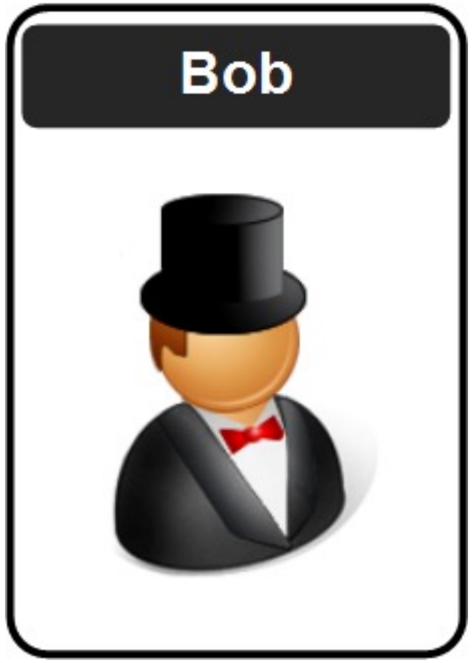
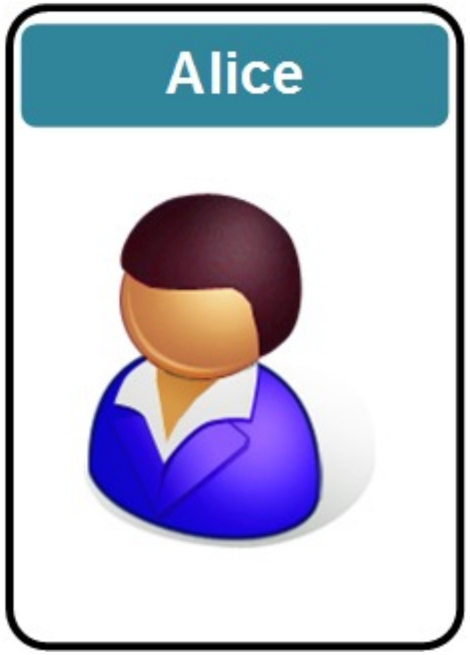


Shamir Secret Shares and Elliptic Curves (and Golang)



