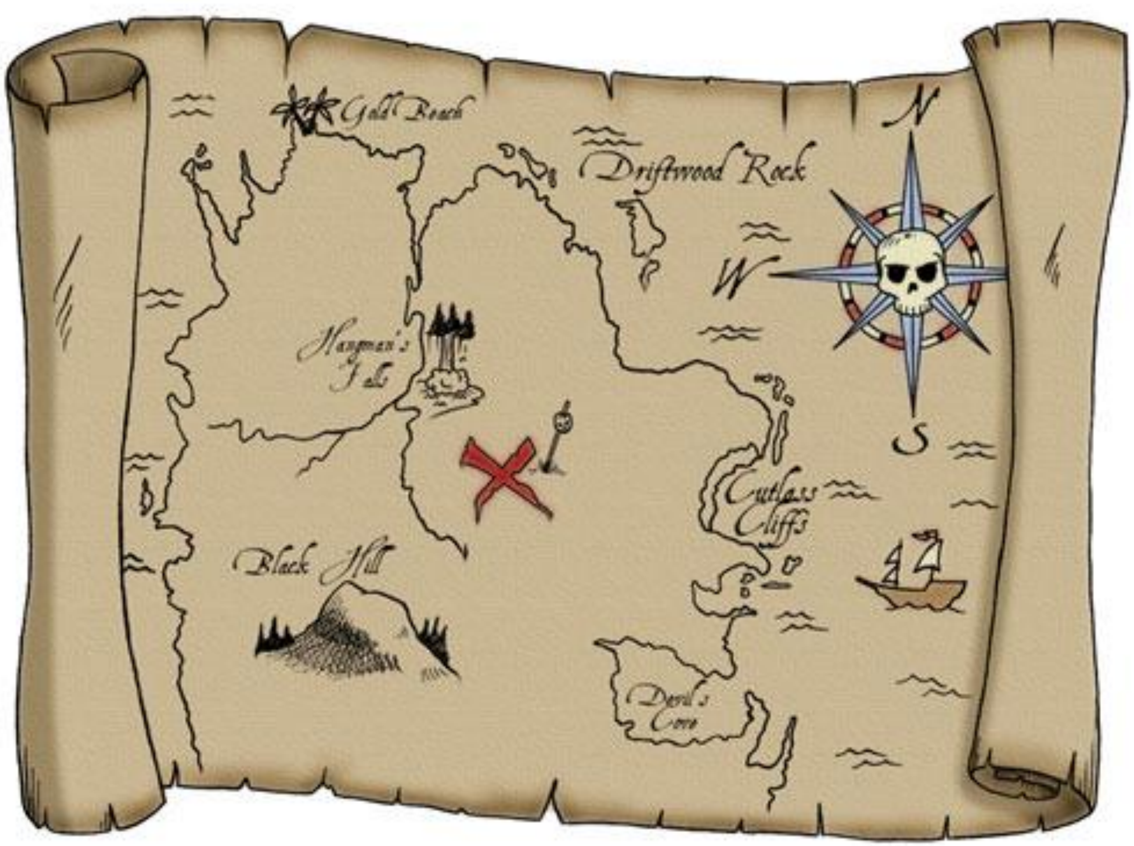
Threshold Cryptography

Cloud Security Mechanisms

Björn Groneberg - Summer Term 2013


Gold Beach
Driftwood Rock
N
W
S
Hangman's Falls
Black Hill
Cutlass Cliffs
Devil's Cove

?

Threshold Cryptography

- Sharing Secrets
  - Treasure Map
  - Sharing keys on multiple server
- Threshold Encryption
  - Protect top secret document, only group of people can decrypt it
- Threshold Signature
  - Signing checks
- E-Voting
  - Do not trust only one voting authority

Threshold Cryptography

1. Basic Maths
2. Lagrange Polynomial Interpolation
3. Shamir‘s Secret Sharing
4. Elgamal Encryption
5. Threshold Elgamal
6. Threshold RSA
7. E-Voting

Basic Maths

- p is a prime ☺
- modulo operator mod:
  - find remainder of division of two numbers

$$20 : 6 = 18\ R{:}\ 2 \Rightarrow 20 \bmod 6 = 2$$

- modulo congruent $\equiv$
  - two numbers are congruent modulo m if they have the same remainder by the division of m

$$20 \bmod 6 = 2 \text{ and } 14 \bmod 6 = 2 \Rightarrow 20 \equiv 14 \bmod 6$$

Basic Maths

- Residue class
  - Collect all integers which are congruent given a modulo m
  - Example: $\mod 6$

$$[0]_6 = \{\dots, -6, 0, 6, 12, 18, \dots\} \qquad [1]_6 = \{\dots, -5, 1, 7, 13, 19, \dots\}$$

$$[2]_6 = \{\dots, -4, 2, 8, 14, 20, \dots\} \qquad [3]_6 = \{\dots, -3, 3, 9, 15, 21, \dots\}$$

$$[4]_6 = \{\dots, -2, 4, 10, 16, 22, \dots\} \qquad [5]_6 = \{\dots, -1, 5, 11, 17, 23 \;\dots\}$$

- Residue class system (ring) $\mathbb{Z}_n$
  - Collect all residue classes and have two operations
  - Example:

$$\mathbb{Z}_6 = \{[0]_6, [1]_6, [2]_6, [3]_6, [4]_6, [5]_6\} = \{0, 1, 2, 3, 4, 5\}$$

$$5 + 4 = 3 \qquad 3 + 4 = 1 \qquad 9 + 12 = 5 \qquad \mod 6$$

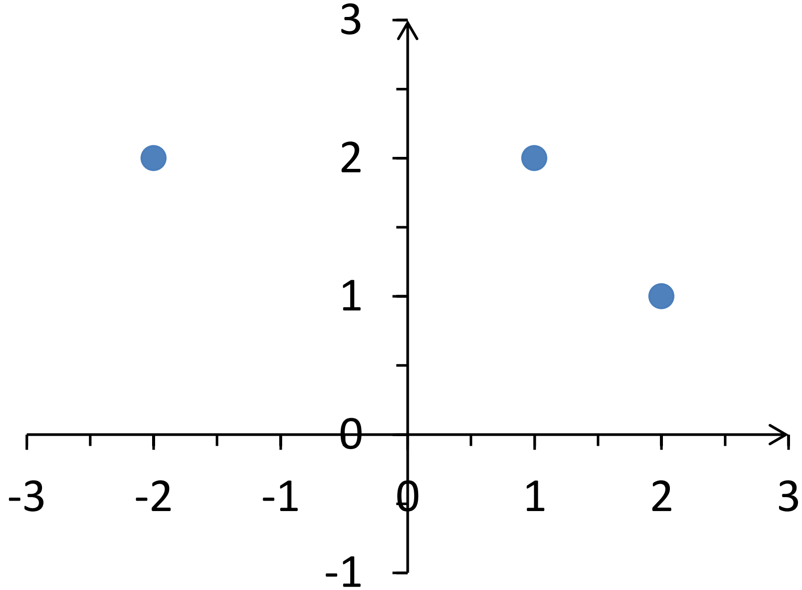
$$5 \cdot 4 = 2 \qquad 3 \cdot 4 = 0 \qquad 9 \cdot 12 = 0 \qquad \mod 6$$

