Some Comments on the Security of ECIES with SECP256K1

Hartwig Mayer
{hartwig.mayer}@coinfabrik.com

CoinFabrik

June 22, 2016

1 Introduction

In this article we attempt to concisely discuss some security aspects of the widely used encryption scheme ECIES – Elliptic Curve Integrated Encryption Scheme – with the SECP256K1 curve as it is currently implemented in Ethereum. This encryption scheme is a hybrid scheme, meaning that it establishes a shared secret key which is used in a symmetric algorithm for both encryption and decryption. Our focus here will be on the security aspect of ECIES during the key exchange procedure which takes place on an open network.
The security of ECIES assumes the intractability of a variant of the decisional Diffie-Hellman problem. This is one of the standard mathematical problems in elliptic curve cryptography (ECC) along with the discrete logarithm problem and the computational Diffie-Hellman problem. We will review how these problems are related to each other and the approaches that are known to solve the decisional Diffie-Hellman problem. We illustrate the different methods of attack with the SECP256K1 curve implementation of ECIES, the curve recommended by the SEC Group and which is currently used in Bitcoin and Ethereum. A quick overview of the SECP256K1 curve can be found for example in [BL13] or [May16].

2 The ECIES Encryption Scheme

The ECIES is the elliptic curve version of the encryption scheme proposed by M. Abdalla, M. Bellare, and P. Rogaway in [ABR01]. As mentioned in the introduction, it is a hybrid public key encryption scheme, and employs the Diffie-Hellman protocol over elliptic curves to establish an encryption key.

2.1. *Some Background on Elliptic Curves.* Let $\mathbb{F}_q$, $q = p^n$ a prime power, be a finite field, e.g., $\mathbb{F}_p = \mathbb{Z}/p\mathbb{Z}$ for $n = 1$. An elliptic curve (in short affine Weierstrass form) E

over $\mathbb{F}_q$ is the set of solutions of an equation

$$E \ : \ y^2 = x^3 + Ax + B \qquad (A, B \in \mathbb{F}_q)$$

satisfying the condition $4A^3 + 27B^2 \neq 0$ to ensure that the curve E is smooth. When we consider the field $\mathbb{F}_q$ for the coordinates, the set is given by

$$E(\mathbb{F}_q) := \{(x, y) \in \mathbb{F}_q \times \mathbb{F}_q \mid y^2 = x^3 + Ax + B\}.$$

The fact that a line L in the (x, y)-plane intersects E in three points allows us to define an operation '$P_1 + P_2$' $\in E(\mathbb{F}_q)$ for any P_1, P_2 on $E(\mathbb{F}_q)$, with properties similar to $+$ in the integers $\mathbb{Z}$. The group defined by the curve E and this operation forms the basis for elliptic curve cryptography (see e.g. [Was08] for more informations). A convenient notation in this context is to define the *scalar multiplication* by

$$aP := \underbrace{P + \ldots + P}_{a-\text{times}} \in E(\mathbb{F}_q) \qquad \big(a \in \mathbb{Z}, P \in E(\mathbb{F}_q)\big).$$

In this paper, we will work with the following assumption:

Set-up: *Let $E/\mathbb{F}_q$ be an elliptic curve over $\mathbb{F}_q$ with $q = p^n$. We assume that the order of the group $E(\mathbb{F}_q)$ is*

$$\#E(\mathbb{F}_q) = r$$

for some prime r.

2.2. *The Scheme of ECIES.* (Reference [BSS05], chapter I and III). The ECIES public-key encryption scheme is standardized in ANSI X9.63, ISO/IEC 15946-3, IEEE P1363a, NESSIE, NSA SUITE B, and SEC 1 v2. Ethereum follows the version specified in SEC 1 v2, section 5.1 (see [Res09]).
ECIES uses three ingredients, and there are many choices for each one (see Ethereum's choices here).