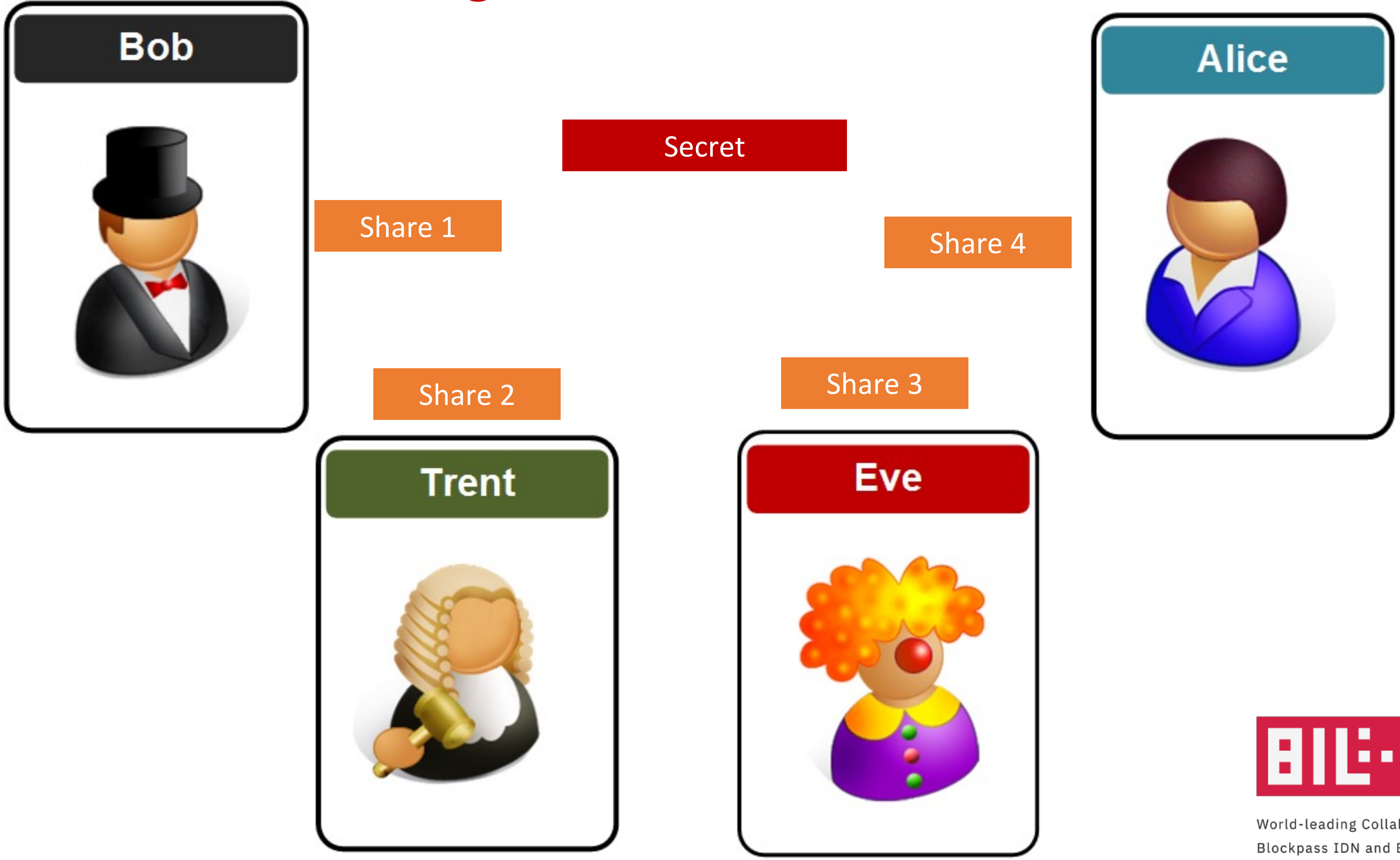
Prof Bill Buchanan OBE, Blockpass ID Lab

http://asecuritysite.com

BLOCKPASS IDENTITY LAB

World-leading Collaboration between Blockpass IDN and Edinburgh Napier University

