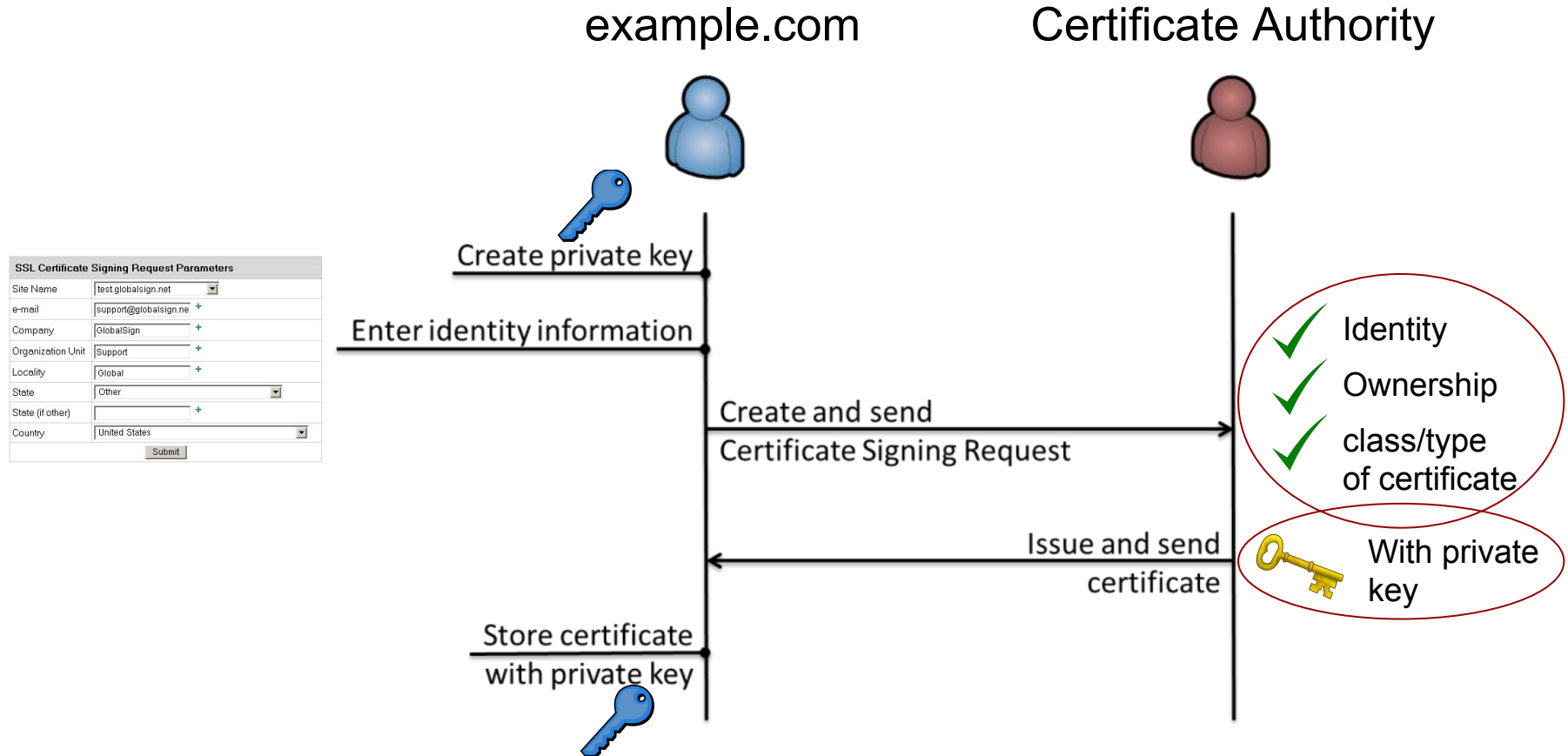


Decentralized Certificate Authority

Bargav Jayaraman
Hannah Li

Review - Creating a Certificate



Motivation

Google Reducing Trust in Symantec Certificates Following Numerous Slip-Ups

By [Catalin Cimpanu](#)