(*a*) a key derivation function KDF : $E(\mathbb{F}_q) \longrightarrow \{0,1\}^l$ mapping a point Z to a bit string of length l. A key KDF(Z) is divided further into two strings k_1, k_2 such that KDF$(Z) = (k_1||k_2)$ where $||$ denotes concatenation. k_1 is used as master key for symmetric encryption and k_2 for authentication. (Ethereum uses, if unspecified, PBKDF2 with hash function SHA256 as key derivation function).

(*b*) a symmetric key encryption scheme which includes an encryption and a decryption function $\text{ENC}_{k_1}/\text{DEC}_{k_1}$ (Ethereum uses AES128/256).

(*c*) a message authentication code $\text{MAC}_{k_2} : \{0,1\}^* \longrightarrow \{0,1\}^*$ to prevent man-in-the-middle attacks (Ethereum uses HMAC with hash function SHA256).

Suppose we have two users, U_1 and U_2, and U_1 wants to send a message m (an arbitrary bit string) to user U_2. Therefore, they agree publicly on an elliptic curve $E/\mathbb{F}_q$ and a base point $P \in E(\mathbb{F}_q)$ of order r. They make their choices for ingredients (a), (b), and (c). User U_2 chooses a secret $a \in \mathbb{Z}$ as his private key and publishes his public key $Q = aP$. Then:

Encryption:

I. U_1 chooses a random integer $k \in \{1, \ldots, r-1\}$.

II. U_1 computes $R = kP$ and $Z = kQ$.

III. U_1 derives $(k_1||k_2) \leftarrow \mathrm{KDF}(Z)$, where $||$ denotes concatenation.

IV. U_1 computes $C = \mathrm{ENC}_{k_1}(m)$ and $t = \mathrm{MAC}_{k_2}(C)$. User U_1 then sends

$$(R, C, t).$$

Decryption:

I. U_2 computes $Z = aR$.

II. U_2 derives $(k_1||k_2) \leftarrow \mathrm{KDF}(Z)$.

III. U_2 computes $t' = \mathrm{MAC}_{k_2}(C)$. If $t' \neq t$ then U_2 rejects the ciphertext.

IV. U_2 computes $m = \mathrm{DEC}_{k_1}(C)$.

We see that the secret key Z shared by U_1 and U_2 is generated by the Diffie-Hellman public key exchange protocol ($Z = a(kP) = k(aP)$). Note that some variants of ECIES use point compression, i.e., the algorithm continues only with the x-coordinate of $Z = (x_Z, y_Z)$. This leads to smaller key size, but steps must be taken to eliminate the resulting malleability problems.

2.3. *The Provable Security of ECIES.* Encryption schemes should be capable of resisting an indistinguishable adaptive chosen ciphertext attack (IND-CCA) in the standard model in order to be called safe. In an IND-CCA attack, the attacker can decrypt ciphertexts and learn about the algorithm. The attacker gives two plaintext messages m_0 and m_1 to a *challenger* who chooses one at random and returns its encryption. The attacker can also consult the decryption oracle about more texts. She then has to guess whether the challenger has given her the encryption of message m_0 or m_1. If the attacker cannot do this with a significantly higher probability than $1/2$, the encryption scheme is called IND-CCA secure. The mathematically precise definition of this security notion can be found e.g. in [BSS05], chapter III, or [HK10]. ECIES is IND-CCA secure in the standard model if:

(a) The hashed decisional Diffie-Hellman assumption holds true on the elliptic curve E. (Please see the following section for details).

(b) The security of the symmetric encryption model is guaranteed.

(c) The security of the authentication MAC scheme is guaranteed.

The security of the symmetric encryption scheme and the MAC scheme is beyond the scope of this paper. We focus on the hashed decisional Diffie-Hellman problem on elliptic curves.

