Elliptic Curve Kleptography

Elsayed Mohamed and Hassan Elkamchouchi
Alexandria University, Alexandria, Egypt

Summary
This paper presents an approach to mount secretly embedded trapdoor with universal protection (SETUP) attacks on the elliptic curve discrete logarithm problem. The new approach allows the attacker to obtain the secret key of a cryptographic device covertly. The attack demonstrates the manufacturer's ability to embed a hidden trapdoor in cryptographic black-box devices used for key exchange. A contaminated device behaves exactly like an honest one while actually leaking the user's secret key only to the attacker. The attacker can then use that secret key to decrypt all the subsequent communications.

Key words:
Elliptic Curve Cryptography, Kleptography, Subliminal Channel, SETUP

1. Introduction

- **Elliptic Curves**

Elliptic curves are known for their security. The common fields used for encryption are prime fields and characteristic 2 fields. Elliptic curves over prime fields are on the form:
$E: y^2 = x^3 + ax + b \bmod p$
where $a, b \in F_p$ and $4a^3 + 27b^2 \neq 0 \bmod p$

The addition of two points $P(x_1, y_1)$ and $Q(x_2, y_2)$ is calculated by:
$R(x_3, y_3) = P + Q$ where:
$x_3 = \lambda^2 - x_1 - x_2$,
$y_3 = \lambda(x_1 - x_3) - y_1$,
$\lambda = (y_2 - y_1)/(x_2 - x_1)$ if $P \neq Q$
$\lambda = (3x_1^2 + a)/2y_1$ if $P = Q$

The multiplication of points by a scalar is a series of doublings and additions of points. The multiplication by -1 converts P to $-P$ by negating the y coordinate of P, i.e., the negative of $P = (x, y)$ gives $-P = (x, -y)$. Similar formulas exist for elliptic curves over characteristic 2 fields.

- **Elliptic Curve Discrete Logarithm Problem (ECDLP)**

Given a point P of order n in an elliptic curve E over a finite field F_p and a point Q in E , the ECDLP is to find an integer m , where $0 \leq m \leq n - 1$, and $Q = m \times P$ if such a number exists.

- **Elliptic Curve Diffie-Hellman Problem (ECDHP)**

Given a point P of order n in an elliptic curve E over a finite field F_p and two points kP and lP where $0 \leq k, l \leq n - 1$, the ECDHP is to find the point ($k \times l \times P$). This problem is used in the elliptic curve Diffie-Hellman key exchange algorithm.

- **Elliptic Curve Key Exchange**

Suppose that users A and B want to agree upon a key that they will use with a symmetric-key cryptosystem. They choose an elliptic curve E defined over a finite field F_p. Users A and B now construct their public keys from a randomly chosen and agreed upon point G lying on the elliptic curve E. E, F_p and G are made public. G need not be a generator of the group E but it is helpful if the subgroup of E generated by G is large and of the same order of size as E.
User A then randomly chooses an integer a and keeps it secret. User A then computes the point aG and sends it to user B. User B also randomly chooses an integer b and keeps it secret . User B then computes the point bG and sends it to user A. Both users A and B multiply their secret value by the received point. The computed shared key is abG. This shared key can now be used for subsequent encryption. The security of this system lies in the fact that a third party C that knows only aG and bG cannot efficiently calculate the shared key abG as per the ECDHP.

- **Subliminal Channels**

Subliminal channels can be used to convey information in the output of a cryptosystem in a way different from the intended output. This notion was put forth by Simmons [1]. He demonstrated how a prisoner could leak secret messages to an outside partner without the warden knowing what is going on. The warden has the ability to read every message but still cannot read the secret message embedded within the cover message. Simmons further developed the concept to other applications including DSA [2].

- **Kleptography**

Kleptography is defined as the study of stealing information securely and subliminally within the context of cryptographic systems [3]. A kleptographic attack on the discrete logarithm problem has been introduced by Young and Yung in [3]. They defined a Secretly Embedded Trapdoor with Universal Protection (SETUP) as an algorithm that can be embedded within a cryptosystem to leak encrypted secret key information to the attacker in the output of that cryptosystem [4]. The encrypted secret key information is noticeable only to the attacker. The types of SETUP [3] and their definitions are listed below.

Definition 1. Assume that C is a black-box cryptosystem with a publicly known specification. A (regular) SETUP mechanism is an algorithmic modification made to C to get C' such that:

1. The input of $C`$ agrees with the public specifications of the input of C.
2. $C`$ computes efficiently using the attacker's public encryption function $E$ (and possibly other functions as well), contained within $C`$.
3. The attacker's private decryption function D is not contained within $C`$ and is known only by the attacker.
4. The output of $C`$ agrees with the public specifications of the output of C. At the same time, it contains published bits (of the user's secret key) which are easily derivable by the attacker (the output can be generated during key-generation or during system operation like message sending).
5. Furthermore, the output of C and C' are polynomially indistinguishable (as in [5]) to everyone except the attacker.
6. After the discovery of the specifics of the SETUP algorithm and after discovering its presence in the implementation (e.g. reverse-engineering of hardware tamper-proof device), users (except the attacker) cannot determine past (or future) keys.

Definition 2. A weak SETUP is a regular SETUP except that the output of C and C ' are polynomially indistinguishable to everyone except the attacker and the owner/user of the device who is in control (knowledge) of his or her own private key (i.e., requirement 5 above is changed).

Definition 3. A strong SETUP is a regular SETUP, but in addition we assume that the users are able to hold and fully reverse-engineer the device after its past usage and before its future usage. They are able to analyze the actual implementation of C' and deploy the device. However, the users still cannot steal previously generated/future generated keys, and if the SETUP is not always applied to future keys, then SETUP-free keys and SETUP keys remain polynomially indistinguishable.