Threshold Cryptography

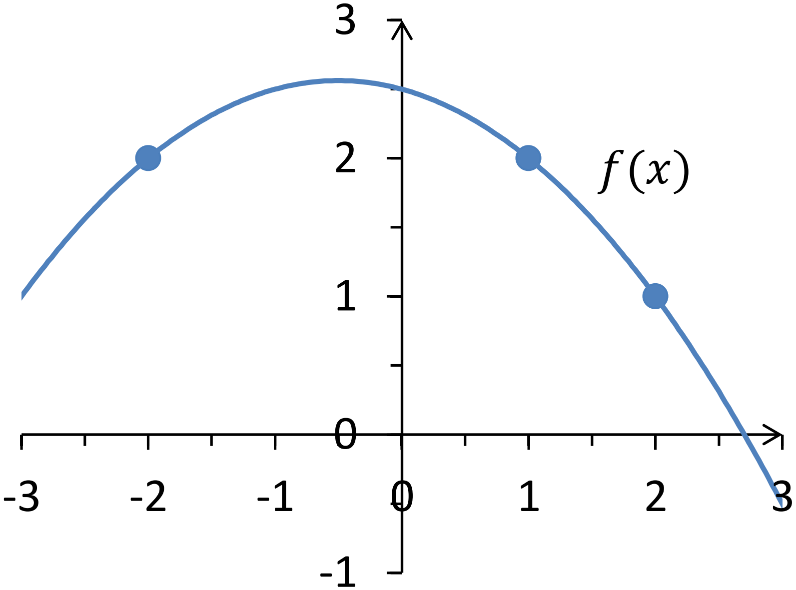
1. Basic Maths
2. **Lagrange Polynomial Interpolation**
3. Shamir‘s Secret Sharing
4. Elgamal Encryption
5. Threshold Elgamal
6. Threshold RSA
7. E-Voting

Lagrange Polynomial Interpolation

- Find polynomial to given set of points

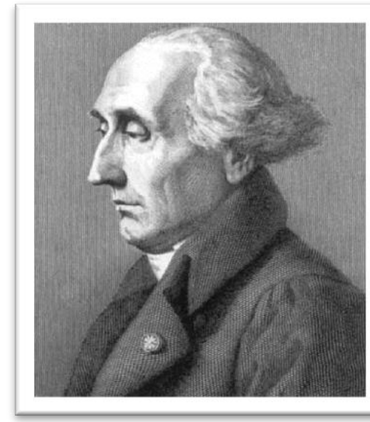


$(1,2), (-2,2), (2,1)$



$f(x) = ?$

Lagrange Polynomial Interpolation



Joseph-Louis Lagrange

Interpolate polynomial function out of given points

Given: $k + 1$ data points:

$(x_0, y_0), \dots, (x_j, y_j), \dots, (x_k, y_k)$

where no two x_j are the same

Lagrange polynomial interpolation is:

$$L(x) \coloneqq \sum_{j=0}^{k} y_j \ell_j \; = \; y_0 \ell_1 + \dots + y_j \ell_j + \dots + y_k \ell_k$$

where ℓ_j is Lagrange basis polynomials:

$$\ell_j \; \coloneqq \; \prod_{\substack{0 \le m \le k \\ m \ne j}} \frac{x - x_m}{x_j - x_m} \; = \; \frac{x - x_0}{x_j - x_0} \dots \frac{x - x_{j-1}}{x_j - x_{j-1}} \frac{x - x_{j+1}}{x_j - x_{j+1}} \dots \frac{x - x_k}{x_j - x_k}$$

[La13]

Lagrange Example

- Given Points: $(1,2), (-2,2), (2,1) \qquad k = 2$
- Calculate Lagrange basis polynomials

$$\ell_0 := \frac{(x-x_1)}{(x_0-x_1)}\frac{(x-x_2)}{(x_0-x_2)} = \frac{(x+2)}{(1+2)}\frac{(x-2)}{(1-2)} = -\frac{1}{3}(x^2-4)$$

$$\ell_1 := \frac{(x-x_0)}{(x_1-x_0)}\frac{(x-x_2)}{(x_0-x_2)} = \frac{(x-1)}{(-2-1)}\frac{(x-2)}{(-2-2)} = \frac{1}{12}(x^2-3x+2)$$

$$\ell_2 := \frac{(x-x_0)}{(x_2-x_0)}\frac{(x-x_1)}{(x_2-x_1)} = \frac{(x-1)}{(2-1)}\frac{(x+2)}{(2+2)} = \frac{1}{4}(x^2+x-2)$$

$$L(x) := \sum_{j=0}^{k} y_j \ell_j$$

$$\ell_j := \prod_{\substack{0\le m\le k \\ m\neq j}} \frac{x-x_m}{x_j-x_m}$$

- Calculate Lagrange polynomial:

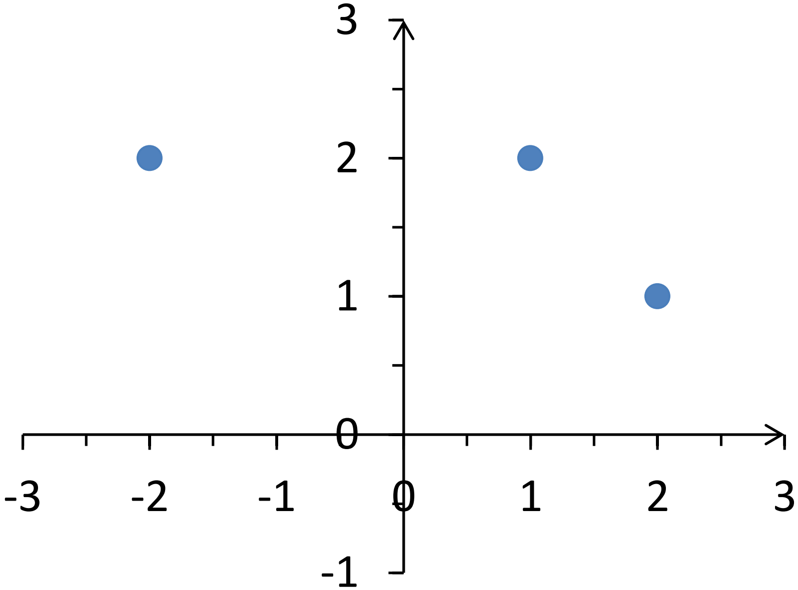
$$L(x) = y_0\ell_0 + y_1\ell_1 + y_2\ell_2$$

$$L(x) = 2\cdot -\frac{1}{3}(x^2-4) + 2\cdot\frac{1}{12}(x^2-3x+2) + 1\cdot\frac{1}{4}(x^2+x-2) = -\boldsymbol{\frac{1}{4}x^2 - \frac{1}{4}x + \frac{5}{2}}$$

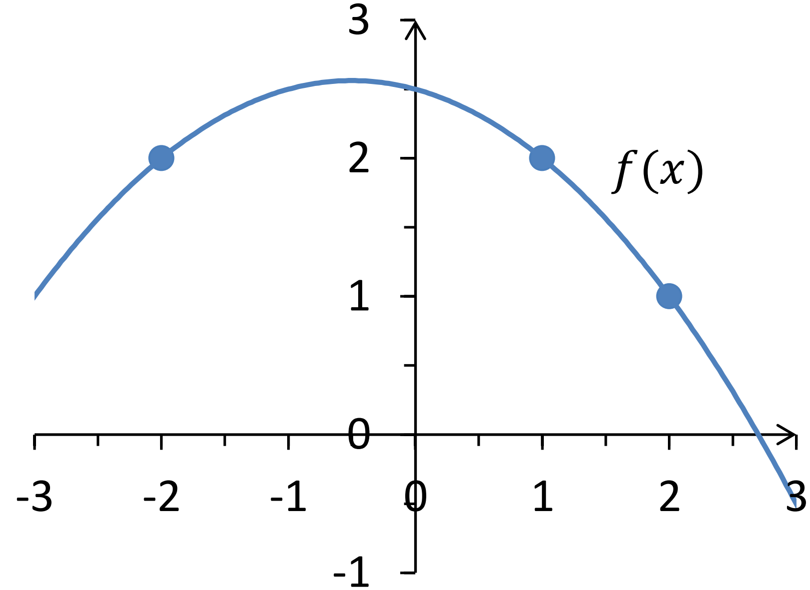
[La13]