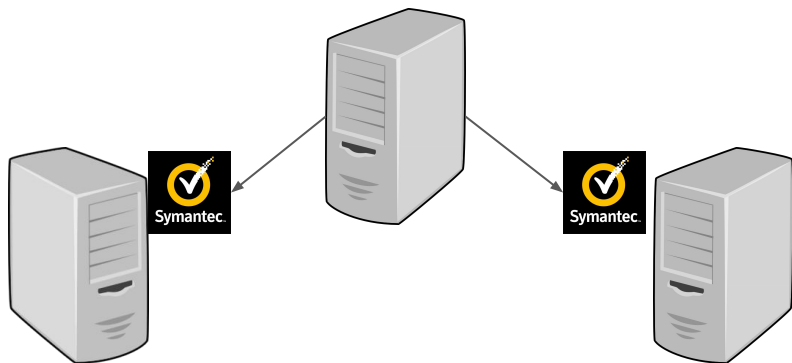
March 23, 2017 04:58 PM

Hacked Certificate Authorities - Nothing Left to Trust

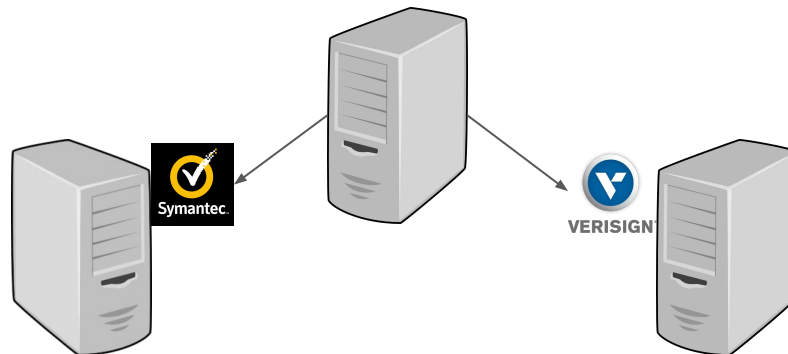
Monday, September 12, 2011

2011 Comodo and DigiNotar

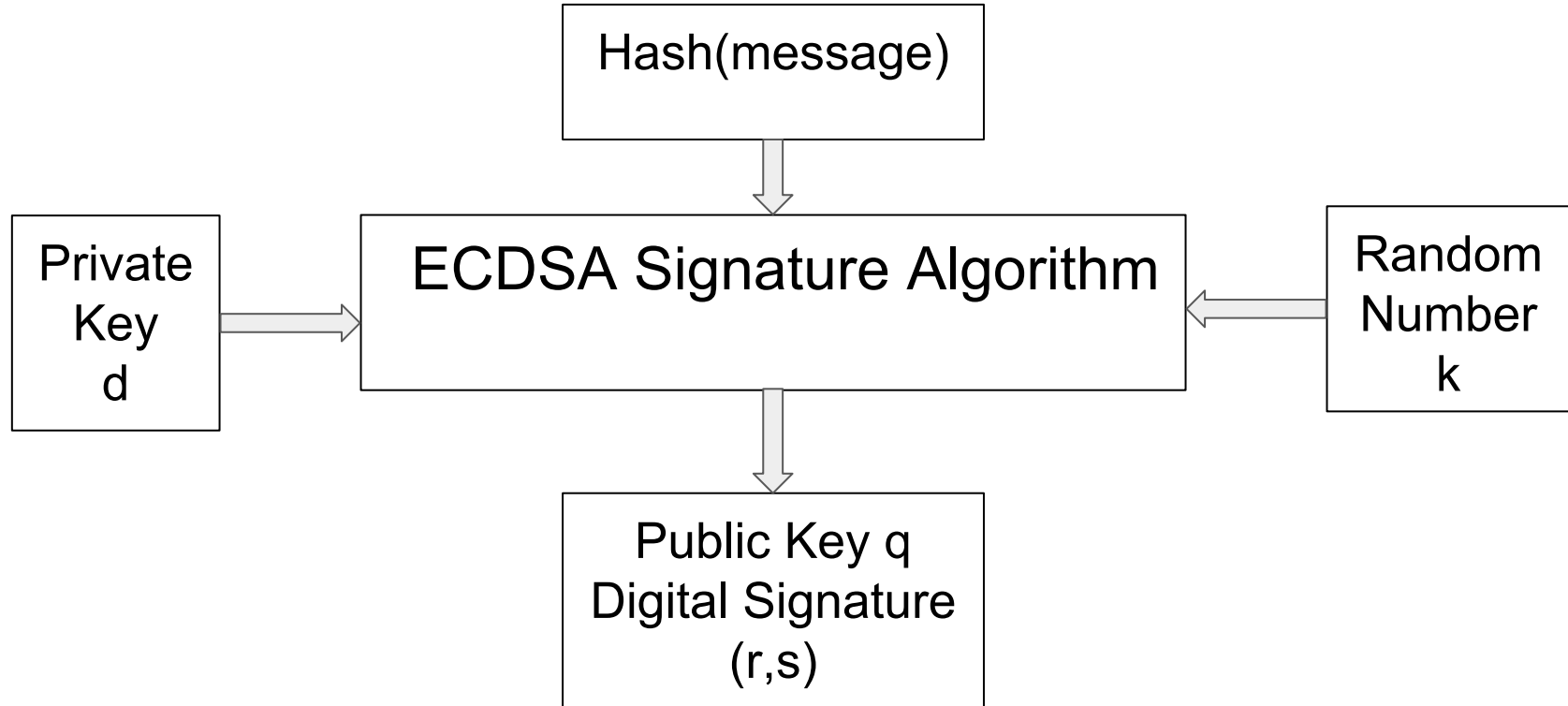
The certificates are
only as trustworthy as
the weakest CA...



