# Validation of Elliptic Curve Public Keys

Adrian Antipa[1], Daniel Brown[1], Alfred Menezes[2],
René Struik[1], and Scott Vanstone[2]

[1] Certicom Research, Canada
{aantipa,dbrown,rstruik}@certicom.com
[2] Dept. of Combinatorics and Optimization, University of Waterloo, Canada
{ajmeneze,savansto}@uwaterloo.ca

**Abstract.** We present practical and realistic attacks on some standardized elliptic curve key establishment and public-key encryption protocols that are effective if the receiver of an elliptic curve point does not check that the point lies on the appropriate elliptic curve. The attacks combine ideas from the small subgroup attack of Lim and Lee, and the differential fault attack of Biehl, Meyer and Müller. Although the ideas behind the attacks are quite elementary, and there are simple countermeasures known, the attacks can have drastic consequences if these countermeasures are not taken by implementors of the protocols. We illustrate the effectiveness of such attacks on a key agreement protocol recently proposed for the IEEE 802.15 Wireless Personal Area Network (WPAN) standard.

## 1 Introduction

The purpose of public key validation, as enunciated by Johnson [16, 17], is to verify that a public key possesses certain arithmetic properties. Public key validation is especially important in Diffie-Hellman protocols where an entity $B$ derives a shared secret $k$ by combining his private key with a public key received from $A$, and subsequently uses $k$ in some symmetric-key protocol (e.g., encryption or message authentication) with $A$. A dishonest $A$ might select an invalid public key in such a way that the use of $k$ reveals information about $B$'s private key. Lim and Lee [20] demonstrated the importance of public key validation by presenting so-called *small subgroup attacks* on some discrete logarithm key agreement and encryption protocols that are effective if the receiver of a group element does not verify that the element belongs to the desired group of high order (e.g., a subgroup of prime order $q$ of $\mathbb{Z}_p^*$).

Although public key validation has become recognized as prudent practice, many cryptographic standards do not mandate that it be performed. In this paper, we present attacks on some standardized elliptic curve key establishment and public-key encryption protocols that are effective if the receiver of an elliptic curve point does not check that the point lies on the appropriate elliptic curve. We argue with considerable care that, despite their simplicity, the attacks are practical and realistic. We illustrate their effectiveness on a key agreement

Y.G. Desmedt (Ed.): PKC 2003, LNCS 2567, pp. 211–223, 2003.

protocol recently proposed for the IEEE 802.15 WPAN standard. The attacks provide further evidence for the necessity of performing public key validation.

The remainder of this paper is organized as follows. Section 2 describes several standardized elliptic curve cryptographic schemes that will be used to demonstrate the attacks. Validation of elliptic curve public keys is defined in Section 3. The invalid-curve attacks are presented and analyzed in Section 4. Some countermeasures are proposed in Section 5. Finally, we draw our conclusions in Section 6.

## 2 Elliptic Curve Cryptographic Schemes

We present some elliptic curve schemes that have been included in several standards and draft standards. The schemes are presented in sufficient detail to convince the reader that the assumptions made in our attacks are plausible, and that the attacks can indeed be a significant threat in practice.

In any public-key cryptographic system, the entities may share common data called *domain parameters*, and they have *key pairs* each consisting of a *public key* and a corresponding *private key*. A key pair may be *static* (long-term) if it is intended to be used for an extended period of time, or *ephemeral* (short-term) if it is only intended to be used for a single run of a protocol.

*Domain parameters.* For elliptic curve cryptographic schemes, the domain parameters $D$ include the following:

1. The *order* $q$ of the underlying finite field $\mathbb{F}_q$.
2. An indication of the *representation* used for elements of $\mathbb{F}_q$ (e.g., the irreducible reduction polynomial if the field has characteristic 2 and a polynomial basis representation is used).
3. The *defining equation* of the elliptic curve $E$ over $\mathbb{F}_q$.
4. A *base point* $P = (x_P, y_P) \in E(\mathbb{F}_q)$ of prime order.
5. The *order* $n$ of $P$.
6. The *cofactor* $h = \#E(\mathbb{F}_q)/n$.

We assume throughout this paper that elliptic curve domain parameters $D$ have been selected so that the elliptic curve discrete logarithm problem resists all known attacks, and that $n^2$ does not divide $\#E(\mathbb{F}_q)$ whence $\langle P \rangle$ is the unique subgroup of $E(\mathbb{F}_q)$ having order $n$. Examples of such parameters are the *NIST domain parameters* specified in the FIPS 186-2 standard [10]. We shall assume that all entities have an authentic copy of $D$.