Lagrange Polynomial Interpolation

- Find polynom to given set of points



$(1,2), (-2,2), (2,1)$



$$f(x) = -\frac{1}{4}x^2 - \frac{1}{4}x + \frac{5}{2}$$

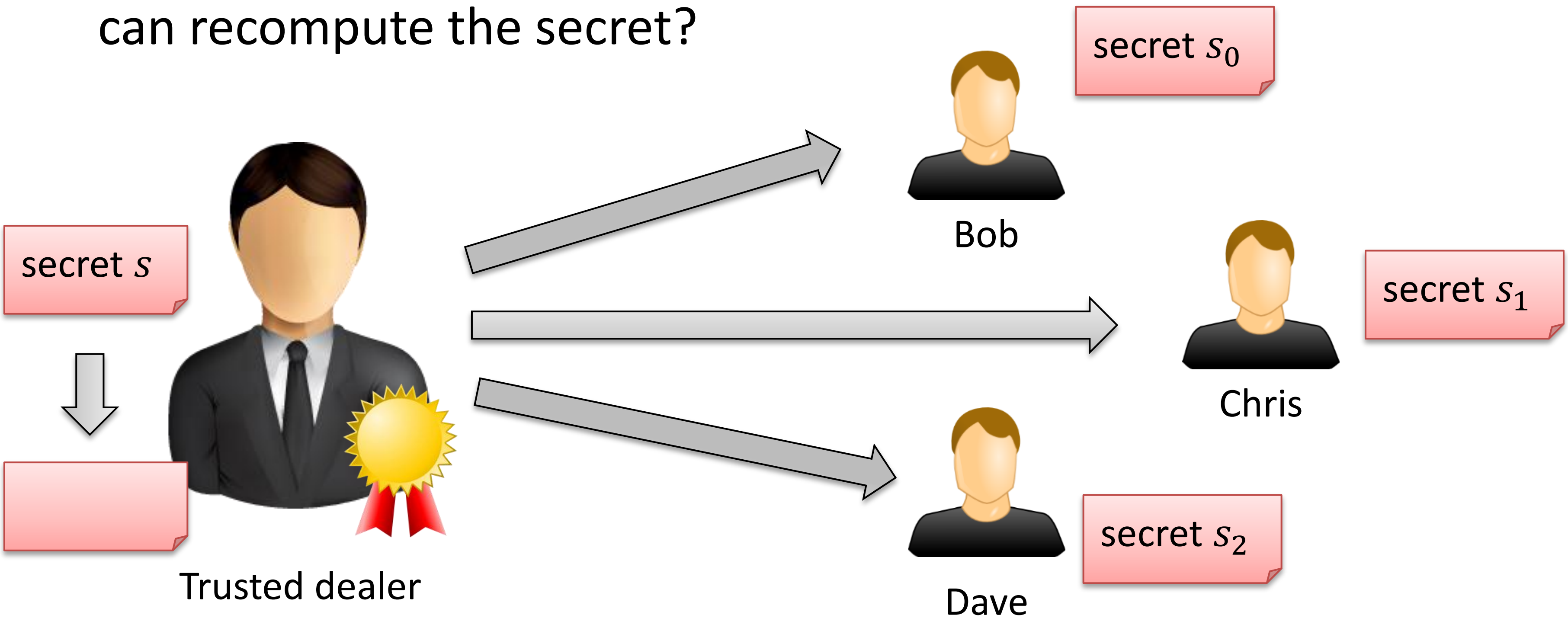
Threshold Cryptography

1. Basic Maths
2. Lagrange Polynomial Interpolation
3. **Shamir's Secret Sharing**
4. Elgamal Encryption
5. Threshold Elgamal
6. Threshold RSA
7. E-Voting

Secret Sharing

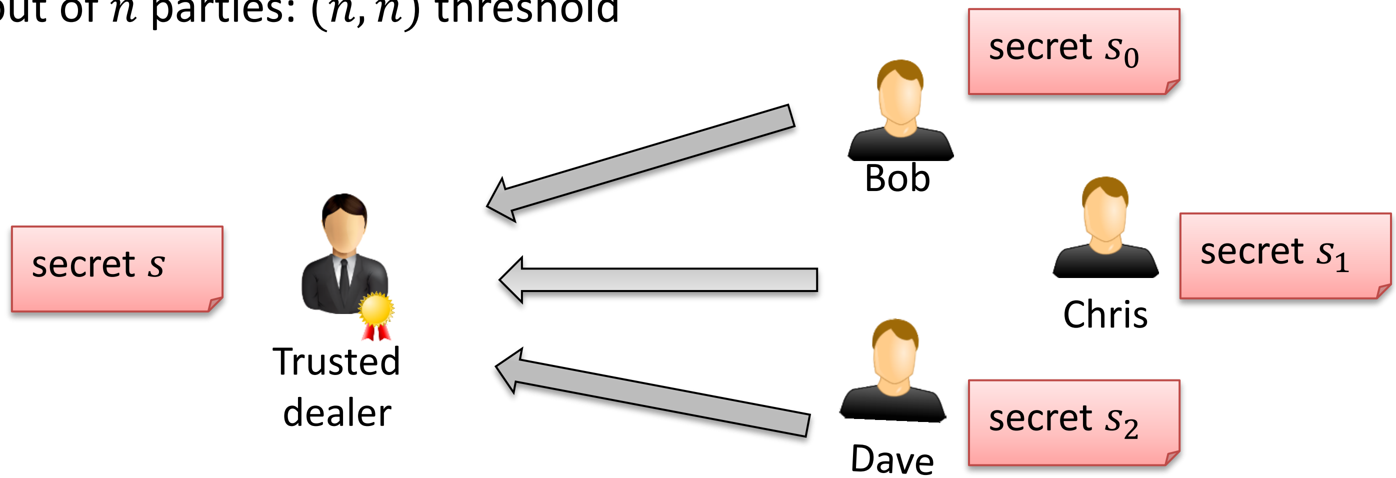
- How to distribute secret s to n parties in that way, that
  - Only all n parties together *or*
  - k out of n parties

  can recompute the secret?



