a deterministic algorithm, one can run exactly $T = -k/\log(1-\alpha)$ times the inversion process, where $\alpha = 2^{n-2}/q$. However, in the general case, a deterministic algorithm is not necessary. The average number of steps of the probabilistic algorithm is $1/\alpha$. This average running time could leak information on the elliptic curve parameter since $\alpha = 2^{n-2}/q$. However, since we have assumed that the cofactor of each curve is 1, we know that q is near from 2^n thanks to the Hasse' bound. This ensures that for two different elliptic curves E_1, E_2, we have

$$\left| \frac{1}{\alpha_1} - \frac{1}{\alpha_2} \right| \leq 2^{4-n/2}$$

Hence it requires an exponential number of observations to distinguish the running times $1/\alpha_1$ and $1/\alpha_2$.

To summarize, the running of the algorithm is $\mathcal{O}(n^3/w)$, while the memory requirement is $\mathcal{O}(n^2)$, leading to the feasibility of practical implementations.

6.3 Minimizing the Communication Cost

It is possible to reduce the size of the exchanged data whenever the admissible encoding is the one defined by Equation (1), Section 2.1. Indeed, instead of sending $l = (u||\lambda)$ in the protocol, both participants can send u only. Assume that the device receives u_2, it then computes $f_\pi(u_2)$ and to get the key, it computes $(\alpha - \lambda_1) \cdot f_\pi(u_2)$. It is easily seen that $f_\pi(u_2) = (\beta - \lambda_2) \cdot G_\pi$, thus $K = (\alpha - \lambda_1)(\beta - \lambda_2) \cdot G_\pi$ for the device. The reader, from u_1 can also compute this key, which is the seed for the future session key.

This simple trick reduces the total amount of data exchanged during the protocol and thus makes it more efficient. However, this trick is not general and can only be use with some particular admissible encoding. We remark that this enables to send only 1 element in $\mathbb{F}_{2^n}$ instead of 2 for a classical representation of an elliptic curve point. It is in fact less than the classical compressed representation (one $\mathbb{F}_{2^n}$ element and one additional bit).

7 Conclusion and Further Work

This paper describes a new and efficient Password-Based Authenticated Key Exchange protocol which is especially adapted for elliptic curves settings. Particularly, it enables to keep the elliptic curves parameters hidden. In the context of contactless ID documents, this opens a way for implementing realistic solutions which preserve privacy of owners's nationality. As extensions of this work, a good perspective is to continue on adaptations of enhanced versions of EKE which are analyzed on a more general model (e.g. UC) or with standard assumptions (standard model).

An implementation in a PKI context with the property that the curve parameters remain hidden is also a possible application of our idea. For instance, a smartcard could contain a certified public key which is stored directly in its admissible representation.

References

1. Michel Abdalla, Pierre-Alain Fouque, and David Pointcheval. Password-based authenticated key exchange in the three-party setting. In Serge Vaudenay, editor, *Public Key Cryptography*, volume 3386 of *Lecture Notes in Computer Science*, pages 65–84. Springer, 2005.
2. Mihir Bellare, David Pointcheval, and Phillip Rogaway. Authenticated key exchange secure against dictionary attacks. In *EUROCRYPT*, pages 139–155, 2000.
3. Mihir Bellare and Phillip Rogaway. The AuthA protocol for password-based authenticated key exchange. In *IEEE P1363*, pages 136–3, 2000.
4. Steven M. Bellovin and Michael Merritt. Encrypted key exchange: Password-based protocols secure against dictionary attacks. In *IEEE Symposium on Research in Security and Privacy*, pages 72–84, 1992.
5. Steven M. Bellovin and Michael Merritt. Augmented encrypted key exchange: A password-based protocol secure against dictionary attacks and password file compromise. In *ACM Conference on Computer and Communications Security*, pages 244–250, 1993.
6. Dan Boneh and Matthew K. Franklin. Identity-based encryption from the weil pairing. In Joe Kilian, editor, *CRYPTO*, volume 2139 of *Lecture Notes in Computer Science*, pages 213–229. Springer, 2001.
7. Colin Boyd, Paul Montague, and Khanh Quoc Nguyen. Elliptic curve based password authenticated key exchange protocols. In Vijay Varadharajan and Yi Mu, editors, *ACISP*, volume 2119 of *Lecture Notes in Computer Science*, pages 487–501. Springer, 2001.
8. Emmanuel Bresson, Olivier Chevassut, and David Pointcheval. Security proofs for an efficient password-based key exchange. In Sushil Jajodia, Vijayalakshmi Atluri, and Trent Jaeger, editors, *ACM Conference on Computer and Communications Security*, pages 241–250. ACM, 2003.
9. Emmanuel Bresson, Olivier Chevassut, and David Pointcheval. New security results on encrypted key exchange. In Feng Bao, Robert H. Deng, and Jianying Zhou, editors, *Public Key Cryptography*, volume 2947 of *Lecture Notes in Computer Science*, pages 145–158. Springer, 2004.
10. Jean-Sébastien Coron and Thomas Icart. A random oracle into elliptic curves. Cryptology ePrint Archive, Report 2009/340, 2009. `http://eprint.iacr.org/`.
11. Thomas Icart. How to hash into an elliptic-curve. In Shai Halevi, editor, *CRYPTO*, volume 5677 of *Lecture Notes in Computer Science, Security and Cryptology*, pages 303–316. Springer, 2009.
12. David P. Jablon. Strong password-only authenticated key exchange. *ACM Computer Communications Review*, 26:5–26, 1996.
13. Sarvar Patel. Number theoretic attacks on secure password schemes. In *IEEE Symposium on Security and Privacy*, pages 236–247. IEEE Computer Society, 1997.
14. Andrew Shallue and Christiaan van de Woestijne. Construction of rational points on elliptic curves over finite fields. In Florian Hess, Sebastian Pauli, and Michael E. Pohst, editors, *ANTS*, volume 4076 of *Lecture Notes in Computer Science*, pages 510–524. Springer, 2006.
15. Markus Ullmann, Dennis Kugler, Heike Neumann, Sebastian Stappert, and Matthias Vogeler. Password authenticated key agreement for contactless smart cards. *RFIDSec*, 2008.