*Key pairs.* A user $A$ now selects $w_A \in_R [1, n-1]$ and computes $W_A = w_A P$. $A$'s static key pair is $(W_A, w_A)$, where $W_A$ is the static public key and $w_A$ is the static private key. $B$'s static key pair is denoted $(W_B, w_B)$. We assume that $A$ and $B$ can obtain authentic copies of each others static public keys, e.g., via certificates.

*Notation and terminology.* If a discrete logarithm protocol takes place in a subgroup $G_1$ of prime order of a group $G_2$, then $G_1$ is called the *main group*, while $G_2$ is called the *supergroup*. For example, in the elliptic curve setting, $E(\mathbb{F}_q)$ is the supergroup while $\langle P \rangle$ is the main group. The point at infinity is denoted by $\infty$, $H$ denotes a cryptographic hash function, and $x(R)$ denotes the $x$-coordinate of a point $R$. In the ECMQV protocol, if $R$ is a finite point then $\overline{R}$ is defined to be the integer $(\overline{x} \bmod 2^{\lceil f/2 \rceil}) + 2^{\lceil f/2 \rceil}$ where $\overline{x}$ is the integer representation of $x(R)$, and $f = \lfloor \log_2 n \rfloor + 1$.

### 2.1 One-Pass ECDH

One-pass ECDH is a basic elliptic curve Diffie-Hellman protocol that combines the sender's ephemeral public key and the receiver's static public key. Although it provides very limited authentication, it might be useful in scenarios where only unilateral authentication is needed, e.g., in the widely deployed SSL/TLS protocol (see the ECDH_ECDSA protocol in [11]) and in SMIME (see [8]). ECDH is fully specified in ANSI X9.63 [4] and IEEE 1363-2000 [12].

1. $A$ selects $r_A \in_R [1, n-1]$, and computes $R_A = r_A P$, $K = r_A W_B$ and $k = H(x(K))$. $A$ sends $R_A$ to $B$.
2. $B$ computes $K = w_B R_A$ and $k = H(x(K))$.
3. The shared secret key is $k$.

### 2.2 ECIES

The elliptic curve integrated encryption scheme (ECIES) is due to Bellare and Rogaway [6] who proposed the scheme in the general setting of a group of prime order. ECIES has been included in several standards and draft standards including ANSI X9.63 [4], IEEE P1363a [13], and ISO/IEC 15946-3 [15]. It can be used to transport a session key (to be used subsequently in some symmetric-key protocol) or to transmit a confidential message of arbitrary length. ECIES uses a hash function $H$, a message authentication algorithm MAC, and a symmetric encryption scheme SYM. Abdalla, Bellare and Rogaway [1] proved that ECIES is semantically secure against adaptive chosen-ciphertext attacks under some variants of the computational Diffie-Hellman assumption, and the assumptions that MAC and SYM are secure.

*Encryption.* To send a message $m$ to $B$, $A$ does:

1. Select $r_A \in_R [1, n-1]$ and compute $R_A = r_A P$ and $K = r_A W_B$.
2. Derive symmetric keys $(k_1, k_2)$ from $H(x(K))$.
3. Compute $c = \mathrm{SYM}_{k_1}(m)$ and $t = \mathrm{MAC}_{k_2}(c)$.
4. Send $(R_A, c, t)$ to $B$.

*Decryption.* To decrypt $(R_A, c, t)$, $B$ does:

1. Compute $K = w_B R_A$.
2. Derive symmetric keys $(k_1, k_2)$ from $H(x(K))$.
3. Compute $t' = \mathrm{MAC}_{k_2}(c)$ and reject the ciphertext if $t' \neq t$.
4. Compute $m = \mathrm{SYM}_{k_1}^{-1}(c)$.

### 2.3 One-Pass ECMQV

The one-pass ECMQV key agreement protocol [18] differs from one-pass ECDH and ECIES in that it combines the static key of the receiver with both the ephemeral and static keys of the sender. The protocol is specified in the ANSI X9.63 [4], IEEE 1363-2000 [12] and ISO 15946-3 [15] standards.

1. $A$ selects $r_A \in_R [1, n-1]$, computes $R_A = r_A P$, and sends this to $B$.
2. $A$ computes $s_A = (r_A + \overline{R}_A w_A) \bmod n$ and $K = hs_A(W_B + \overline{W}_B W_B)$. If $K = \infty$, then $A$ terminates the protocol run with failure; otherwise $A$ computes $k = H(x(K))$.
3. $B$ computes $s_B = (w_B + \overline{W}_B w_B) \bmod n$ and $K = hs_B(R_A + \overline{R}_A W_A)$. If $K = \infty$, then $B$ terminates the protocol with failure; otherwise $B$ computes $k = H(x(K))$.
4. The shared secret key is $k$.

### 2.4 ECDSA

The elliptic curve digital signature algorithm (ECDSA) is specified in the ANSI X9.62 [3], IEEE 1363-2000 [12], FIPS 186-2 [10] and ISO 15946-2 [14] standards.