Secret Sharing

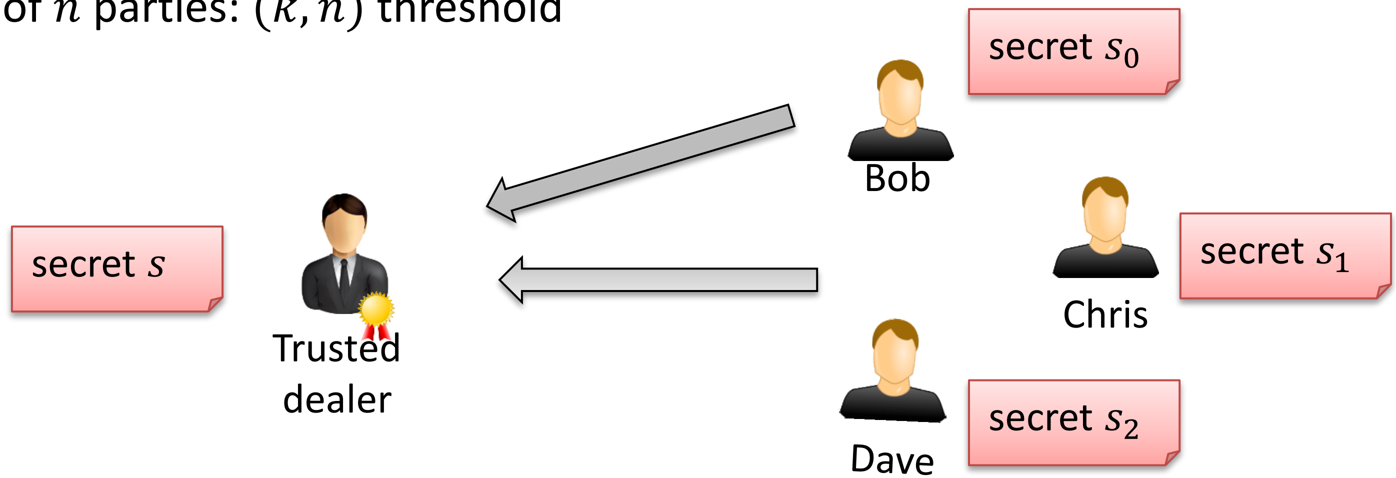
- Recomputation of the secret
  - all n out of n parties: (n, n) threshold



  - $n-1, n-2, \ldots$ parties should not be able to recompute the secret
  - Every party (or group of parties) should not be able to retreive any information about the global secret from their own secret(s)

Secret Sharing

- Recomputation of the secret
  - k out of n parties: (k, n) threshold



  - $k-1, \mathrm{k}-2, \ldots$ parties should not be able to recompute the secret
  - Every party (or group of parties) should not be able to retreive any information about the global secret from their own secret(s)

Secret Sharing

- Real world‘s solution:
  - Multiple locks with keys → heavy key ring

- Naive solution (bad):
  - Split secret in parts:

    1873 7632 8732 3253 2312

    1873 | 7632 | 8732 | 3253 | 2312

  - Disadvantage:
    - needs (n, n) threshold
    - $n-1$ out of n parties dramatically reduce possible keys

Shamir‘s Secret Sharing

- Published 1979 by Adi Shamir
- (k, n) threshold sharing
- Based on Lagrange polynomials

- Dealing Algorithm:
  - Given: (k, n) threshold and secret $s \in \mathbb{Z}_q$
  - Randomly choose $k-1$ coefficients $a_1, \dots, a_{k-1}$
  - Set $a_0 := s$
  - Build polynomial $f(x) = \ a_0 + \ a_1 x + a_2 x^2 + a_{k-1} x^{k-1}$
  - Set $i = 1, \dots, n$ and calculate Points $s_i = \big(i, f(i)\big) \bmod q$
  - Every party gets (at least) one point s_i

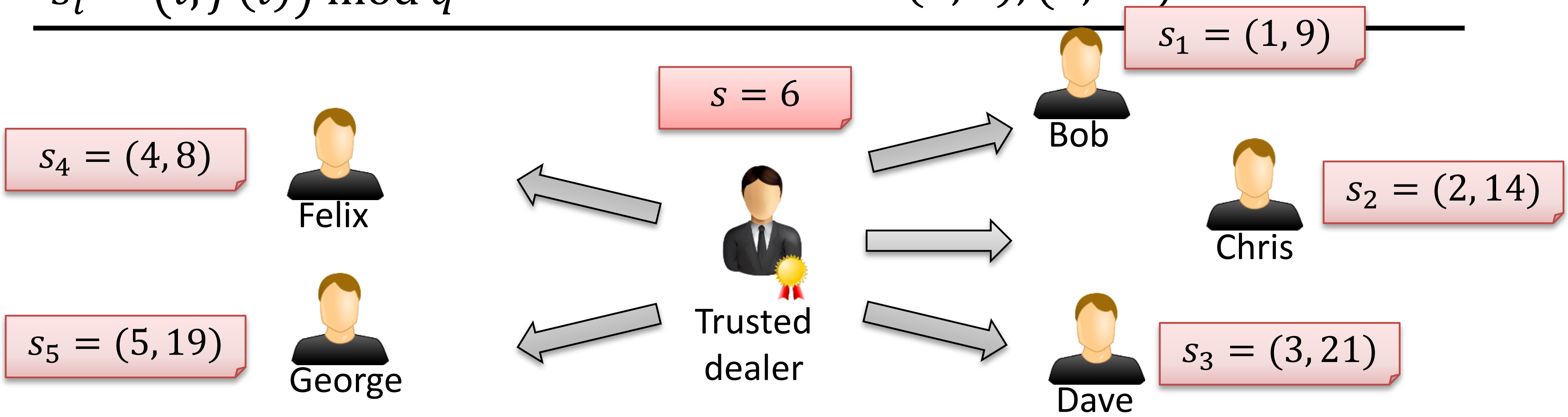
Adi Shamir –
The „S“ in RSA

[Sha79]

Shamir‘s Secret Sharing - Example

- Dealing Algorithm

Given: (k, n) **and secret** $s \in \mathbb{Z}_q$	$(3, 5)$ **threshold** $s = 6 \in \mathbb{Z}_{22}$
Randomly $k-1$: $a_1, \dots, a_{k-1}$	$a_1 = 2 \;\; a_2 = 1$
Set $a_0 := s$	$a_0 = 6$
$f(x) = a_0 + a_1 x + a_2 x^2 + a_{k-1} x^{k-1}$	$f(x) = x^2 + 2x + 6$
$i = 1, \dots, n$ calculate $s_i = (i, f(i)) \bmod q$	$(1, 9)\ (2, 14), (3, 21),$ $(4, 8), (5, 19)$



Shamir‘s Secret Sharing

- Recomputation
  - Given: k Points $s_i = (x_i, y_i)$
  - Goal: find $f(x) = a_0 + a_1 x + a_2 x^2 + a_{k-1} x^{k-1}$
    with $f(0) = a_0$ as the secret
  - Using $f(x) = L(x)$,
    $S \subseteq \{1, \dots, n\}, |S| = k$ and calculate

$$f(0) = L(0) = \sum_{j \in S} y_j \ell_{j,0,S} \bmod q$$

with $\ell_{j,0}$ as Lagrange basis polynomials with $x = 0$ and S:

$$\ell_{j,0,S} := \prod_{\substack{m \in S \\ m \neq j}} \frac{-x_m}{x_j - x_m} \bmod q$$

Lagrange:

$$L(x) := \sum_{j=0}^{k} y_j \ell_j$$

$$\ell_j := \prod_{\substack{0 \leq m \leq k \\ m \neq j}} \frac{x - x_m}{x_j - x_m}$$

[Sha79]

Shamir's Secret Sharing - Example

- Recomputation of basis polynomials:

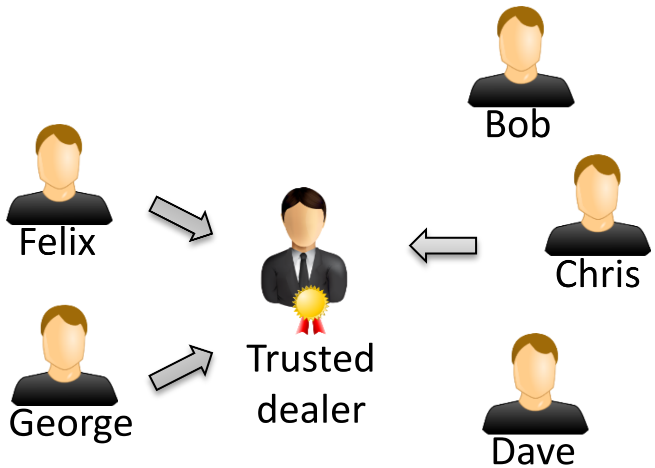
$$\ell_{2,0,\{2,4,5\}} = \frac{-x_4}{(x_2-x_4)}\frac{-x_5}{(x_2-x_5)} = \frac{-4}{(2-4)}\frac{-5}{(2-5)} = 10 \cdot 3^{-1} = 10 \cdot 15 = 18 \mod 22$$

$$\ell_{4,0,\{2,4,5\}} = \frac{-x_2}{(x_4-x_2)}\frac{-x_5}{(x_4-x_5)} = \frac{-2}{(4-2)}\frac{-5}{(4-5)} = -5 = 17 \mod 22$$

$$\ell_{5,0,\{2,4,5\}} = \frac{-x_2}{(x_5-x_2)}\frac{-x_4}{(x_5-x_4)} = \frac{-2}{(5-2)}\frac{-4}{(5-4)} = 8 \cdot 3^{-1} = 8 \cdot 15 = 10 \mod 22$$

$s_4 = (4, 8)$

$s_5 = (5, 19)$



$s_2 = (2, 14)$

„Shamir's Lagrange":

$$L(0) = \sum_{j \in S} y_j \ell_{j,0,S}$$

$$\ell_{j,0,S} := \prod_{\substack{m \in S \\ m \neq j}} \frac{-x_m}{x_j - x_m}$$

Shamir‘s Secret Sharing - Example

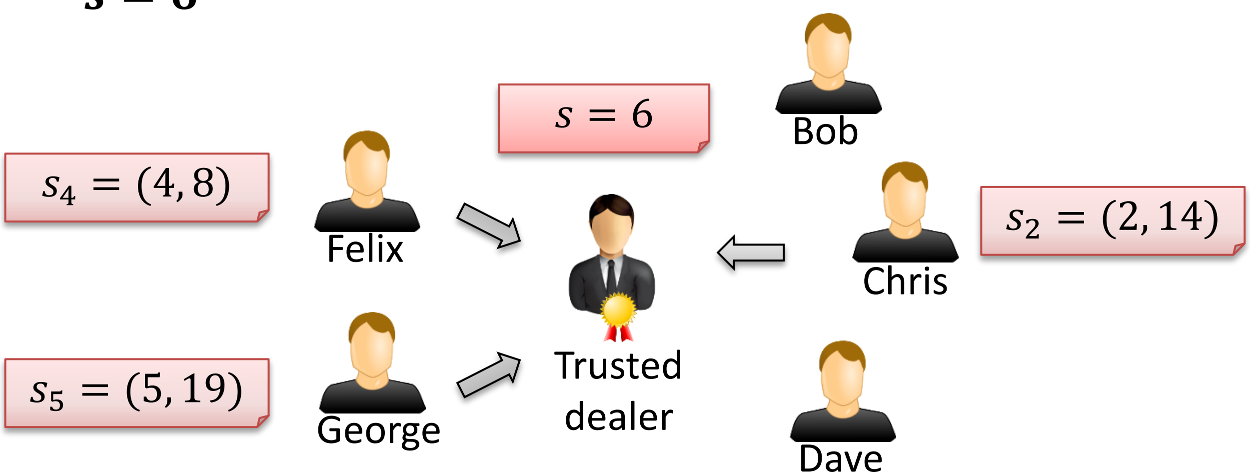
- Recomputation:

$\ell_{2,0,\{2,4,5\}} = 18, \quad \ell_{4,0,\{2,4,5\}} = 17, \quad \ell_{5,0,\{2,4,5\}} = 10$

$$s = L(0) = y_2 \cdot \ell_{2,0,\{2,4,5\}} + y_4 \cdot \ell_{4,0,\{2,4,5\}} + y_5 \cdot \ell_{5,0,\{2,4,5\}}$$

$$s = L(0) = 14 \cdot 18 + 8 \cdot 17 + 19 \cdot 10 \mod 22$$

$$\boldsymbol{s = 6}$$



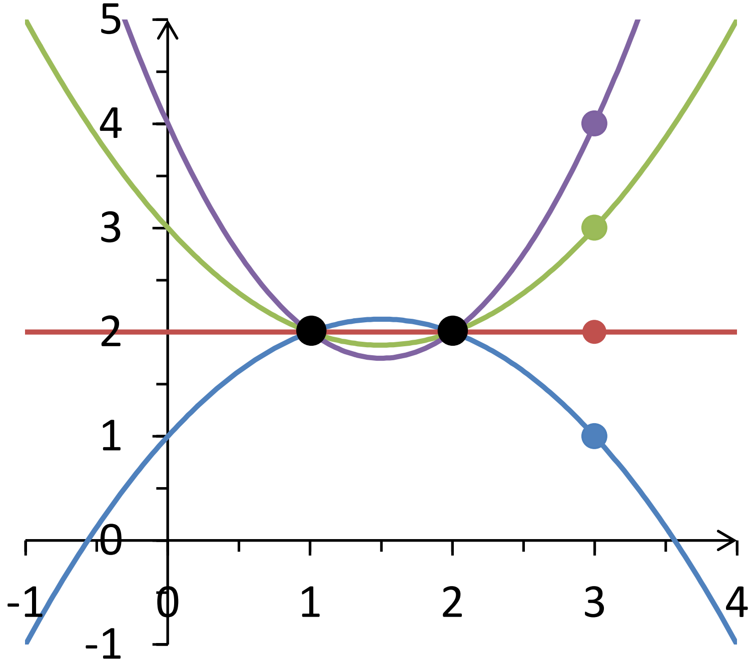
„Shamir‘s Lagrange“:

$$L(0) = \sum_{j \in S} y_j \ell_{j,0,S}$$

$$\ell_{j,0,S} := \prod_{\substack{m \in S \\ m \neq j}} \frac{-x_m}{x_j - x_m}$$

Shamir's Secret Sharing - Remarks

- Graphical Interpretation



- Flexibility
  - Increase n and compute new shares without affecting other shares
  - Removing existing shares (shares have to be destroyed)
  - Replace shares without changing the secret: new polynomial $f^*(x)$
  - One party can have more than one share

[Li04]

Threshold Cryptography

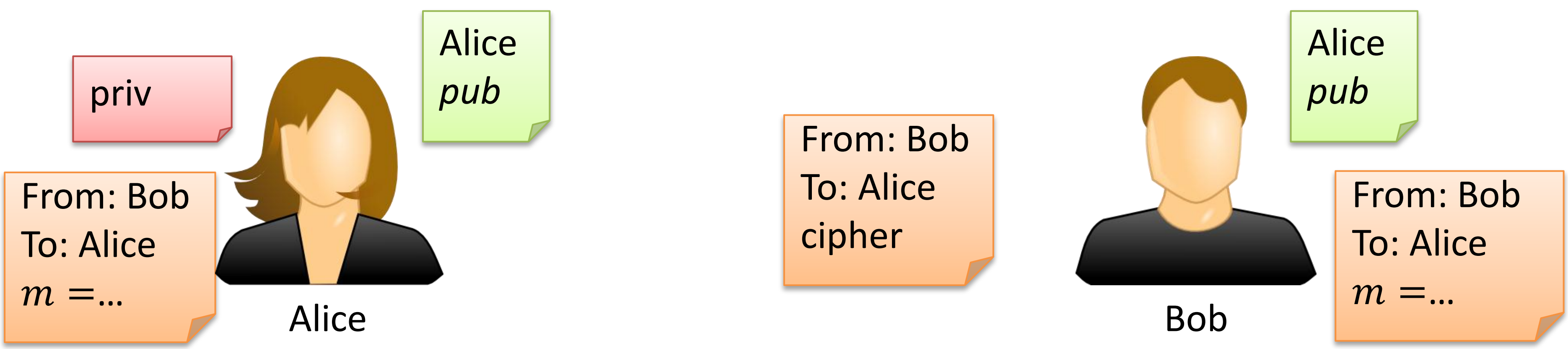
1. Basic Maths
2. Lagrange Polynomial Interpolation
3. Shamir‘s Secret Sharing
4. **Elgamal Encryption**
5. Threshold Elgamal
6. Threshold RSA
7. E-Voting