OR

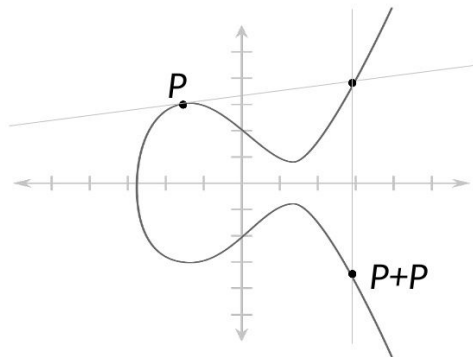


Background - Digital Signature Algorithm (DSA)



Background - Elliptic Curve DSA

Sample Curve



ECDSA Signature Algorithm

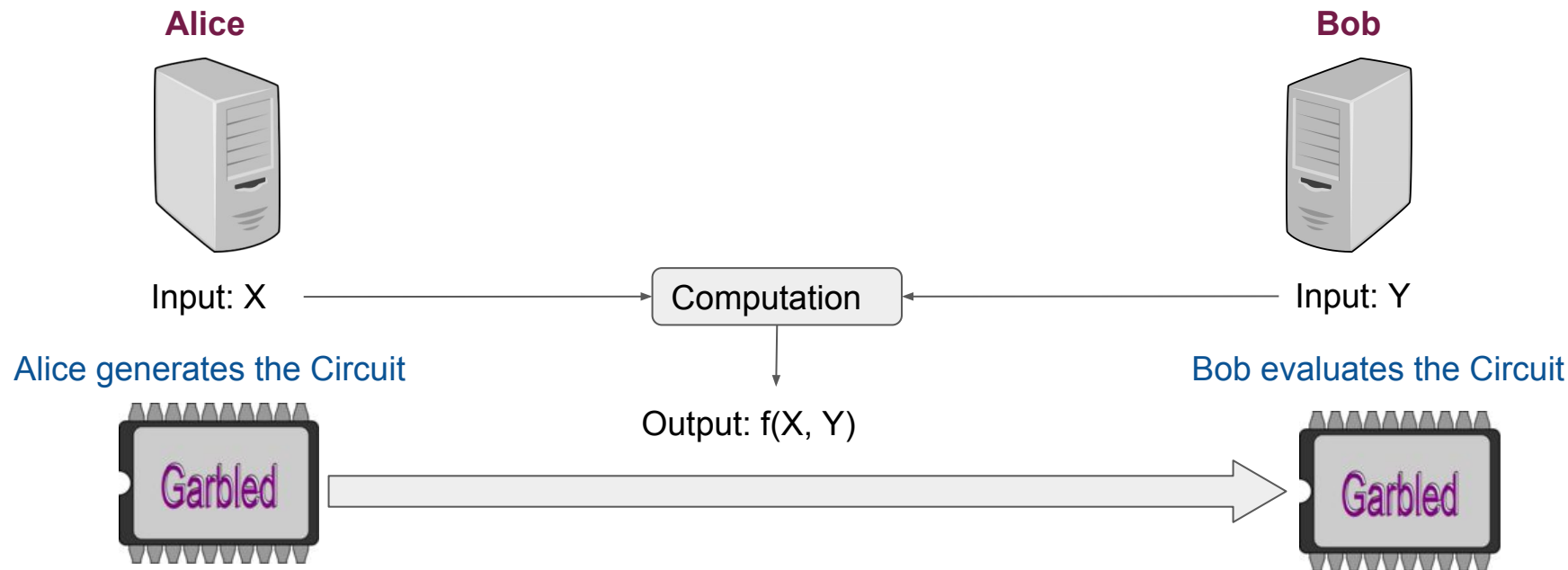
Parameters = (p, b, G, n, h)

Public Key:
 $Q = d * G$

Signature:
 $(x, y) = k * G$
 $r = x \bmod n$
 $s = k^{-1}(\text{Hash}(m) + rd)$

d = CA private key
 k = Random number
 m = info to be signed

Background - Multiparty Secure Computation



Framework ==> OblivC + Absentminded Crypto

Generic Computation

Obliv-C



```
obliv int x, y, z;
```

```
feedOblivInt(&x, io->x, 1);
```

```
feedOblivInt(&y, io->y, 2);
```

```
z = x + y;
```

```
revealOblivInt(&io->z, z, 0);
```

Absentminded Crypto provides 'big number' arithmetic over OblivC!