*Signature generation.* To sign a message $m$, $A$ does:

1. Select $k \in_R [1, n-1]$ and compute $r = x(kP)$. (Check that $r \neq 0$.)
2. Compute $e = H(m)$ and $s = k^{-1}(e + w_A r) \bmod n$. (Check that $s \neq 0$.)
3. $A$'s signature on $m$ is $(r, s)$.

*Signature verification.* To verify $A$'s signature $(r, s)$ on $m$, $B$ does:

1. Reject the signature if $r \notin [1, n-1]$ or if $s \notin [1, n-1]$.
2. Compute $e = H(m)$, $u_1 = s^{-1} e \bmod n$ and $u_2 = s^{-1} r \bmod n$.
3. Compute $V = u_1 P + u_2 W_A$ and $v = x(V) \bmod n$.
4. Accept the signature iff $v = r$.

## 3 Public Key Validation

Validation of an elliptic curve public key $W$ ensures that $W$ is a point of order $n$ in $E(\mathbb{F}_q)$, where $\mathbb{F}_q$, $E$ and $n$ are specified by the associated domain parameters.

**Definition 1** A point $W = (x_W, y_W)$ (static or ephemeral) associated with a set of domain parameters $D$ is *valid* if the following four conditions are satisfied:

1. $W \neq \infty$.
2. $x_W$ and $y_W$ are properly represented elements of $\mathbb{F}_q$ (e.g., integers in the interval $[0, q-1]$ if $\mathbb{F}_q$ has prime order).
3. $W$ satisfies the defining equation of the elliptic curve $E$.
4. $nW = \infty$.

If any one of these conditions is violated, then $W$ is *invalid*.

There may be ways of verifying condition 4 of Definition 1 that are much faster than performing an expensive point multiplication $nW$. For example, if $h = 1$ (which is usually the case for elliptic curves over prime fields that are used in practice), then condition 4 is implied by the other three conditions. In some protocols the check that $nW = \infty$ may either be embedded in the protocol computations or replaced by the check that $hW \neq \infty$ (which guarantees that $W$ is not in a small subgroup of $E(\mathbb{F}_q)$ of order dividing $h$).

*Small subgroup attacks.* Lim and Lee [20] presented attacks on some discrete logarithm key agreement and encryption protocols to demonstrate the importance of checking that group elements received from another entity belong to the main group, and not to some small subgroup of the supergroup. Their attacks are effective if the cofactor $h$ (the index of the main group in the supergroup) has many small factors—the attacker can then determine the victim's static key modulo these small factors and combine the results using the Chinese remainder theorem. This is often the case in the ordinary discrete logarithm setting where the main group is a subgroup of prime order $q$ of the multiplicative group $\mathbb{Z}_p^*$. In practice, one may have $q \approx 2^{160}$ and $p \approx 2^{1024}$, and the cofactor $h = (p-1)/q$ may have many small factors.

In the elliptic curve setting, the cofactor $h$ is typically very small, (e.g., $h \in \{1, 2, 4\}$ for the 15 elliptic curves in FIPS 186-2 [10]). In this case, the small subgroup attacks are not effective in determining private keys since an adversary has very limited choices for small subgroup elements and therefore can learn at most a few bits of the victim's private key.

*Differential fault analysis.* Biehl, Meyer and Müller [7] presented several differential fault attacks [9] on elliptic curve cryptographic schemes. The main observation in their attacks is that the usual formulae for adding points (in either affine coordinates or in projective coordinates) on an elliptic curve given by the general Weierstrass equation

$$y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6 \tag{1}$$

do not involve $a_6$. Similarly, the addition formulae given in IEEE 1363-2000 [12] for elliptic curves over prime fields with reduced equation

$$y^2 = x^3 + ax + b \tag{2}$$

and for non-supersingular elliptic curves over characteristic two finite fields with reduced equation

$$y^2 + xy = x^3 + ax^2 + b \tag{3}$$

do not involve $b$. Thus the addition formulae are the same for two curves of the form (2) (or of the form (3)) whose defining equations have the same $a$ coefficient but different $b$ coefficients. In the attack, the adversary sends a point $Q$ that has small order $l$ on some elliptic curve whose defining equation has a different coefficient $b$ (but the same coefficient $a$) as the victim's elliptic curve. Now, if the victim does not check whether $Q$ is a point on his elliptic curve, then the victim would proceed to compute $wQ$ where $w$ is her private key. Since $wQ$ has order dividing $l$, subsequent use of $wQ$ may reveal $w \bmod l$ to the adversary. By repeating the attack with points $Q$ of orders $l$ that are pairwise relatively prime, the adversary can eventually recover $w$ by the Chinese Remainder Theorem. Biehl, Meyer and Müller [7] described their attack on the basic ElGamal encryption scheme which was already known to be insecure against active attacks.