Perfect Secret Sharing



Bob

Secret

Share 1

Share 4

Alice

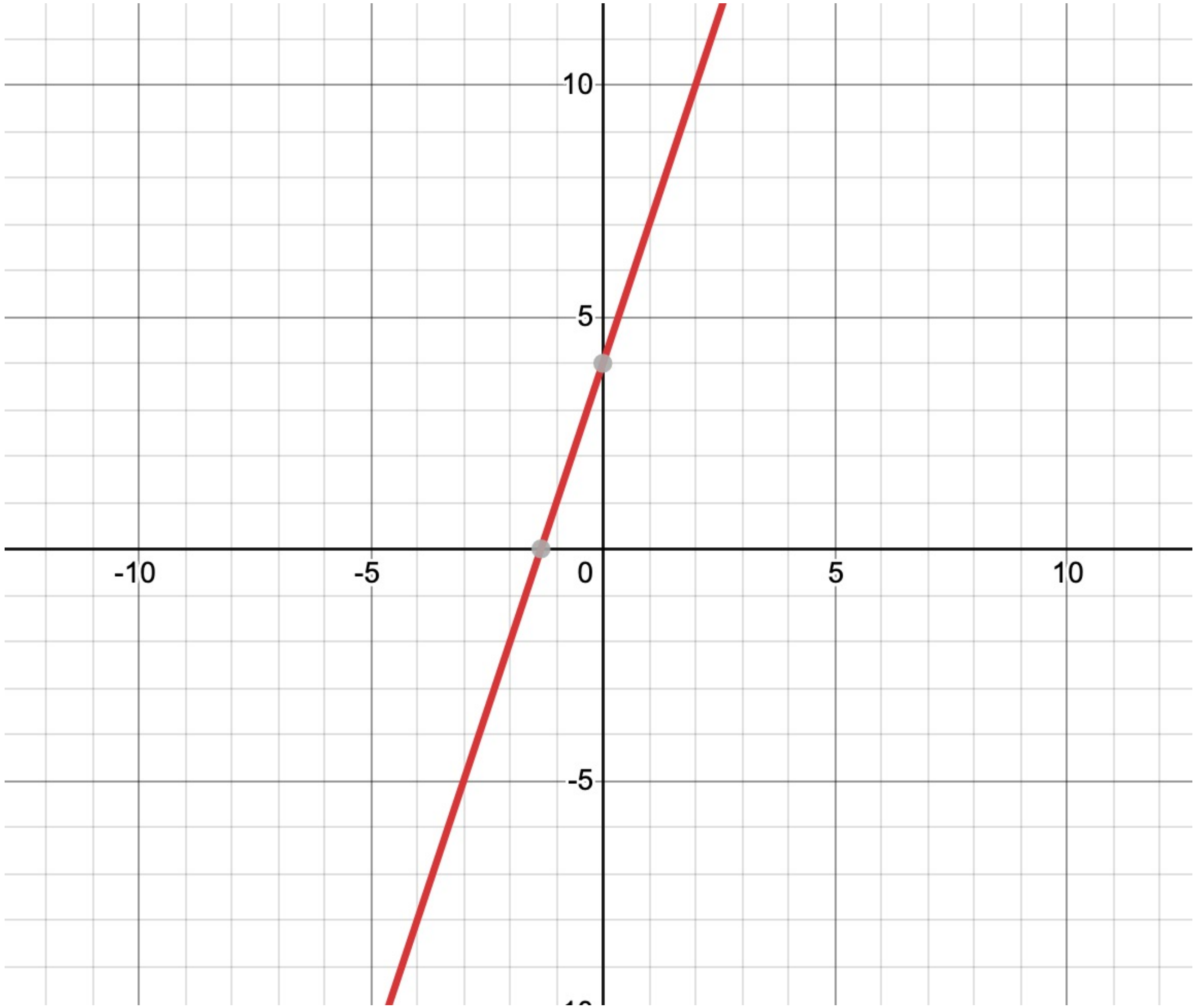
Share 2

Share 3

Trent

Eve

Shamir Secret Sharing (SSS)



All or nothing:
f(x)=3x + 4

Bob (1,7)
Alice (2,10)