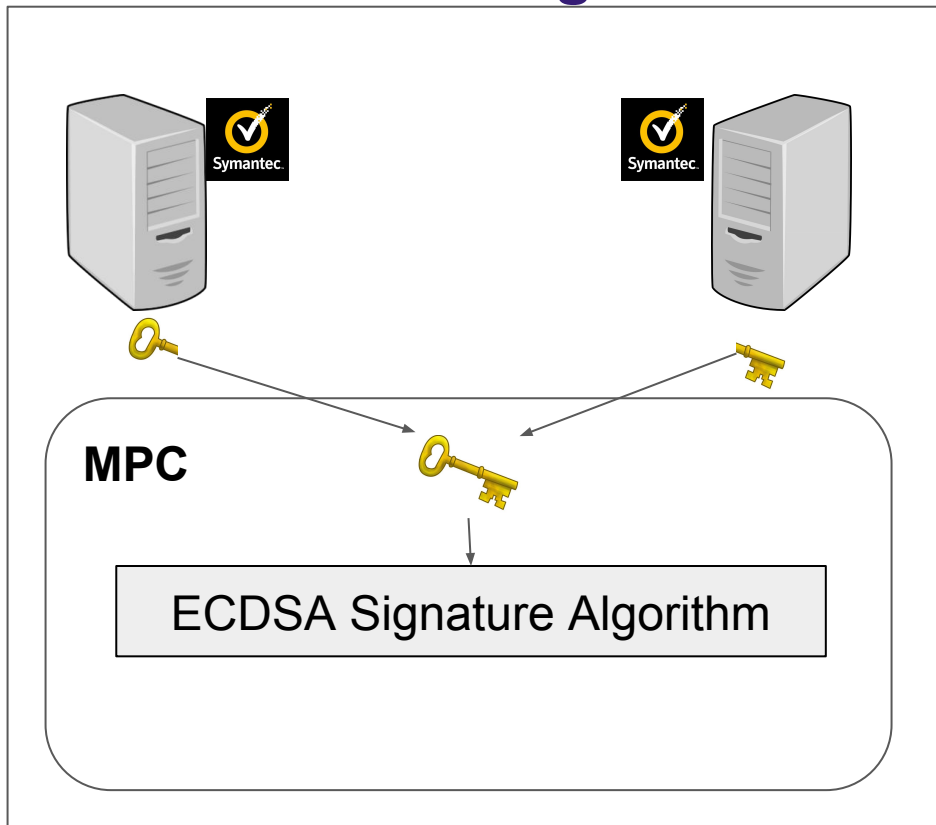
Use **obig** keyword for big numbers..