3 Security of ECIES

A signature scheme is secure if no one can sign for someone else. It is not as easy to determine when an encryption scheme is secure. If a ciphertext cannot be decrypted completely but allows one to guess the general topic it is talking about, should this be called 'breaking the scheme' or not? Mathematical problems can be used to model the different levels of security. The most common mathematical assumptions in security proofs in ECC are listed in Table 1. In this table, the expression '$\mathcal{A}$ (adversary) has no advantage' basically means that the probability of an attacker solving or deciding the corresponding problem correctly in polynomial time is negligible (see [HK10] for a precise definition).

Discrete Logarithm Assumption (EC-DL)	*$\mathcal{A}$ has no advantage in solving: Given $P, Q \in E(\mathbb{F}_q)$, find $a \in \mathbb{Z}$, such that $Q = aP$.*
Computational Diffie-Hellman assumption (EC-CDH)	*$\mathcal{A}$ has no advantage in solving: Given $aP, bP \in E(\mathbb{F}_q)$, $a, b \in \mathbb{Z}$, compute abP.*
Hashed Diffie-Hellman assumption (EC-HDH)	*$\mathcal{A}$ has no advantage in deciding: Given $aP, bP \in E(\mathbb{F}_q)$, $a, b \in \mathbb{Z}$, and α an l-bit string. Decide whether $\alpha = \mathrm{KDF}(abP)$.*
Decisional Diffie-Hellman assumption (EC-DDH)	*$\mathcal{A}$ has no advantage in deciding: Given aP, bP, and $cP \in E(\mathbb{F}_q)$ on E with $a, b, c \in \mathbb{Z}$. Decide whether $cP = abP$.*

Table 1: Conventional assumptions in security proofs against an adversary $\mathcal{A}$.

3.1. *Relationships Between Conventional Assumptions in ECC.* To understand the complexity of the assumptions in Table 1, lets think a bit about their implications. For example, EC-CDH implies EC-DL, because, if you can solve discrete logs, you can also solve the computational Diffie-Hellman problem: *Choose aP and solve the discrete logarithm to get a. Then compute $a(bP) = abP$.* And similarly, both EC-DDH and EC-HDH imply EC-CDH, and if the key derivation function is a cryptographic hash function, EC-DDH implies EC-HDH (see Figure 1).

It is more difficult to go in the opposite direction. For example does being able to solve the computational Diffie-Hellman problem in polynomial time imply that one can solve discrete logarithm problems in polynomial time? In [MW99] Maurer and Wolf were able to prove that it is possible to go from DLP to CDH in polynomial time for most cryptographic groups. Maurer and Wolf's algorithm is applicable for most of the elliptic curves recommended by the SEC Group, including the secp256k1 curve, as demonstrated in [MSV04] and [Ben05].
Joux and Nguyen show in [JN03] that an elliptic curve that is safe for EC-CDH will not

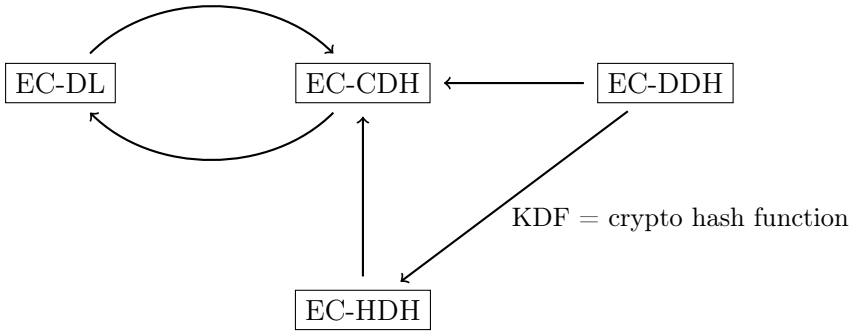

Figure 1: Implications for **most** elliptic curves. Read '$X \to Y$' as 'assumption X implies assumption Y'.

necessarily be safe for EC-DDH (though in the generic group model these assumptions are equivalent; see [Bon98]). The remaining two implications do not hold (see [GKR04]).