Definition 4. A kleptogram is an encryption of a value (hidden value) that is displayed within the bits of an encryption/signature of a plaintext value (outer value). Note that we say that a kleptogram is an encryption of a value, not a plaintext message. It is often the case in kleptography that the device is not free to choose this value. The device may calculate this hidden value, and then use it (for the 'randomness') in a subsequent computation, thus compromising that computation.

Definition 5. A SETUP that has (m, n)-leakage bandwidth leaks m secret messages over the course of n messages that are output by the cryptographic device (or n of its executions).

2. Proposed ECDLP SETUP Attack

The SETUP attack on ECDLP assumes that the only value the device outputs is $M = cG$, where c is the generated secret and G is a base point of order n. The private key of the attacker is v and the public key is $V = vG$. H is a cryptographically secure hash function that generates values less than n. Hashing an elliptic curve point can be defined as hashing its x coordinate. The algorithm in the device works as follows:
If this is the first time, run Algorithm 1, else run Algorithm 2:

```
Algorithm 1:
    1.1. Choose $c_1$ randomly where $2 \leq c_1 \leq n-1$
    1.2. Store $c_1$ in non-volatile memory of the device
    1.3. Output $M_1 = c_1G$
```

```
Algorithm 2:
    2.1 $Z = a.c_1G + b.c_1V + h.jG + e.uV$, where:
        $a, b, h, e$ are fixed integers $< n$
        $j, u \in_R \{0, 1\}$ are uniformly and independently chosen at random
    2.2 $c_2 = H(Z)$
    2.3 Store $c_2$ in non-volatile memory of the device
    2.4 Output $M_2 = c_2G$
```

The attacker needs to monitor the communication channel and obtain M_1 and M_2. The attacker can then calculate the user's secret c_2 using Algorithm 3 as follows:

```
Algorithm 3:
    3.1 $Z_1 = a.M_1 + b.vM_1 = a.M_1 + b.v.c_1G = a.c_1G + b.c_1V$
    3.2 For each possible value of $j, u$:
    {
        Calculate $Z_2 = Z_1 + h.jG + e.uV$
        $c_2 = H(Z_2)$
        If $c_2G = M_2$ then output $c_2$ and exit
    }
```

3. Discussion and Analysis

A. Security

Since c_1 is random it follows that Z is uniformly distributed within the group generated by G. This SETUP attack is secure in the sense that a user not knowing the random choice c_1 cannot calculate the second private key c_2 as long as ECDHP is hard. This can be proven by supposing that an oracle A can solve the ECDH problem so that $A(aG, bG) = abG$. If A is applied on M_1 and V then: $A(M_1, bV) = b.v.c_1G = b.c_1V$, which can be used to calculate Z. Also an adversary that does not know the attacker's private key v cannot calculate Z and therefore cannot calculate c_2. This makes the universal protection property of the SETUP attack. Assuming that H is a pseudorandom function and that the device can be reverse-engineered, the outputs of C and C` are polynomially indistinguishable. This results from Z being uniformly distributed and H being a pseudorandom function. This proves that the ECDLP SETUP attack is a strong SETUP as long as ECDHP is hard and the secret value generated by the device is inaccessible to the owner. The random values j and u are used to add randomization to further insure undetectability of SETUP in a black-box implementation. Adding them serves as a precaution so that if the secret values c_i are available to the user and the hash function H is invertible, the user still cannot detect the presence of a SETUP in the device by running the device many times and guessing several different values of V. It also helps to curb trying to notice any possible probabilistic relations between some properties in V and some corresponding properties in Z. This kind of probabilistic detection by the user is very difficult in elliptic curve cryptosystems compared to discrete log systems where quadratic residuosity can be used to test a possible relation between the attacker's public key and Z [3]. This makes elliptic curve devices a better candidate for kleptographic attacks in addition to the improved security and key length advantages of elliptic curve systems.

B. Strong SETUP in ECDHP

A strong SETUP attack on ECDHP can be implemented using the ECDLP SETUP attack as long as the contaminated device does not output the secret value it chooses to the user. To implement the attack, the attacker includes his public key V within the user's device and uses it to compute the secret value starting from the second key exchange as described above. The attacker can then compute the shared key by multiplying the known private key of one partner by the public key of the other and thus be able to decrypt all encrypted communications that follow.

C. Bandwidth

We can increase the leakage bandwidth of the attack by chaining the leaked secret values such that Algorithm 2 is used to calculate c_3 from c_2, c_4 from c_3 and so on. Running the device for m + 1 times leaks m secret values. The leakage bandwidth is (m, m + 1).

4. Conclusion

We have shown that a strong SETUP attack can be mounted on ECDLP and ECDHP key exchange. This enables a malicious manufacturer of black-box cryptosystems like smart card devices to implement such attacks to get exclusive access to the user's private key. The output of a dishonest device is indistinguishable from the output of an honest one.

References

[1] G. J. Simmons, "The Prisoner's Problem and the Subliminal Channel", *Crypto '83,* pp. 51-67, 1983.

[2] G. J. Simmons, "Subliminal Communication is Easy Using the DSA", *Eurocrypt '93*, pp. 218-232, 1993.

[3] A. Young, M. Yung, "Kleptography: Using Cryptography Against Cryptography", *Eurocrypt '97*, pp. 62-74.

[4] A. Young, M. Yung, "The Dark Side of Black-Box Cryptography or Should We Trust Capstone?", *Crypto '96*, pp. 89-103.

[5] S. Goldwasser, S. Micali, "Probabilistic Encryption", *J. Comp. Sys. Sci.* 28, pp. 270-299, 1984.