Elgamal Encryption

- Published 1985 by Taher Elgamal
- Based on Diffie-Hellman key exchange

- Public / private key encryption:
- Generation: pub, priv
- Encryption: cipher $= \mathrm{enc}_{\mathrm{pub}}(m)$
- Decryption: $m = \mathrm{dec}_{\mathrm{priv}}(\mathrm{cipher})$

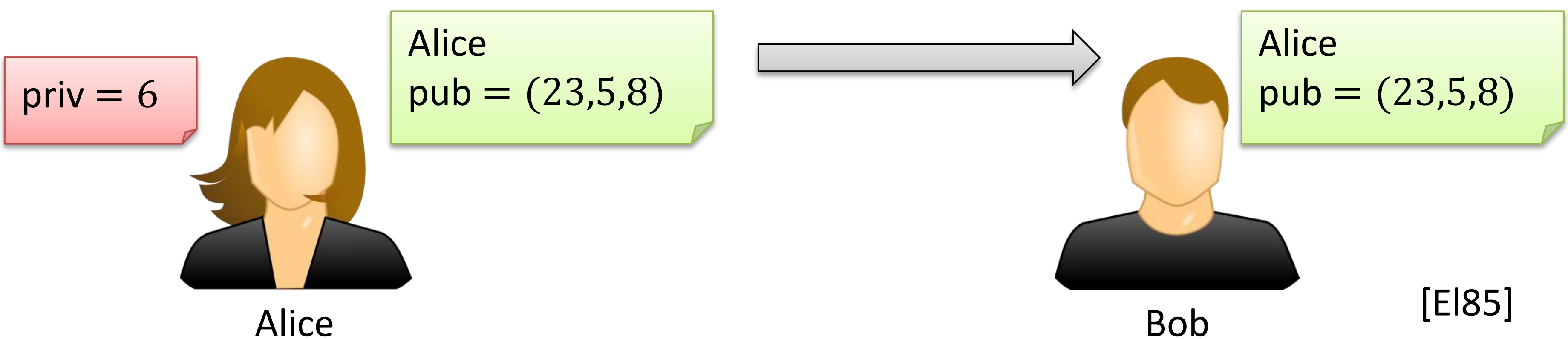
Taher Elgamal



Elgamal Encryption - Example

- Public / private key generation

1. large prime p with generator g	$p = 23 \quad g = 5$
2. randomly $a \in \{1, \dots, p-1\}$	$a = 6$
3. Calculate $A = g^a \ mod \ p$	$A = 5^6 = 8 \text{ mod } 23$
4. pub $= (p, g, A)$ priv = a	pub $= (23, 5, 8)$ priv $= 6$

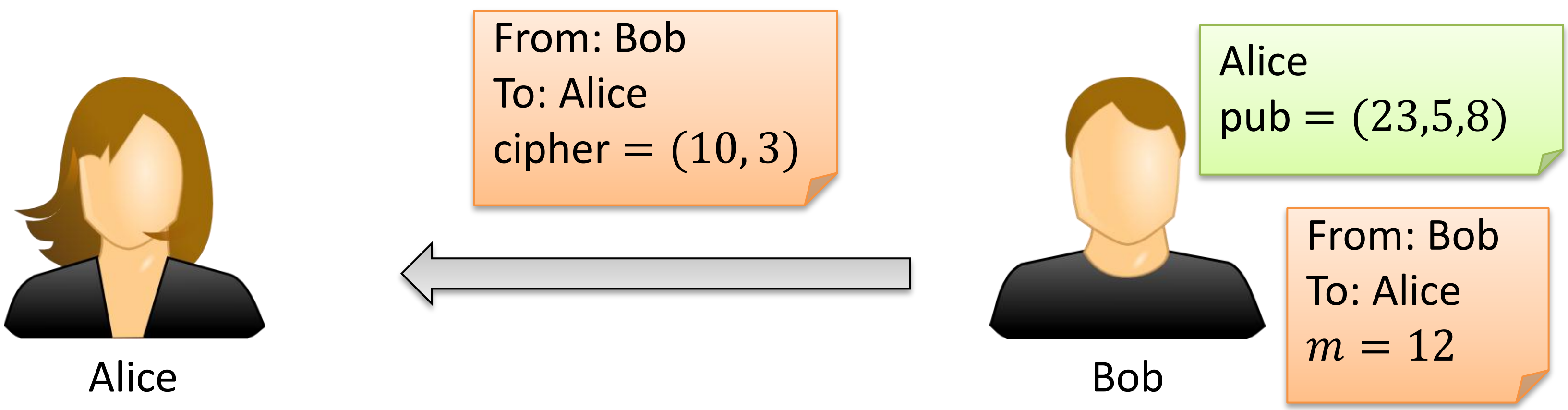


[El85]

Elgamal Encryption - Example

- Encryption

Given: message $m \in \{0, \dots, p-1\}$	$m = 12$
Randomly $b \in \{1, \dots, 1-p\}$	$b = 3$
Calculate $B = g^b \bmod p$ $c = A^b m \bmod p$	$B = 5^3 = 10 \bmod 23$ $c = 8^3 \cdot 12 = 3 \bmod 23$
Cipher text is $\text{cipher} = (B, c)$	$\text{cipher} = (10, 3)$



Elgamal Encryption - Example

- Decryption

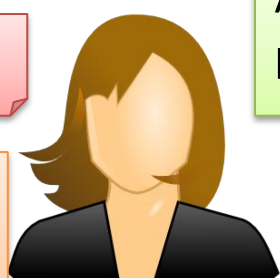
Given: cypher $= (\mathbf{B}, \mathbf{c})$ and priv $= a$	cypher $= (10{,}3)$ priv $= 6$
Calculate $x = p - 1 - a$	$x = 23 - 1 - 6 = 16$
Calculate $m = B^x c \bmod p$	$m = 10^{16} \cdot 3 = 12 \bmod 23$
Encrypted message m	$m = 12$

- General Idea: $m = (B^a)^{-1} \cdot c = B^{(p-1-a)} \bmod p$

a $= 6$

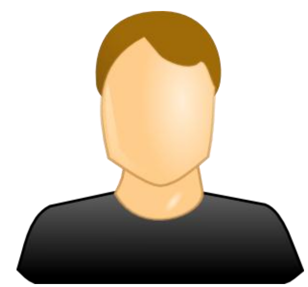
Alice
pub $= (23{,}5{,}8)$

From: Bob
To: Alice
$m = 12$



Alice

From: Bob
To: Alice
cipher $= (10, 3)$



Bob

[El85]