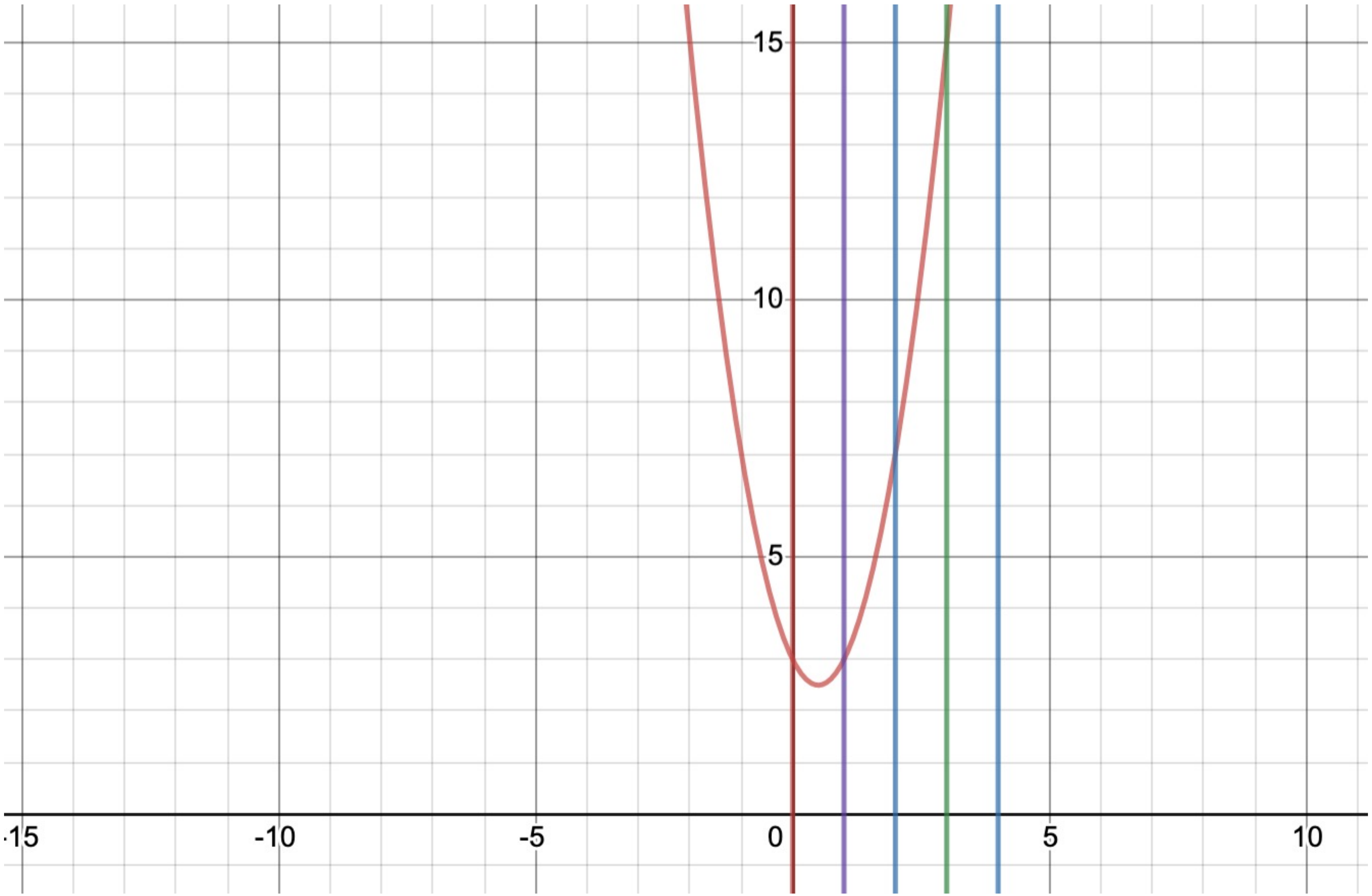
Any 2 from 3:
f(x)=3x + 4

Bob (1,7)
Alice (2,10)
Carol (3,13)

BLOCKPASS IDENTITY LAB

World-leading Collaboration between
Blockpass IDN and Edinburgh Napier University

Shamir Secret Sharing (SSS)