Conclusion: The decisional Diffie-Hellman assumption (used in encryption schemes) is stronger than the discrete logarithm assumption (used in signatures). The hashed decisional Diffie-Hellman assumption (ECIES) depends on the choice of the hash function which is used for KDF. If KDF is a cryptographic hash function, it is weaker than the decisional Diffie-Hellman assumption; it is always stronger than the discrete logarithm assumption.

3.2. *Security of ECIES with* SECP256K1. One strategy to solve the hashed decisional Diffie-Hellman problem (and break ECIES) is to solve the discrete logarithm problem. And, almost all known solutions for the decisional Diffie-Hellman problem on elliptic curves rely on discrete logarithms (see [GG16], p. 3). The only exceptions are some cases in which the Weil pairing can be used to solve the decisional Diffie-Hellman problem directly. To make the second approach work one needs the *embedding degree* of the elliptic curve to be 0 or 1 (see [Ver04]). Since the embedding degree of the SECP256K1 curve is much greater than 1 (see e.g. [BL13]), it is not vulnerable to this type of attack. Those who are interested in the math behind this attack can see Appendix A. The SECP256K1 curve satisfies the EC-DL assumption, as discussed in [May16], which makes it safe against the first strategy.

Conclusion: In Ethereum, the ECIES is implemented with the SECP256K1 curve. Although the security of ECIES is based on EC-HDH, it is enough to assume EC-DL for this choice of curve. For the SECP256K1 curve the EC-DL assumption holds (see e.g. [May16]). To read more about attacks targeting implementation issues with SECP256K1 please see Bernstein and Lange's website [BL13] or our previous paper [May16].

4 Further Remarks

Modern cryptography seeks encryption schemes which are both efficient and secure, and require as few assumptions as possible. It is difficult to satisfy all three of these conditions. We mention three efficient schemes (secure in the standard model) below:

- Boyen-Mei-Waters 2005 (only secure under the bilinear Diffie-Hellman assumption)
- Kiltz 2007 (under the Gap HDH assumption)
- Hanaoka-Kurosawa 2010 (under the intermediate HDH assumption)

Less efficient schemes but secure in the standard model under weaker assumptions:

- Cash-Kiltz-Shoup 2008 (under the CDH assumption)
- Haralambiev-Jager-Kiltz-Shoup 2010 (under the CDH assumption)

See for example [HK10] for a more detailed overview of this research field.

A Attack with Weil Pairing

One attack on the decisional Diffie-Hellman problem uses the Weil pairing. The r-torsion points $E[r]$ of an elliptic curve E are all points on E which satisfy $rP = P + \ldots + P = O$, where O denotes the neutral element. The coordinates of these points are not necessarily defined over $\mathbb{F}_q$, but they are in a field extension $\mathbb{F}_{q^m}$ of $\mathbb{F}_q$. The smallest integer m satisfying $E[r] \subseteq E(\mathbb{F}_{q^m})$ is the *embedding degree*. The Weil pairing is a non-degenerate bilinear map

$$e_r : E[r] \times E[r] \longrightarrow \mathbb{F}_{q^m}.$$

What is key here is the bilinearity of the Weil pairing which we can exploit to solve the decisional Diffie-Hellman problem as follows: given aP, bP, and cP on E. To decide whether or not $abP = cP$ one could try the following: since

$$e_r(aP, bP) = e_r(P, bP)^a = e_r(P, P)^{ab} \qquad \text{and} \qquad e_r(P, cP) = e_r(P, P)^c$$

it suffices to check whether

$$e_r(aP, bP) = e_r(P, cP), \tag{1}$$

but only if $e_r(P, P) \neq 1$ which is not the case for $P \in E(\mathbb{F}_q)$. The solution to this problem is to use an endomorphism $\varphi : E \longrightarrow E$ with the property $e_r(P, \varphi(P)) \neq 1$ and replace bP and cP in the second argument of the Weil pairing in (1) by $b\varphi(P)$ and $c\varphi(P)$, respectively. The endomorphism φ is called a distortion map. Note that one does not even have to solve a DLP in $\mathbb{F}_{q^m}$! Such maps exist for ordinary elliptic curves only if the embedding degree is less than or equal to 1 (see [Ver04] or [GR04]) .