Threshold Cryptography

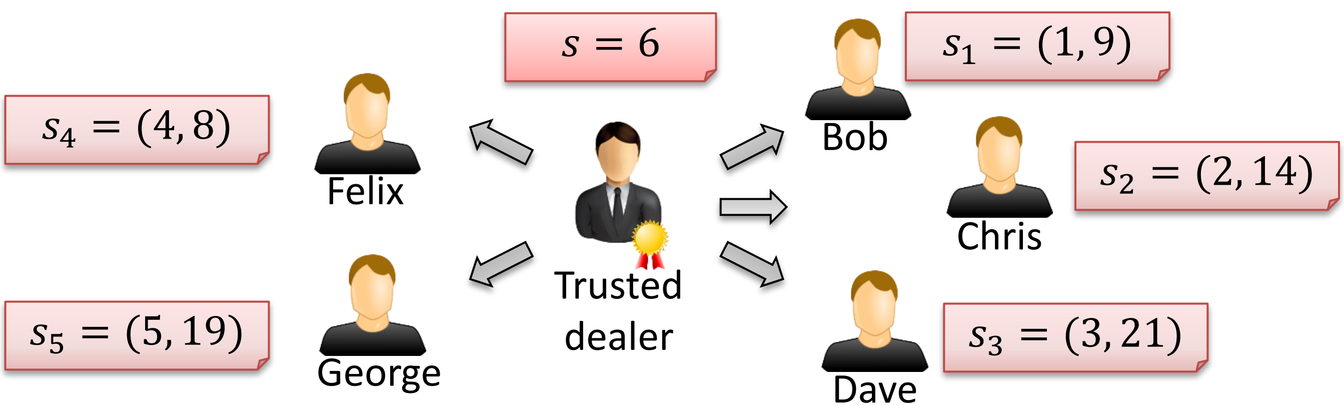
1. Basic Maths
2. Lagrange Polynomial Interpolation
3. Shamir‘s Secret Sharing
4. Elgamal Encryption
5. **Threshold Elgamal**
6. Threshold RSA
7. E-Voting

Threshold Elgamal

- Using Elgamal encryption scheme in a treshold environment
- Generation:
  - Generate pub $= (p, g, A)$ priv = a like normal **Elgamal encryption**
  - Share priv = a among n parties, using **Shamir's secret sharing** with $q = \varphi(p) =^{*} p - 1$
  - Every party j gets (at least) one point $s_j = (x_j, y_j)$

* if p is prime

Example: pub $= (23, 5, 8)$ priv $= 6$ (3,5)-threshold

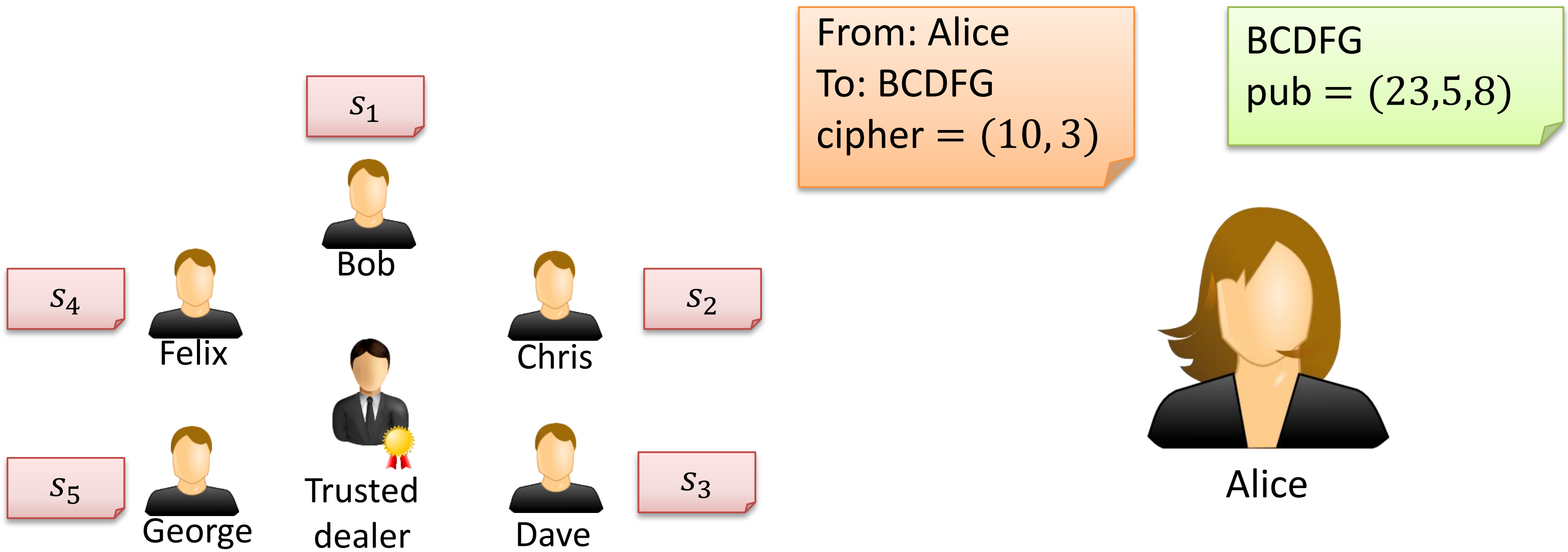


BCDFG
pub $= (23{,}5{,}8)$

[Ca06]

Threshold Elgamal

- Encryption
  - Normal Elgamal encryption with message m and pub $= (p, g, A)$

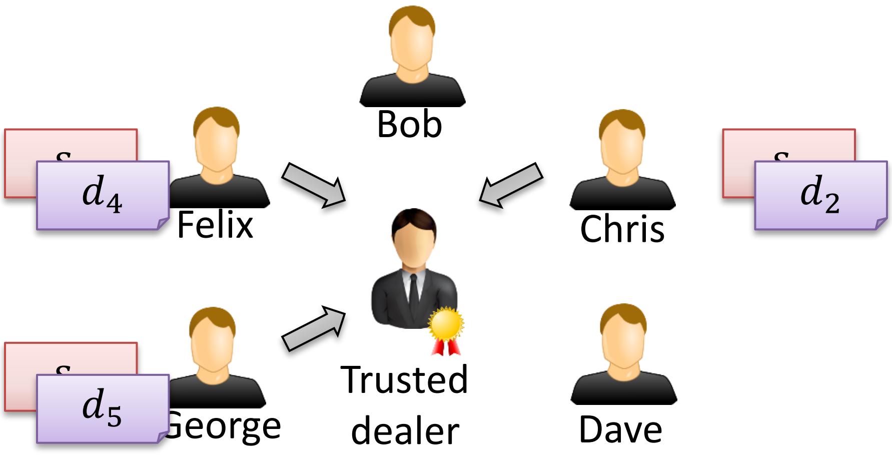


[Ca06]

Threshold Elgamal

- Decryption
  - Trusted dealer and every party can receive cipher $= (B, c)$
  - at least k parties have to compute decryption share $d_j = B^{y_j} \bmod p$
  - Trusted dealer can compute m with set S of $j \in \{1, \dots, n\}$ which returned their d_j
  - Party:
    $d_j = B^{y_j} \bmod p$
  - Trusted Dealer:

$$m = \left(\prod_{j \in S} {d_j}^{\ell_{j,0,S}} \right)^{-1} \cdot c \bmod p$$



[Ca06]

Threshold Elgamal - Example

- Decryption
  - Every party computes decryption share:

$$d_2 = B^{y_2} = 10^{14} = 12 \mod 23$$
$$d_4 = B^{y_5} = 10^{8} = 2 \mod 23$$
$$d_5 = B^{y_5} = 10^{19} = 21 \mod 23$$