## 4 Invalid-Curve Attacks

In this section, we combine the small subgroup attack of Lim and Lee and the differential fault attack of Biehl, Meyer and Müller to obtain attacks that are effective on the one-pass ECDH, ECIES, and one-pass ECMQV protocols if the receiver of an elliptic curve point does not verify that the point does indeed lie on the elliptic curve specified by the domain parameters. In essence, the attacks use the observations of Biehl, Meyer and Müller to extend the small subgroup attacks to elliptic curves different from the one specified by the domain parameters. We call these attacks *invalid-curve attacks*. We illustrate the effectiveness of such attacks on a key agreement protocol that was recently proposed for the IEEE 802.15 WPAN standard.

The invalid-curve attacks we are going to describe fail if the receiver of a point $W$ checks that $nW = \infty$ by performing a point multiplication operation. We argue, however, that it is plausible that an implementor may have elected not to perform this operation. First, in the case where the cofactor is small, it is not known how small subgroup attacks on the one-pass ECDH, ECIES, and one-pass ECMQV protocols can be effectively mounted to determine static private keys and hence the check $nW = \infty$ might have been omitted. Second, an implementor may have elected to verify that $nW = \infty$ using a faster method, e.g., simply checking that $W \neq \infty$ in the case that $h = 1$ (and neglected to check that $W$ is on the curve). Indeed, none of the ANSI X9.63, IEEE 1363-2000, IEEE P1363a, ISO 15946-2 and ISO 15946-3 standards mandate that public key validation be performed. ANSI X9.63 does mandate that *some* form of public

key validation be performed, but this can include simply receiving the assurance (in some unspecified way) that the owner generated the public key itself using trusted routines; hence *explicit* public key validation as specified in Definition 1 is not mandated. We note, however, that some standards such as ANSI X9.63 specify a conversion routine from octet strings to elliptic curve points that explicitly verifies that the recovered point is on the elliptic curve. Thus, if public keys are represented using octet strings and these conversion routines are used, then the invalid-curve attacks we are going to describe are thwarted.

**Definition 2** Let $E$ be an elliptic curve defined over the finite field $\mathbb{F}_q$ with defining equation (1) in Weierstrass form. Then an *invalid curve* (relative to $E$) is an elliptic curve $E'$ defined over $\mathbb{F}_q$ whose Weierstrass equation differs from $E$'s only in the coefficient $a_6$.

Note that $E(\mathbb{F}_q) \cap E'(\mathbb{F}_q) = \{\infty\}$. If $Q \in E'(\mathbb{F}_q)$ and $Q \neq \infty$, then a private key $w$ such that $Q = wP$ does not exist. We assume henceforth that the addition formulae used for $E$ do not involve $a_6$. Hence, if $Q \in E'(\mathbb{F}_q)$, $t$ is an integer, and the addition formulae for $E$ are used in any point multiplication algorithm to compute $R = tQ$, then $R$ is indeed equal to $tQ$ as points in $E'(\mathbb{F}_q)$.

### 4.1 Invalid-Curve Attack on One-Pass ECDH

Suppose that one-pass ECDH is used by $A$ to establish a shared secret $k$ with $B$, and that $k$ is subsequently used by $B$ to send messages authenticated with a message authentication algorithm MAC to $A$. $A$ selects an invalid curve $E'$ such that $E'(\mathbb{F}_q)$ contains a point $Q$ of small order $l$, and sends $Q$ to $B$. $B$ computes $K = w_B Q$ and $k = H(x(K))$. Later, when $B$ sends $A$ a message $m$ and its tag $t = \mathrm{MAC}_k(m)$, $A$ can determine the correct $K$ up to sign[1] by finding a $K' \in \langle Q \rangle$ satisfying $t = \mathrm{MAC}_{k'}(m)$, where $k' = H(x(K'))$. Since $Q$ has order $l$, the expected number of trials before $A$ succeeds is $l/2$, whereafter $A$ has determined $w_l = \pm w_B \bmod l$. Hence $A$ knows that $w_l^2 \equiv w_B^2 \pmod{l}$. By repeating the attack with points $Q$ (on perhaps different invalid curves) having orders that are pairwise relatively prime, $A$ can eventually recover $z = w_B^2 \bmod N$ for some $N > n^2$ by the Chinese Remainder Theorem. Since $w_B^2 < n^2 < N$, we have $z = w_B^2$, and hence $A$ can compute $w_B = \sqrt{z}$. Observe that $B$ is unaware that the attack has taken place.

In many applications, such as email, the sender $A$ will send its ephemeral public key $R_A$ together with a message encrypted or authenticated with $k$. In this scenario, an invalid-curve attack similar to the one on ECIES described in Section 4.2 can be mounted.

