Practical Invalid Curve Attacks on TLS-ECDH

MAR 7TH, 2016

论文下载

Abstract

椭圆曲线密码学(ECC)是基于群的运算，在做密码学运算之前都应该检查群元素的合法性。但是在实际使用中，有些密码学库忽略了“点是否位于曲线上”这一检查，从而导致使用TLS-ECDH的服务器可能被攻击者获得私钥。

Introduction

ECC $y^2 = x^3 + a*x + b$ 生成元，曲线的阶数，点的阶数， C = (F,a,b)

TLS-ECDH

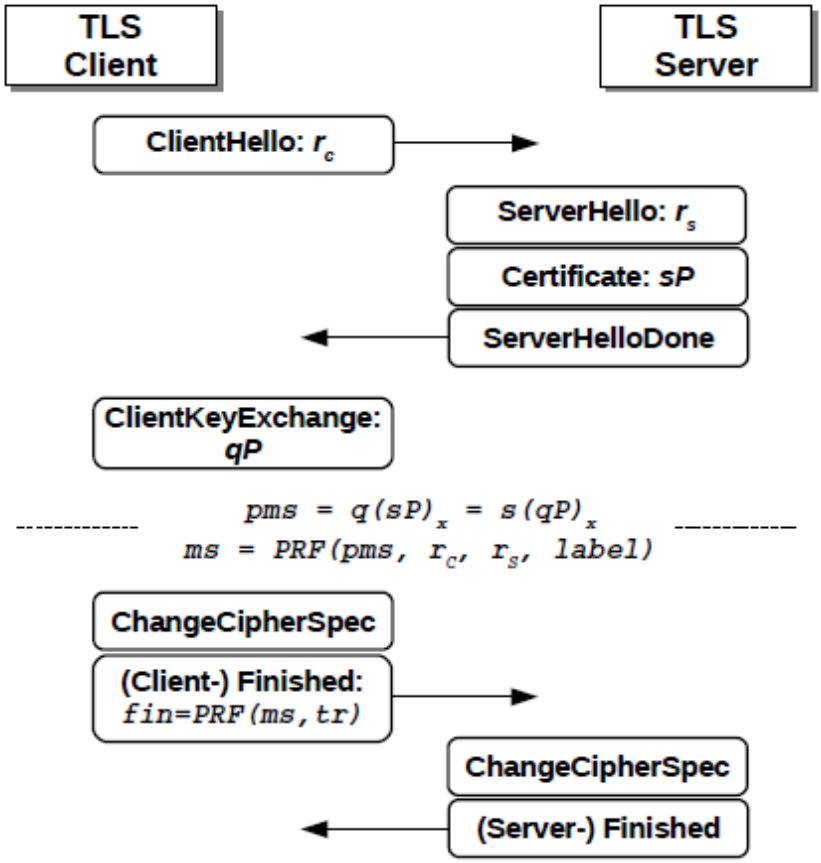

Fig. 1. Structure of the SSL/TLS Handshake protocol for TLS_ECDH cipher suites.

Invalid Curve Attacks on ECC

Crypto 2000, Biehl et al.: Differential fault attacks on elliptic curve cryptosystems

ADD Double

ADD(P,Q) :	DBL(P) :
$(x_P, y_P) := P; (x_Q, y_Q) := Q$	$(x_P, y_P) := P$
If $P = O_\infty$ **then Return** Q	**If** $P = O_\infty$ **then Return** P
If $Q = O_\infty$ **then Return** P	$\lambda := (3x_p^2 - a)/(2y_P)$
$\lambda := (y_P - y_Q)/(x_P - x_Q)$	$x_R := \lambda^2 - 2x_P$
$x_R := \lambda^2 - x_P - x_Q$	$y_R := y_P + \lambda(x_R - x_P)$
$y_R := y_P + \lambda(x_R - x_P)$	**Return** (x_R, y_R)
Return (x_R, y_R)	

Fig. 2. Algorithms DBL and ADD for point doubling and addition. Note that both algorithms are independent of the curve parameter b.

```
def mul(n,P):
    R = INF
```

```
    while n!=0:
        if n&1 ==1:
            R += P
        P += P
        n >>=1
    return R
```

注意到加法和乘法运算都没有使用b!

如果不检查点的合法性，运算就可能在另外一条曲线上（a相同，b不同)。

Oracle:“在曲线（F,a,b)上运算，随机私钥s，对于输入的点G，使用double-and-add 算法计算s*G*，并返回*s*G的x坐标。

oracle并不检查输入的点是否在曲线上。返回x坐标的原因是在ECC中key通常来自点的x坐标。

攻击的整体思路： 寻找不同的b',使得新曲线(F,a,b')的阶可被小素数p_i整除(2,3,5,7...),这样新曲线上就存在阶数为p_i的点G',假设存在oracle可以将 s*G'* 作为结果返回， 有$t < p_i$, $sG' = t*G'$.就得到s mod p_i的值。使用足够多的小素数p_i,利用中国剩余定理，就可计算出s的值。

已知曲线(F,a,b) 和曲线的阶q

Offline Precomputations

第一步： $p_1,p_2,...,p_n$ 为前n个素数，其积大于q^2.随机选取b_i,使得曲线(F,a,b_i)的阶能被某个p_i整除。

n是很小的，O(log q*log log q).