Decentralized CA using MPC

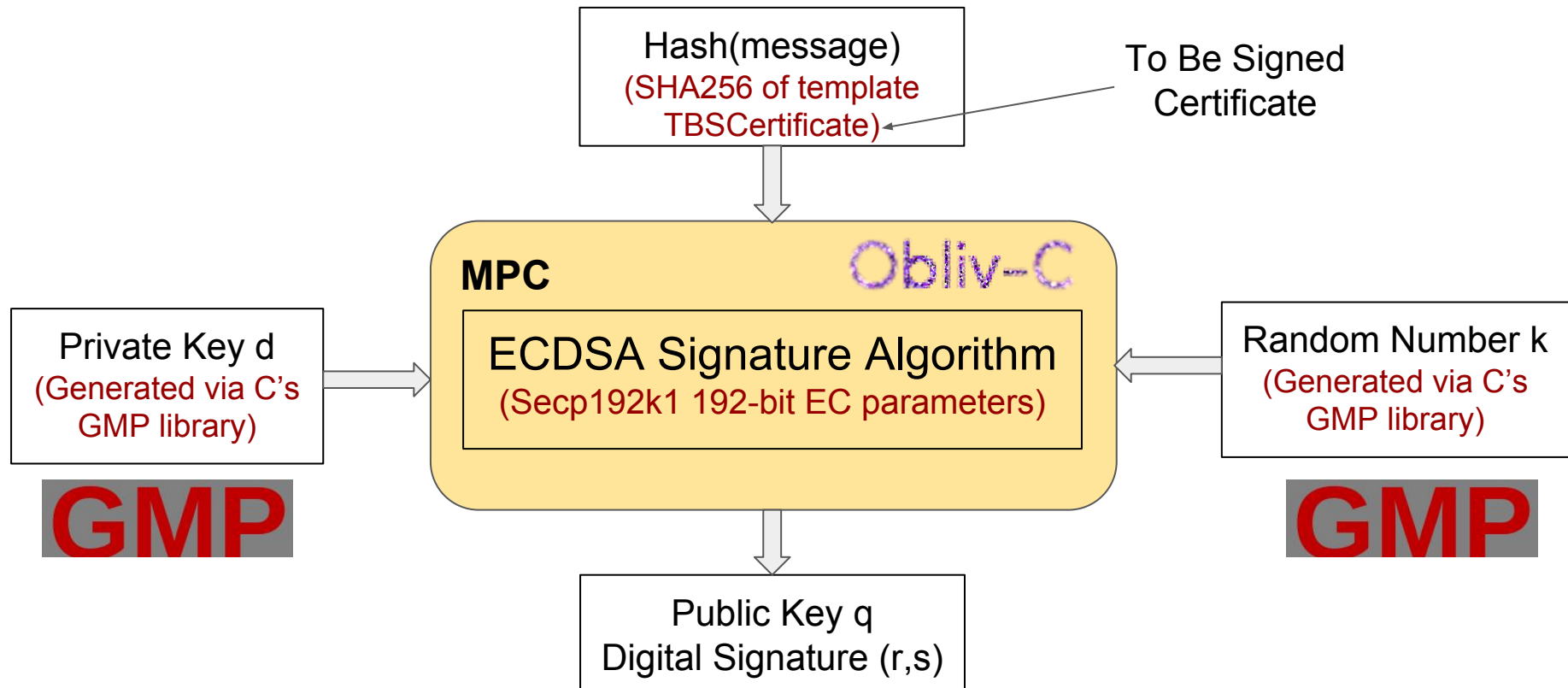
CA Organization



example.com



Implementation in C



Experiments

3-4 hours to perform certificate signing using 'secp192k1' elliptic curve.

Computation Bottleneck: Curve Point-multiplication

```
pointMultiplication(G, d):
```

```
  N ← G
```

```
  Q ← 0
```

```
  for i from 0 to m do
```

```
    if  $d_i = 1$  then
```

```
      Q ← pointAdd(Q, N)
```

```
      N ← pointDouble(N)
```

```
  return Q
```

```
pointAdd(P, Q):
```

$$\lambda = \frac{Q_y - P_y}{Q_x - P_x}$$

$$R_x = \lambda^2 - P_x - Q_x$$

$$R_y = \lambda(P_x - R_x) - P_y$$

Signature:

$$(x, y) = k * G$$

$$r = x \bmod n$$

$$s = k^{-1}(\text{Hash}(m) + rd)$$

```
pointDouble(P):
```

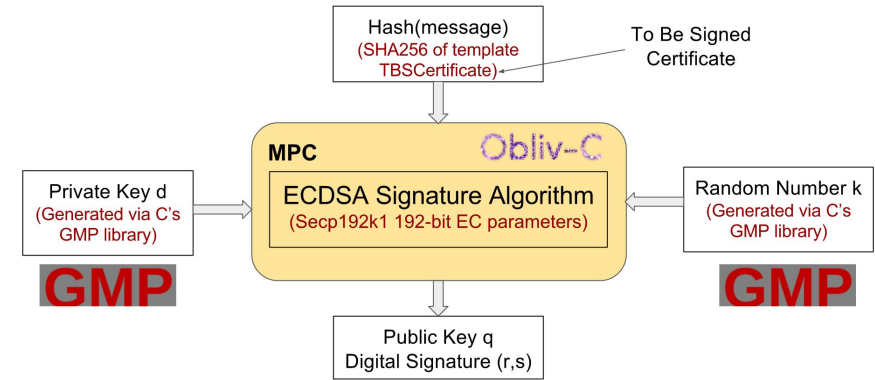
$$\lambda = \frac{3P_x^2 + a}{2P_y}$$

$$R_x = \lambda^2 - 2P_x$$

$$R_y = \lambda(P_x - R_x) - P_y$$

Conclusion

Decentralized CA is feasible in practice



Deployment in open source tool OpenSSL



Certificate Signing takes around 3-4 hours, which can be optimized