### 4.2 Invalid-Curve Attack on ECIES

The attack on ECIES is somewhat more complicated than the attack on one-pass ECDH since the attacker $A$ has to demonstrate possession of the shared

[1] Recall that two points on an elliptic curve have the same $x$-coordinate if and only if they are negatives of each other.

secret point $K$ by producing the proper MAC tag $t$ on the ciphertext $c$. As with the attack on ECDH, $A$ selects a point $Q$ of order $l$ on an invalid curve $E'$. $A$ then makes a guess $w_l \in [0, l-1]$ for $w_B \bmod l$ and computes $K = w_l Q$ (instead of $K = r_B W_A$). $A$ transmits $Q$ (instead of $R_A$) to $B$, who computes $K' = w_B Q$. With overwhelming probability, the key $k_2'$ derived from $K'$ satisfies $t = \mathrm{MAC}_{k_2'}(c)$ if and only if $w_l \equiv \pm w_B \pmod{l}$. If $A$ is able to determine whether or not $B$ accepted the ciphertext, then $A$ is expected to determine $\pm w_B \bmod l$ after about $l/4$ iterations. As before, the attack can be repeated to recover $w_B$. Unlike the case of one-pass ECDH, the victim $B$ may be aware than an invalid-curve attack on ECIES is being launched if he notices that he is receiving many invalid ciphertexts from $A$.

We comment that this invalid-curve attack on ECIES does not contradict its provable security since the protocol and proof in [1] assume that received points are always in the main group.

### 4.3 Invalid-Curve Attack on One-Pass ECMQV

As with one-pass ECDH, suppose that one-pass ECMQV is used by $A$ to establish a shared secret $k$ with $B$, and that $k$ is subsequently used by $B$ to send messages authenticated with a message authentication algorithm MAC to $A$. The attack on one-pass ECMQV is more complicated than the attack on one-pass ECDH since the victim $B$ uses $A$'s static and ephemeral public keys to derive the shared secret.

$A$ selects an invalid curve $E'$ such that $E'(\mathbb{F}_q)$ contains a point $Q_A$ of small prime order $l$ with $\gcd(l, h) = 1$. $A$ selects $Q_A$ as its static public key and obtains a certificate for it (see below). $A$ next selects $T_A \in \langle Q_A \rangle$, $T_A \neq \infty$, such that $T_A + \overline{T}_A Q_A \neq \infty$, and sends $T_A$ to $B$ (instead of $R_A$). Since $T_A + \overline{T}_A Q_A \in \langle Q_A \rangle$, the point $K$ that $B$ computes is also in $\langle Q_A \rangle$. (If $K = \infty$ and $B$ terminates the protocol then it must be the case that $s_B \equiv 0 \pmod{l}$.) As with the attack on one-pass ECDH, $A$ can deduce $\pm s_B \bmod l$. Repeating the attack gives $s_B$ and $w_B = s_B(1 + \overline{W}_B)^{-1} \bmod n$.

*Certifying invalid public keys.* Suppose that $A$ wishes to have a point $Q$ of (small) order $l$ that is on an invalid curve $E'$ certified by a Certification Authority (CA). In practice, as dictated by PKI standards such as [2, Section 2.3] and [22, Section 4], the CA does not validate $Q$. Rather, the CA performs a proof of possession (POP) of a private key test whereby $A$ has to submit a signature generated with respect to $Q$ on some message $m$ of a predetermined format (perhaps chosen by the CA), and the signature is thereafter verified by the CA. We show that if the signature scheme used is ECDSA (Section 2.4), then $A$ is able to generate a signature on $m$ that is accepted by the CA. $A$ does the following:

1. Select arbitrary $s, u_2' \in [1, n-1]$.
2. Compute $e = H(m)$ and $u_1 = s^{-1} e \bmod n$.
3. Compute $T_1 = u_1 P \in E(\mathbb{F}_q)$, $T_2 = u_2' Q \in E'(\mathbb{F}_q)$, $V = T_1 + T_2$, and $r = x(V)$. (Note that $V$ is computed using the addition formulae for $E$, and is in neither $E(\mathbb{F}_q)$ nor $E'(\mathbb{F}_q)$.)

4. Compute $u_2 = s^{-1}r \bmod n$.
5. If $u_2 \not\equiv u_2' \pmod{l}$ then go to step 1.
6. Output the signature $(r, s)$.

A straightforward heuristic argument shows that the expected number of iterations of the main loop before $A$ terminates is about $l$. The CA will accept the signature since $u_2 \equiv u_2' \pmod{l}$ and hence $u_2Q = u_2'Q$.