- NIST P-192,$q < 2^{192}$, n = 60, p_n = 283.
- NSIT P-256,$q < 2^{256}$, n = 76, p_n = 383.

第二步：曲线(F,a,bi)就存在某个点G_i,其阶为p_i。（small subgroup attack。 how？）

2.3 GHz CPU, 4GB RAM, 90 minutes for NIST P-192, 5 hours for NIST P-256

Online Attack

计算出(b_i,G_i,p_i)后
第一步：攻击者将G_i输入给oracle，oracle返回$s*G_i$的横坐标。oracle认为自己在曲线(F,a,b)上运算，但由于double-and-add算法与b无关，因此实际上oracle是在曲线(F,a,b_i)上运算。
第二步: 攻击者计算t，$t < (p_i+1)/2$,使得 $[sG_i]x = [tG_i]x$. $s = t \bmod p_i$, 或者 $-s = t \bmod p_i$. $s_2 = t_2 \bmod p_{ii}$ 得到$s_2 \bmod p_i$的值。
第三步：对所有的i，i=1,...,n, 获得$s_2 = t_{i2} \bmod p_i$,而且$s_2 < q_2$. 利用中国剩余定理，计算出s_2，从而得到s。

Invalid Curve Attacks on TLS-ECDH

实际的TLS server有一点不符合之前oracle，TLS server并不会将s*G的值返回给客户端。
oracle非常聪明的在ClientKeyExchange的时候将G_i发送给TLS server，猜测t，使得$[sG]x = [tG]x$,并计算出master secret，如果猜测正确，ClientFinished将被server接收并返回ServerFinihed，否则server会alert并结束连接。

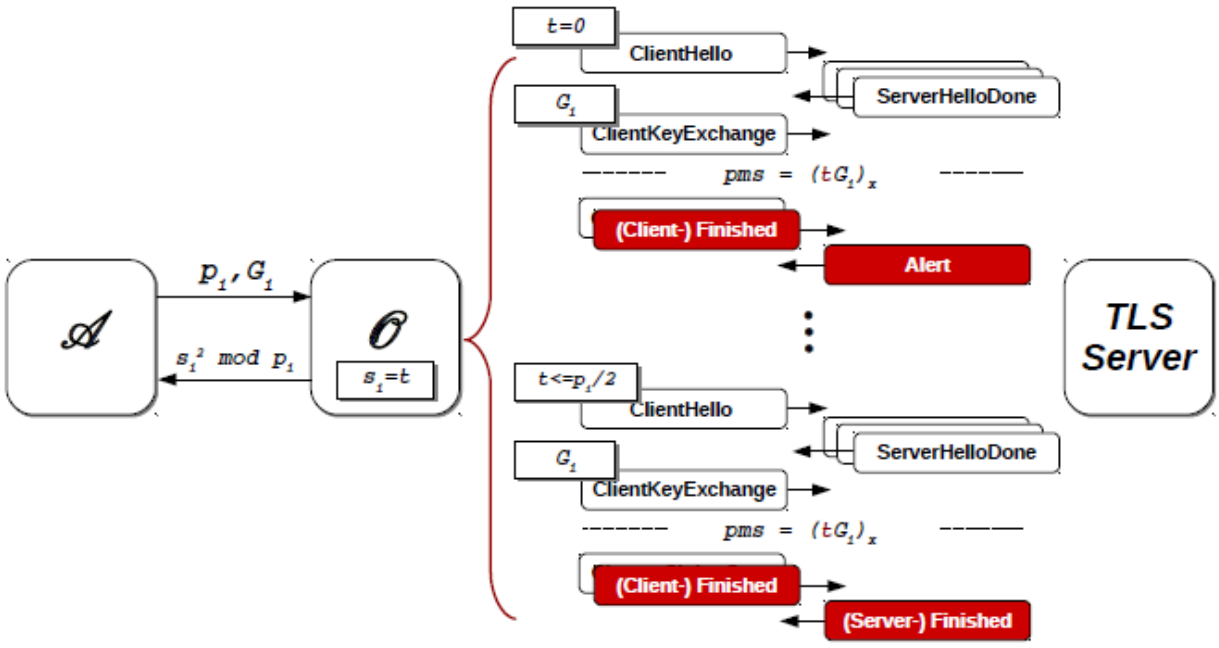

Fig. 3. Constructing an oracle $\mathcal{O}$ from a vulnerable TLS server supporting TLS-ECDH cipher suites.

Practical Evaluation & Analysis

不检查点合法性的密码库： Bouncy Castle Java 1.50，SunEC Security Provider 1.8，WolfSSL 3.4.6(embeded)

JSSE Elliptic Curve # of oracle queries # of server queries Duration [sec] secp256r1 74 3300 155

CVE-2015-6924 Hardware Security Modules：crypto storage device

ECDH is less used than ECDHE, ephemeral

Check the point! Old attacks still applicable, we can learn a lot from them

https://github.com/mimoo/socat_backdoor