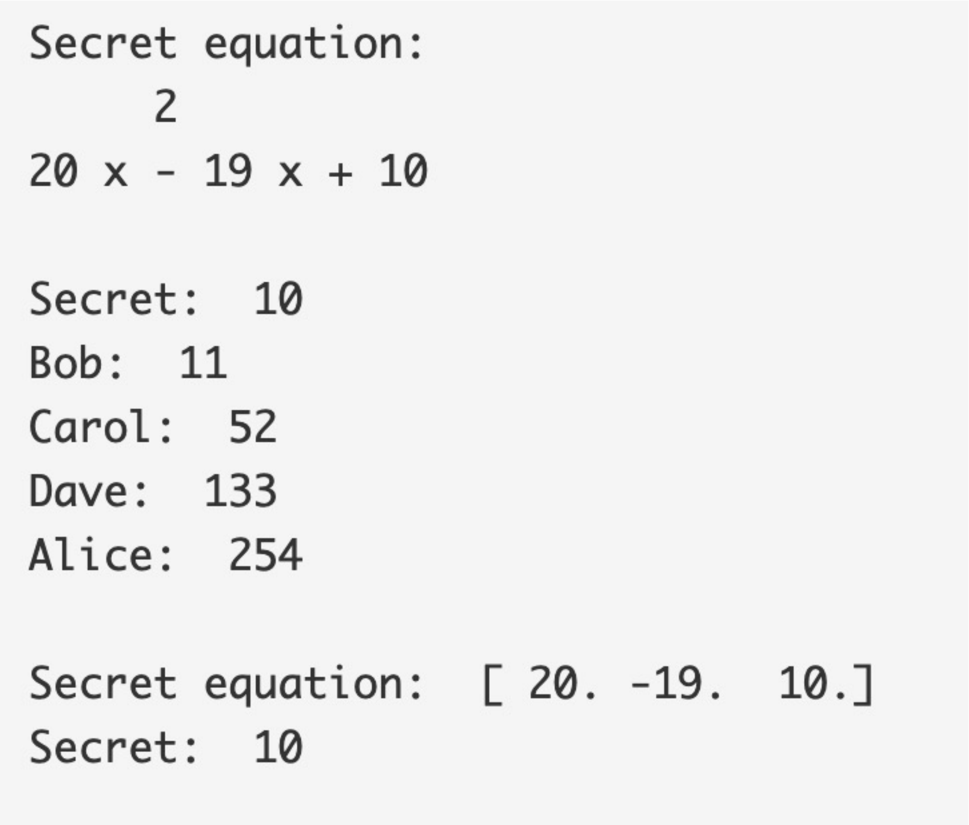


Any three from four requires a **quadratic equation**:
$f(x)=2x^2 -2x + 3$

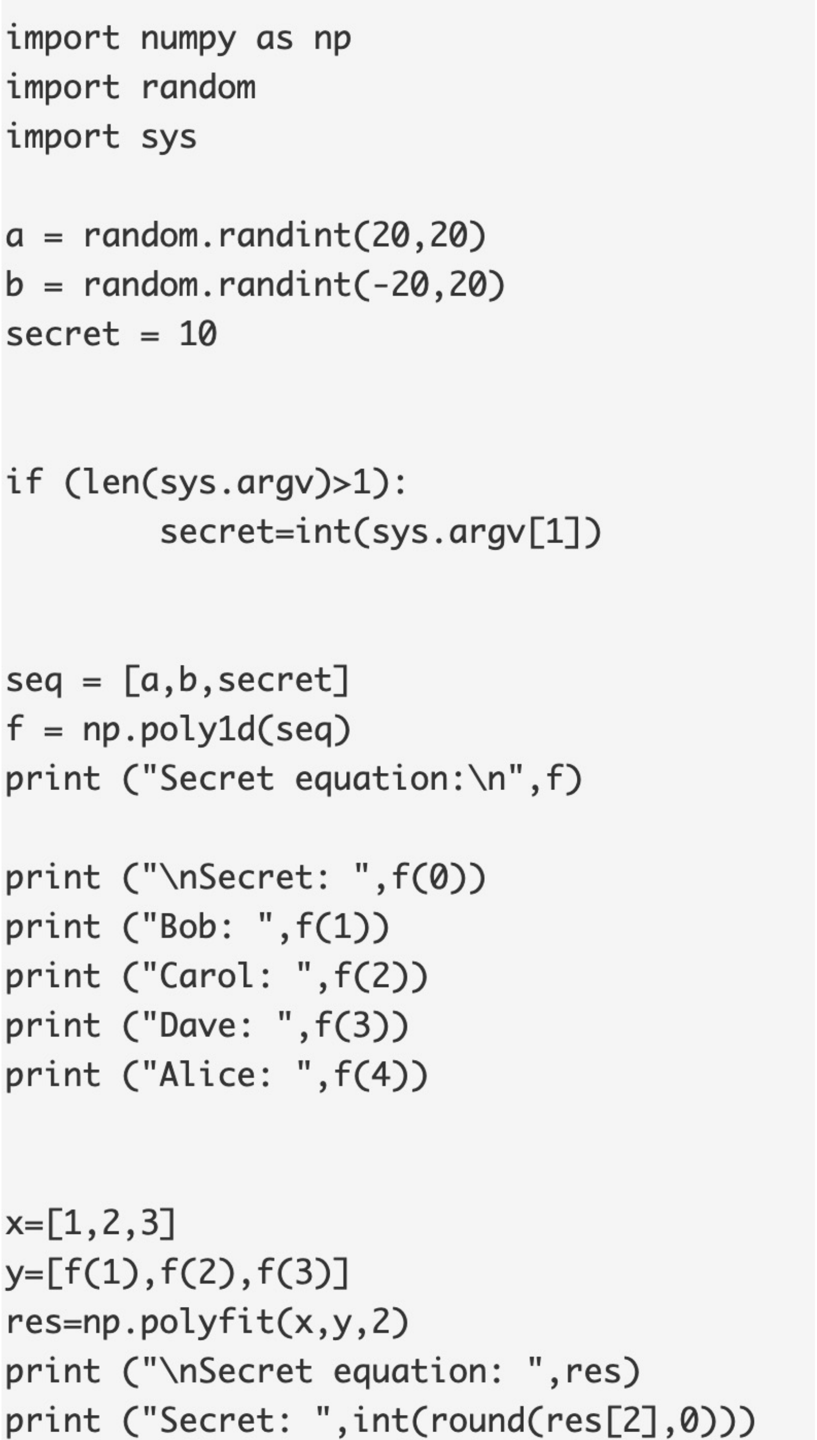
Bob (1,3)
Carol (2,7)
Dave (3,14)
Alice (4,37)