### 4.4 Analysis

We first note that the restriction that $a$ is fixed in equations (2) and (3) is without much loss of generality since approximately half of all isomorphism classes of elliptic curves over $\mathbb{F}_q$ will have a representative with the given $a$ coefficient. Since the orders of elliptic curves over $\mathbb{F}_q$ are almost uniformly distributed among the admissible orders in the Hasse interval $[q+1-2\sqrt{q}, q+1+2\sqrt{q}]$ (see [19]), we can expect to quickly find an elliptic curve $E$ where $\#E(\mathbb{F}_q)$ is divisible by a specified small prime $l$. Table 1 lists some elliptic curves defined over the prime field $\mathbb{F}_p$, where $p = 2^{192} - 2^{64} - 1$, that have points of small prime order $l$. This field is the smallest of the five prime fields recommended by NIST [10]. The curves in Table 1, and also the NIST curve over $\mathbb{F}_p$, all have $a = -3$ in their defining equations (2).

Let $p_i$ denote the $i$th prime number, and suppose that the attacker uses points of orders $l = p_1, p_2, p_3, \ldots, p_s$. The attacker needs $N > n^2 \approx q^2$, so $s$ should be the smallest positive integer such that

$$N = \prod_{i=1}^{s} p_i > q^2.$$

Let $T = \sum_{i=1}^{s} p_i/4$. Then the attack on one-pass ECDH requires about $s$ interactions (partial protocol runs) with the victim, and about $2T$ MAC computations by the attacker. The attack on ECIES requires about $T$ interactions with the victim. The attack on one-pass ECMQV requires certification of about $s$ invalid public keys, $s$ interactions with the victim, and about $2T$ MAC computations by the attacker. The parameters $s$ and $T$ for the five NIST prime fields in FIPS 186-2 are presented in Table 2.

*Optimizations of the invalid-curve attacks.* By selecting larger primes $p_i$, the attacker in the one-pass ECDH and one-pass ECMQV scenarios can decrease the number of interactions with the victim at the expense of an increased number of MAC computations. The number of interactions with the victim can also be decreased by using elliptic curves having points whose orders are divisible by powers of small primes. Consider, for example, an implementation of ECIES using the NIST elliptic curve over $\mathbb{F}_p$ where $p = 2^{521} - 1$. Now, if $p$ is a prime satisfying $p \equiv 3 \pmod 4$, and if $a$ is a quadratic residue modulo $p$, then it is known that the (supersingular) elliptic curve

$$E : y^2 = x^3 + ax$$

**Table 1.** Elliptic curves $E : y^2 = x^3 - 3x + b$ over $\mathbb{F}_p$ where $p = 2^{192} - 2^{64} - 1$ having points of small prime order $l$

| $b$ | $l$ | $\#E(\mathbb{F}_p)$ |
|---|---|---|
| 7 | 17 | 6277101735386680763835789423152579575769728003507816940176 |
| 8 | 29 | 6277101735386680763835789423134254969503970241284573199712 |
| 12 | 7, 103 | 6277101735386680763835789423330235564423778417310050448358 |
| 20 | 11 | 6277101735386680763835789423241857284589896488855666951129 |
| 21 | 53,83 | 6277101735386680763835789423280657011091039330875664478827 |
| 24 | 109 | 6277101735386680763835789423338875685246211614336097240571 |
| 25 | 2,3,23,79 | 6277101735386680763835789423122144658648828167097184768836 |
| 28 | 131 | 6277101735386680763835789423268748257445578040429701148868 |
| 30 | 127 | 6277101735386680763835789423210775104109650069520558949736 |
| 31 | 37, 107 | 6277101735386680763835789423204441525191330010353698233123 |
| 32 | 43 | 6277101735386680763835789423258617709088937800879867650520 |
| 34 | 19 | 6277101735386680763835789423250038640299327488215705493600 |
| 35 | 41 | 6277101735386680763835789423231000109287096932320643629080 |
| 39 | 31 | 6277101735386680763835789423184749274510329350288277847727 |
| 40 | 5, 47 | 6277101735386680763835789423362437408900078197305511428080 |
| 42 | 97 | 6277101735386680763835789423100126242413526789909842794735 |
| 43 | 71 | 6277101735386680763835789423348140981991719874891656011856 |
| 59 | 13, 59 | 6277101735386680763835789423214305021449488084661039666632 |
| 68 | 101 | 6277101735386680763835789423152535895119425594340719306505 |
| 69 | 73 | 6277101735386680763835789423316686344024991599507825887793 |
| 74 | 89 | 6277101735386680763835789423207521611646327664801082254174 |
| 82 | 151 | 6277101735386680763835789423205224777875334437065750522006 |
| 104 | 139 | 6277101735386680763835789423352646215168136324998796285949 |
| 107 | 113 | 6277101735386680763835789423073056697678061416724633187404 |
| 119 | 137 | 6277101735386680763835789423312863477238674670518689754010 |
| 142 | 61 | 6277101735386680763835789423320128694158340103743426909958 |
| 166 | 67 | 6277101735386680763835789423259116488526354652819809060300 |
| 201 | 149 | 6277101735386680763835789423174207583130851129966653984609 |

