

伪造中本聪签名

2022 年 7 月 31 日

学	校:	山东大学
学	院:	网络空间安全学院 (研究院)
姓	名:	张起萌 202000460118

目录

1 实验目的	3
2 实验过程	3
3 实验结果	3
4 参考文献	4

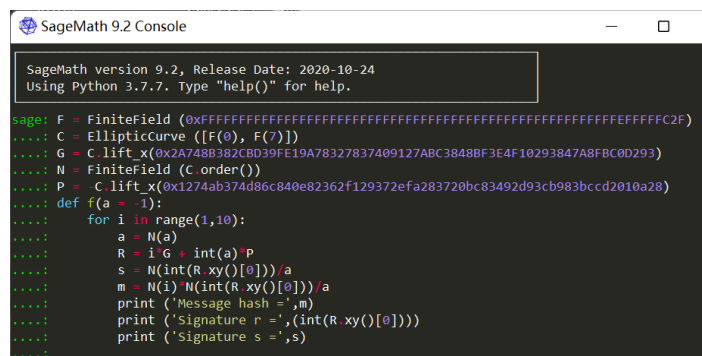
1 实验目的

利用 ECDSA 算法伪造中本聪签名。

2 实验过程

由于 ECDSA 算法中涉及到大量椭圆曲线问题，而这些问题往往在 c/c++/python 中很难解决，通过查询资料 (表明为参考文献 1) 发现，sagemath 中有专门的椭圆曲线软件包 EllipticCurve，所以我决定使用 sagemath 来实现。

- 利用 sagemath 中自带的函数 FiniteField 构建一个有限域 F。随后选择该有限域中两点作为椭圆曲线的两个系数，利用函数 EllipticCurve 构造椭圆曲线 C。
- 利用 EllipticCurve.lift_x 计算基点 G，同时在椭圆曲线 C 的基础上重新建立一个有限域 N，方便产生多组密钥。
- 选择私有密钥 $a=-1$ ，遍历 i 从 1 到 10(得到 10 组结果，方便检验正确性)，每次都重新产生一个密钥 a，该密钥为前一个密钥在有限域 N 上的对应点。
- 计算点 $R = iG$ 。
- 计算 $s \equiv r - Hash * a(modn)$ 。
- 将 i 和点 R 的坐标值 x,y 作为参数输入，计算 hash 值 m。
- r 和 s 做为签名值，消息 m 作为 hash 值输出。



```
SageMath 9.2 Console
SageMath version 9.2, Release Date: 2020-10-24
Using Python 3.7.7. Type "help()" for help.

sage: F = FiniteField(0xFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF2F)
....: C = EllipticCurve([F(0), F(7)])
....: G = C.lift_x(0x2A748B382C8D39FE19A78327837409127ABC3848BF3E4F10293847A8FBC0D293)
....: N = FiniteField(C.order())
....: P = C.lift_x(0x1274ab374d86c840e82362f129372efa283720bc83492d93cb983bccd2010a28)
....: def f(a = -1):
....:     for i in range(1,10):
....:         a = N(a)
....:         R = i*G + int(a)*P
....:         s = N(int(R.xy()[0]))/a
....:         m = N(i)*N(int(R.xy()[0]))/a
....:         print('Message hash =',m)
....:         print('Signature r =',(int(R.xy()[0])))
....:         print('Signature s =',s)
....: 
```

图 1: CodeBlocks 上 SM4 基础实现

3 实验结果

实验结果中一共输出了 10 组消息 hash(m) 和签名值 (r,s)，可以验证每一组值都满足以下条件：

- 计算： $sG + H(m)P = (x1,y1)$, $r1 \equiv x1 \bmod p$ 。
- $r1 \equiv r \bmod p$ 等式成立

伪造中本聪签名成功。

```
SageMath 9.2 Console
sage: f()
Message hash = 77632427137175313995946521445958794908687355084765773004810358360058052007316
Signature r = 38159662100140881427624463562729112944150209194309131377794804781460109487021
Signature s = 77632427137175313995946521445958794908687355084765773004810358360058052007316
Message hash = 107252840860941142282782915752566597198424690417122985475966526208305419099991
Signature r = 4269624188187526570394034628060655327206436930975959453319318466606371197173
Signature s = 111522465049128668853176950380627252525631127348098944929285844674911790297164
Message hash = 10038697813868290749066530968250559524554517758443266026431200564154593744837
Signature r = 43732400111983161118825553444956740153376650324572382833632773547163443081279
Signature s = 72059689125333034304745431563731167699460913954502521548972389594354718413058
Message hash = 104577351945298480434336785702959970686261156241992353264520869007744600148084
Signature r = 89647751250991575314986788582947915181272275218576816066474945889582011457316
Signature s = 26144337986324620108584196425739992671565289060498088316130217251936150037021
Message hash = 18300898323397247370490718684744881083958926924504804566795540294746790778209
Signature r = 42656656030247028695330250266526186924343240326729000839682957197657906442093
Signature s = 73135433207069166728240734742161720928494323952345903542922205943860255052244
Message hash = 105806529115125033989705821246767580486916905940737494700768593636451354230347
Signature r = 78858986178575990521358183966111993129545152575772837868709537011856575540223
Signature s = 36933103058740204902212801042575914723292411703302066513895626129661585954114
Message hash = 38964392298642268331422131902195442832027131819692010053512349970306094756090
Signature r = 93684020446464986315714825449990286326428321979251059463159804125543267744276
Signature s = 22108068790851209107856159558697621526409242299823844919445359015974893750061
Message hash = 27543965674625658145673090420965730258190512155135989158275147454256600845732
Signature r = 54453048909329890443576356201723237644144968120145453546518188138977005641452
Signature s = 61339040327986304979994628806964670208692596158929450836086975002541155852885
Message hash = 3757925846565921744429763665727700366843063250403480012481073693580337865897
Signature r = 7677717884192569453299057182071108297224280260338438475905545017280959011125
Signature s = 39014910395390500890571413187976824880595084009736465906699618124237202483212
```

图 2: 实验结果

4 参考文献

ECDSA 数字签名算法

如何看待 Craig Wright 宣称自己是比特币创始人中本聪?

Greg Maxwell: CSW 再次伪造中本聪签名