References

[ABR01] M. Abdalla, M. Bellare, and P. Rogaway, *The oracle Diffie-Hellman assumptions and an analysis of DHIES.*, Topics in cryptology - CT-RSA 2001. The cryptographer's track at RSA conference 2001, San Francisco, CA, USA, April 8–12, 2001. Proceedings, Berlin: Springer, 2001, pp. 143–158.

[Ben05] K. Bentahar, *The equivalence between the DHP and DLP for elliptic curves used in practical applications, revisited.*, Cryptography and coding. 10th IMA international conference, Cirencester, UK, December 19–21, 2005. Proceedings., Berlin: Springer, 2005, pp. 376–391.

[BL13] D. J. Bernstein and T. Lange, *Safecurves: choosing safe curves for elliptic-curve cryptography*, `http://safecurves.cr.yo.to` (2013).

[Bon98] D. Boneh, *The decision Diffie-Hellman problem.*, Algorithmic number theory. 3rd international symposium, ANTS-III, Portland, OR, USA, June 21–25, 1998. Proceedings, Berlin: Springer, 1998, pp. 48–63.

[BSS05] I. F. Blake, G. Seroussi, and N. P. Smart (eds.), *Advances in elliptic curve cryptography.*, Cambridge: Cambridge University Press, 2005.

[GG16] S. Galbraith and P. Gaudry, *Recent progress on the elliptic curve discrete logarithm problem*, Designs, Codes and Cryptography **78** (2016), 51–72.

[GKR04] R. Gennaro, H. Krawczyk, and T. Rabin, *Secure hashed Diffie-Hellman over non-DDH groups.*, Advances in cryptology – EUROCRYPT 2004., Berlin: Springer, 2004, pp. 361–381.

[GR04] S. Galbraith and V. Rotger, *Easy decision Diffie-Hellman groups.*, LMS J. Comput. Math. **7** (2004), 201–218.

[HK10] G. Hanaoka and K. Kurosawa, *Between hashed dh and computational dh: Compact encryption from weaker assumption*, 2010, hanaoka-goichiro@aist.go.jp 14632 received 22 Jan 2010.

[JN03] A. Joux and K. Nguyen, *Separating decision Diffie-Hellman from computational Diffie-Hellman in cryptographic groups.*, J. Cryptology **16** (2003), no. 4, 239–247.

[May16] H. Mayer, *ECDSA in Bitcoin and Etherum: a Research Survey*, `http://blog.coinfabrik.com/author/hartwig--mayer/` (2016).

[MSV04] A. Muzereau, N.P. Smart, and F. Vercauteren, *The equivalence between the DHP and DLP for elliptic curves used in practical applications.*, LMS J. Comput. Math. **7** (2004), 50–72 (English).

[MW99] U.M. Maurer and S. Wolf, *The relationship between breaking the Diffie-Hellman protocol and computing discrete logarithms.*, SIAM J. Comput. **28** (1999), no. 5, 1689–1721.

[Res09] Certicom Research, *SEC 1: Elliptic Curve Cryptography version 2.0.*, `http://www.secg.org/sec1--v2.pdf`.

[Ver04] E. R. Verheul, *Evidence that XTR is more secure than supersingular elliptic curve cryptosystems.*, J. Cryptology **17** (2004), no. 4, 277–296.

[Was08] L. C. Washington, *Elliptic Curves: Number Theory and Cryptography*, Discrete Mathematics and Its Applications, Chapman and Hall/CRC, 2008.