over $\mathbb{F}_p$ has order $p + 1$ and $E(\mathbb{F}_p)$ is cyclic [21, Example 2.18]. For the case of the NIST prime $p = 2^{521} - 1$, we have $p \equiv 3 \pmod 4$ and $a = -3$ is a quadratic residue modulo $p$. Thus the group of $\mathbb{F}_p$-rational points on

$$E : y^2 = x^3 - 3x$$

is cyclic of order $2^{521}$. This group can be used in the invalid-curve attack with the attacker iteratively querying the victim with points of order $2, 4, 8, \ldots$. Only about 521 interactions with the victim are needed, versus the 11548 interactions in Table 2.

**Table 2.** Attack parameters for the five NIST prime fields $\mathbb{F}_p$

| prime $p$ | $s$ | $T$ |
|---|---|---|
| $2^{192} - 2^{64} - 1$ | 61 | 1996 |
| $2^{224} - 2^{96} + 1$ | 68 | 2548 |
| $2^{256} - 2^{224} + 2^{192} + 2^{96} - 1$ | 76 | 3275 |
| $2^{384} - 2^{128} - 2^{96} + 2^{32} - 1$ | 105 | 6735 |
| $2^{521} - 1$ | 134 | 11548 |

### 4.5 Invalid-Curve Attack on a Key Agreement Protocol Proposed to IEEE 802.15

IEEE 802.15 is a working group developing standards for Wireless Personal Area Networks (WPANs) for short-distance wireless networks. A WPAN can be comprised of as many as 256 *devices*, one of which is designated by the devices as the *controller*. One of the controller's tasks is to consider requests from new devices to enter the network. If the controller approves the request, then it securely transports data keys to the device. The data keys can then be used by the device to securely communicate with other devices in the WPAN.

Bailey, Singer and Whyte [5] proposed the following key agreement protocol based on ECIES for this purpose. The elliptic curve chosen is the NIST curve over the prime field $\mathbb{F}_p$ with $p = 2^{256} - 2^{224} + 2^{192} + 2^{96} - 1$. Suppose that each legitimate device has a certificate for its elliptic curve public key. If a device $A$ wishes to enter the network, it transports a session key $k_A$ to the controller $B$ using ECIES (and $B$'s public key). Similarly, the controller transports a session key $k_B$ to $A$ using ECIES (and $A$'s public key). Both devices now derive two shared keys from $k_A$ and $k_B$, one of which is used in a key confirmation stage and the other to transport data keys.

The proposal [5] explicitly mentions that public key validation is optional. If validation if not performed, then a rogue device $A$, using some other device's public key certificate, can launch the invalid-curve attack for ECIES on the controller. Since $B$ terminates the run of the key agreement protocol if ciphertext received from $A$ is not valid, $A$ easily learns whether its guess $w_l$ for $w_B \bmod l$ is correct. The roughly 3275 (see Table 2) interactions required with $B$ is quite feasible given that the controller is expecting frequent interactions with the many other devices in the network. We conclude that the attack is very effective in this scenario.

## 5 Preventing Invalid-Curve Attacks

The simplest way to prevent the invalid-curve attacks is to check that a received point does indeed lie on the legitimate elliptic curve.

There are many other techniques that can potentially guard against the invalid-curve attacks. For example, one may use formulas for the addition law

that use both coefficients $a$ and $b$ of the equation of the legitimate elliptic curve—it is then unlikely that the addition law is valid for any other elliptic curve. The invalid-curve attacks may also fail on some classes of elliptic curves that use special, faster forms of point multiplication. For example, the fast point multiplication algorithms [23] for Koblitz curves (elliptic curves whose coefficients belong to $\mathbb{F}_2$) repeatedly apply the Frobenius map $\tau : (x, y) \mapsto (x^2, y^2)$. If $Q \in E'(\mathbb{F}_{2^m})$ where $E'$ is not a Koblitz curve, then $\tau(Q)$ is generally not in $E'(\mathbb{F}_{2^m})$. Hence the fast point multiplication algorithms with inputs an integer $k$ and $Q \in E'(\mathbb{F}_{2^m})$ generally will not compute $kQ$.

## 6 Conclusions

We have presented invalid-curve attacks on some elliptic curve key establishment and public-key encryption protocols. The simplest and most effective way to prevent these attacks is to check that a received point $Q$ indeed lies on the right elliptic curve. Preferably, the receiver should perform a full validation on $Q$. The attacks reinforce the importance of performing validation on public keys in protocols where a public key is combined with the receiver's static private key.