Shamir Secret Sharing (SSS)

$20x^2 - 19x + 10$

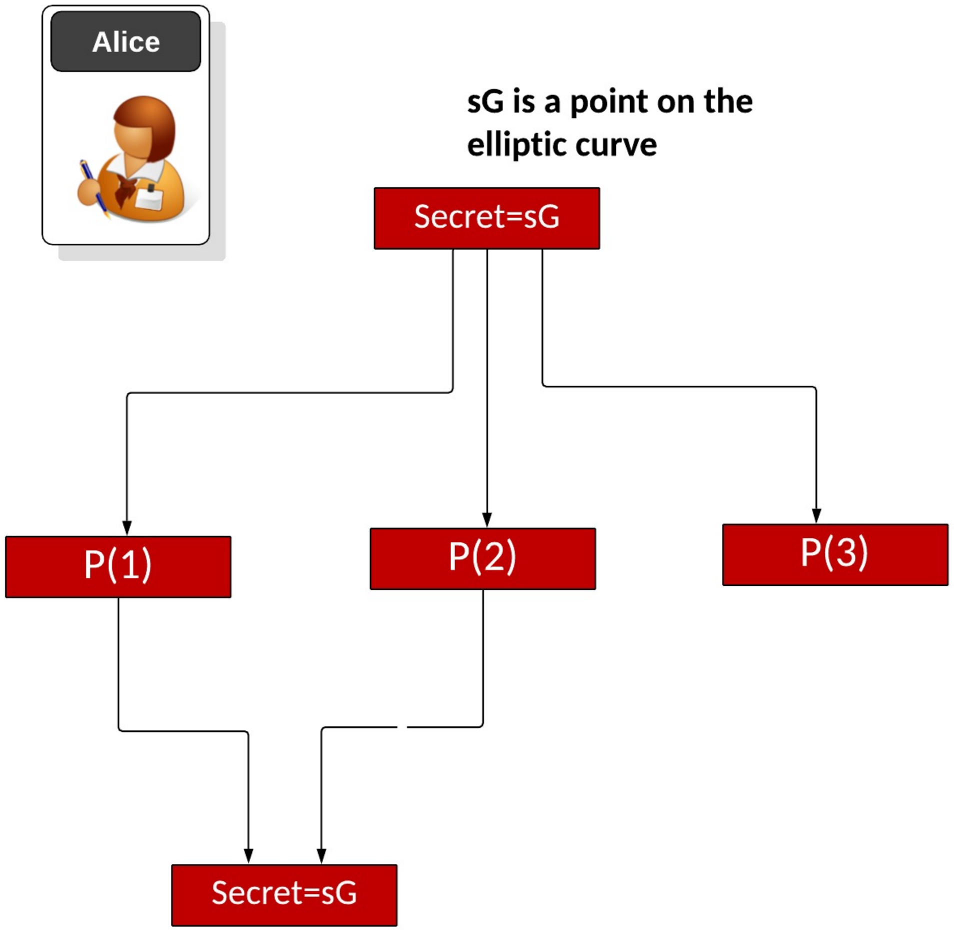


```
Secret equation:
    2
20 x - 19 x + 10

Secret:  10
Bob:  11
Carol:  52
Dave:  133
Alice:  254

Secret equation:  [ 20. -19.  10.]
Secret:  10
```



```python
import numpy as np
import random
import sys

a = random.randint(20,20)
b = random.randint(-20,20)
secret = 10


if (len(sys.argv)>1):
        secret=int(sys.argv[1])


seq = [a,b,secret]
f = np.poly1d(seq)
print ("Secret equation:\n",f)

print ("\nSecret: ",f(0))
print ("Bob: ",f(1))
print ("Carol: ",f(2))
print ("Dave: ",f(3))
print ("Alice: ",f(4))


x=[1,2,3]
y=[f(1),f(2),f(3)]
res=np.polyfit(x,y,2)
print ("\nSecret equation: ",res)
print ("Secret: ",int(round(res[2],0)))
```

Sharing With Elliptic Curves



Alice

sG is a point on the elliptic curve

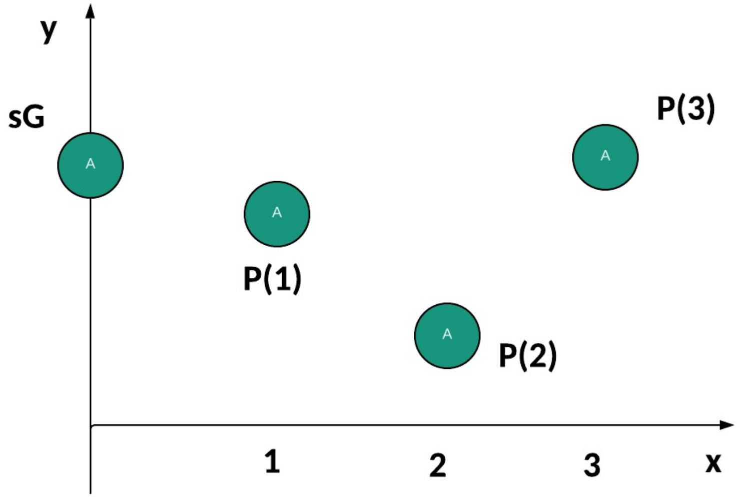
Secret=sG

P(1)

P(2)

P(3)