## References

[1] M. Abdalla, M. Bellare and P. Rogaway, "The oracle Diffie-Hellman assumptions and an analysis of DHIES", *Topics in Cryptology—CT-RSA 2001*, Lecture Notes in Computer Science, vol. 2020 (2001), 143-158. 213, 218

[2] C. Adams and S. Farrell, *Internet X.509 Public Key Infrastructure: Certificate Management Protocols*, RFC 2510, March 1999. Available from `http://www.ietf.org`. 218

[3] ANSI X9.62, *Public Key Cryptography for the Financial Services Industry: The Elliptic Curve Digital Signature Algorithm (ECDSA)*, American National Standards Institute, 1999. 214

[4] ANSI X9.63, *Public Key Cryptography for the Financial Services Industry: Key Agreement and Key Transport using Elliptic Curve Cryptography*, American National Standards Institute, 2001. 213, 214

[5] D. Bailey, A. Singer and W. Whyte, "IEEE P802-15_TG3 NTRU full security text proposal", submission to the IEEE P802.15 Working Group for Wireless Personal Area Networks (WPANs), April 22, 2002. Available from `http://grouper.ieee.org/groups/802/15/pub/2002/May02/02210r0P802-15_TG3-NTRU-Full-Security-Text-Proposal.pdf`. 221

[6] M. Bellare and P. Rogaway, "Minimizing the use of random oracles in authenticated encryption schemes", *Information and Communications Security*, Lecture Notes in Computer Science, vol. 1334 (1997), 1-16. 213

[7] I. Biehl, B. Meyer and V. Müller, "Differential fault analysis on elliptic curve cryptosystems", *Advances in Cryptology—CRYPTO 2000*, Lecture Notes in Computer Science, vol. 1880 (2000), 131-146. 215, 216

[8] S. Blake-Wilson, D. Brown and P. Lambert, *Use of Elliptic Curve Cryptography (ECC) Algorithms in Cryptographic Message Syntax (CMS)*, RFC 3278, April 2002. Available from `http://www.ietf.org`. 213

[9] D. Boneh, R. DeMillo and R. Lipton, "On the importance of checking cryptographic protocols for faults", *Advances in Cryptology—EUROCRYPT '97*, Lecture Notes in Computer Science, vol. 1233 (1997), 37-51. 215

[10] FIPS 186-2, *Digital Signature Standard (DSS)*, Federal Information Processing Standards Publication 186-2, National Institute of Standards and Technology, 2000. 212, 214, 215, 219

[11] V. Gupta, S. Blake-Wilson, B. Moeller and C. Hawk, *ECC Cipher Suites for TLS*, IETF Internet-Draft, August 2002. Available from `http://www.ietf.org`. 213

[12] IEEE Std 1363-2000, *IEEE Standard Specifications for Public-Key Cryptography*, 2000. 213, 214, 216

[13] IEEE P1363a, *Draft Standard Specifications for Public-Key Cryptography — Amendment 1: Additional Techniques*, working draft 10.5, April 26 2002. Available from `http://grouper.ieee.org/groups/1363/tradPK/P1363a/draft.html`. 213

[14] ISO/IEC 15946-2, *Information Technology — Security Techniques — Cryptographic Techniques Based on Elliptic Curves — Part 2: Digital Signatures*, draft, February 2001. 214

[15] ISO/IEC 15946-3, *Information Technology — Security Techniques — Cryptographic Techniques Based on Elliptic Curves — Part 3: Key Establishment*, draft, February 2001. 213, 214

[16] D. Johnson, Contribution to ANSI X9F1 working group, 1997. 211

[17] D. Johnson, "Key validation", Contribution to IEEE P1363 working group, 1997. 211

[18] L. Law, A. Menezes, M. Qu, J. Solinas and S. Vanstone, "An efficient protocol for authenticated key agreement", *Designs, Codes and Cryptography*, to appear. 214

[19] H. Lenstra, "Factoring integers with elliptic curves", *Annals of Mathematics*, 126 (1987), 649-673. 219

[20] C. Lim and P. Lee, "A key recovery attack on discrete log-based schemes using a prime order subgroup", *Advances in Cryptology—CRYPTO '97*, Lecture Notes in Computer Science, vol. 1294 (1997), 249-263. 211, 215

[21] A. Menezes, *Elliptic Curve Public Key Cryptosystems*, Kluwer Academic Publishers, 1993. 220

[22] M. Myers, C. Adams, D. Solo and D. Kemp, *Internet X.509 Certificate Request Message Format*, RFC 2511, March 1999. Available from `http://www.ietf.org`. 218

[23] J. Solinas, "Efficient arithmetic on Koblitz curves", *Designs, Codes and Cryptography*, 19 (2000), 195-249. 222