Secret=sG

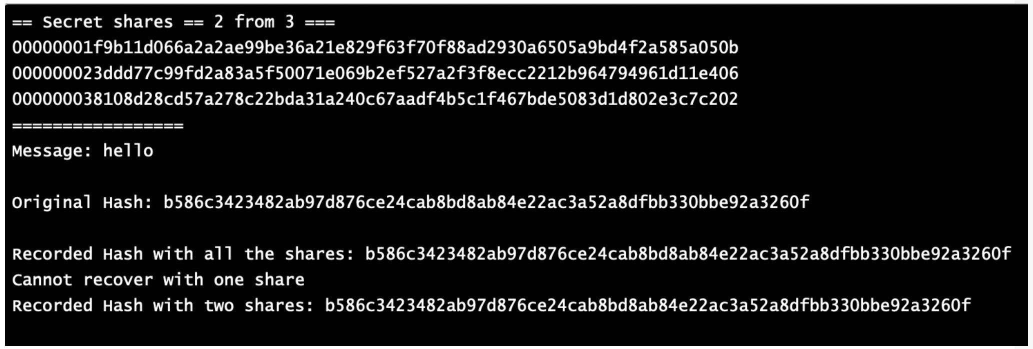


(t,n) polynomial secret shares

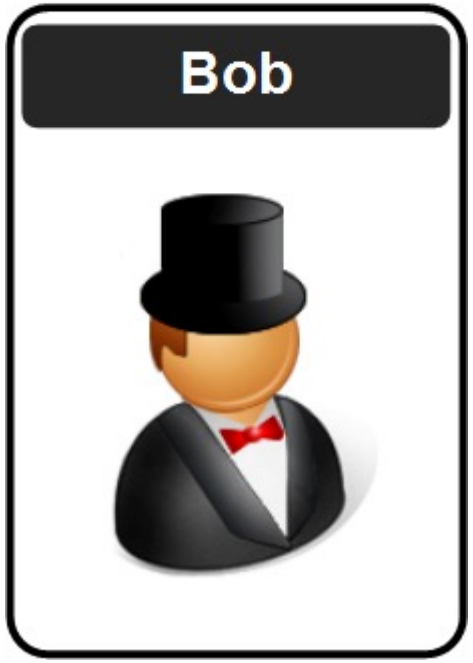
$$P(x) = p_0+p_1x+p_2x^2+\ldots$$

Integration with Golang

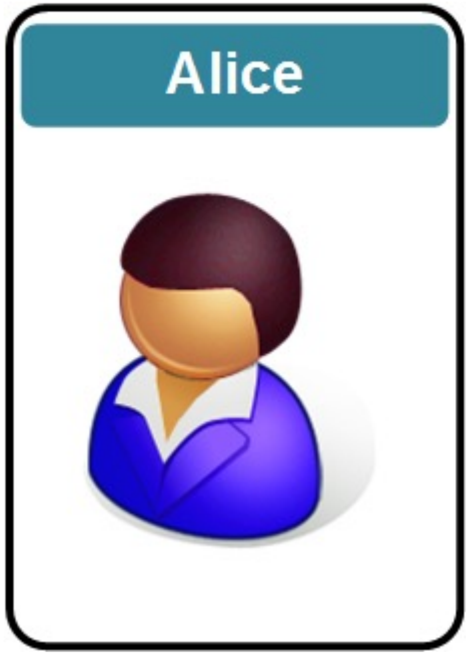
```go
curve := curves.ED25519()
scheme, _ := sharing.NewShamir(t, n, curve)

shares, _ := scheme.Split(curve.NewScalar().Hash([]byte(msg)), crand.Reader)

fmt.Printf("== Secret shares == %d from %d ===\n", t, n)
for _, s := range shares {
        fmt.Printf("%x ", s.Bytes())
}
fmt.Printf("\n=================\n")

mysecret := curve.NewScalar().Hash([]byte(msg))

fmt.Printf("Message: %s\n", msg)
fmt.Printf("\nOriginal Hash: %x\n\n", mysecret.Bytes())

secret, err := scheme.Combine(shares...)
if err == nil {
        fmt.Printf("Recorded Hash with all the shares: %x\n", secret.Bytes())
} else {
        fmt.Printf("Cannot recover with all shares\n")
}

secret, err = scheme.Combine(shares[0])
if err == nil {
        fmt.Printf("Recorded Hash with one share: %x\n", secret.Bytes())
} else {
        fmt.Printf("Cannot recover with one share\n")
}
```



```
== Secret shares == 2 from 3 ===
00000001f9b11d066a2a2ae99be36a21e829f63f70f88ad2930a6505a9bd4f2a585a050b
000000023ddd77c99fd2a83a5f50071e069b2ef527a2f3f8ecc2212b964794961d11e406
000000038108d28cd57a278c22bda31a240c67aadf4b5c1f467bde5083d1d802e3c7c202
=================
Message: hello

Original Hash: b586c3423482ab97d876ce24cab8bd8ab84e22ac3a52a8dfbb330bbe92a3260f

Recorded Hash with all the shares: b586c3423482ab97d876ce24cab8bd8ab84e22ac3a52a8dfbb330bbe92a3260f
Cannot recover with one share
Recorded Hash with two shares: b586c3423482ab97d876ce24cab8bd8ab84e22ac3a52a8dfbb330bbe92a3260f
```



Shamir Secret Shares and Elliptic Curves



Prof Bill Buchanan OBE, Blockpass ID Lab

http://asecuritysite.com

BLOCKPASS IDENTITY LAB

World-leading Collaboration between Blockpass IDN and Edinburgh Napier University