  - Trusted dealer computes $\ell_{j,0,S}$:

$$\ell_{2,0,\{2,4,5\}} = 18$$
$$\ell_{4,0,\{2,4,5\}} = 17$$
$$\ell_{5,0,\{2,4,5\}} = 10$$

→ Shamir‘s secret sharing, slide 20

„Shamir‘s Lagrange“:

$$\ell_{j,0,S} := \prod_{\substack{m \in S \\ m \neq j}} \frac{-x_m}{x_j - x_m}$$

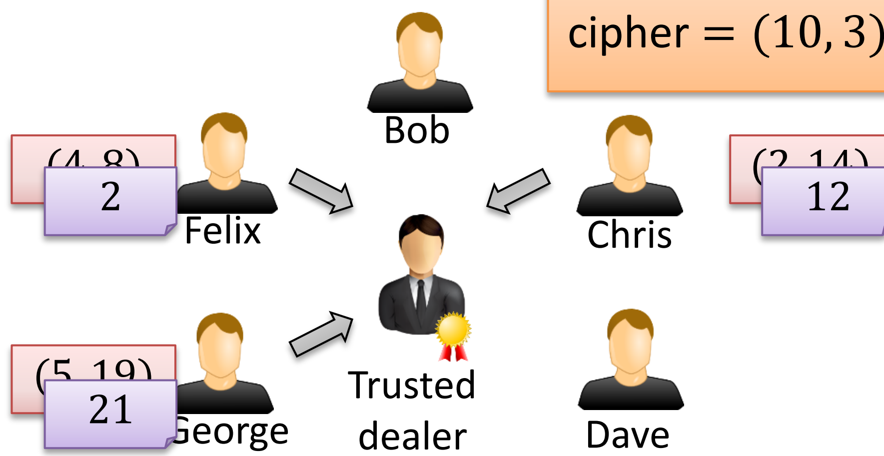
Threshold Elgamal

$$\text{cipher} = (B, c)$$

$$d_j = B^{y_j} \bmod p$$

$$m = \left(\prod_{j \in S} d_j^{\ell_{j,0,S}} \right)^{-1} \cdot c \bmod p$$

From: Alice
To: BCDFG
cipher = $(10, 3)$



Threshold Elgamal - Example

- Decryption

$$d_2 = 12,\ d_4 = 2,\ d_5 = 21$$
$$\ell_{2,0,\{2,4,5\}} = 18,\ \ell_{4,0,\{2,4,5\}} = 17,\ \ell_{5,0,\{2,4,5\}} = 10$$

  - Trusted dealer computes m:

$$m = \left(d_2^{\ell_{2,0,\{2,4,5\}}} \cdot d_4^{\ell_{4,0,\{2,4,5\}}} \cdot d_5^{\ell_{5,0,\{2,4,5\}}}\right)^{-1} \cdot c \mod p$$

$$m = (12^{18} \cdot 2^{17} \cdot 21^{10})^{-1} \cdot 3 \mod 23$$
$$m = (6)^{-1} \cdot 3 \mod 23$$
$$m = 4 \cdot 3 \mod 23$$

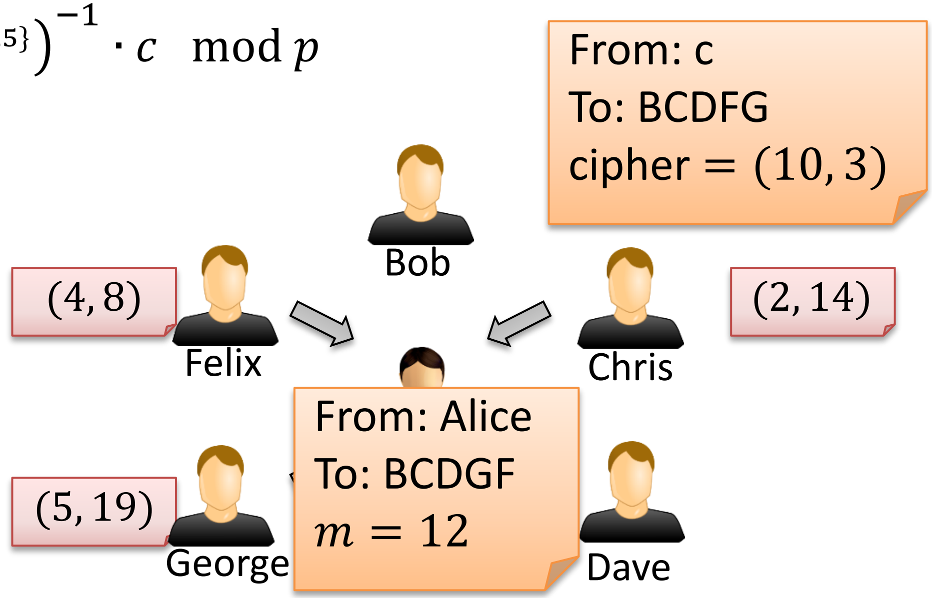
$$\boldsymbol{m = 12}$$

Note: $(6)^{-1} = 4 \mod 23$
(Extended Euclidean algorithm)

Threshold Elgamal

$\text{cipher} = (B, c)$

$$d_j = B^{y_j} \bmod p$$
$$m = \left(\prod_{j \in S} d_j^{\ell_{j,0,S}}\right)^{-1} \cdot c \bmod p$$

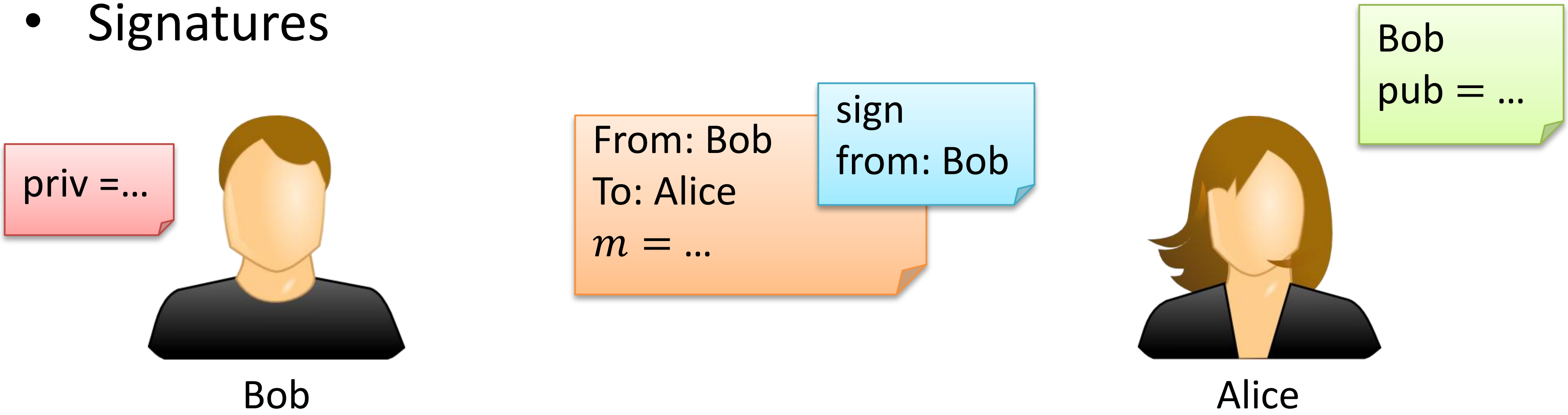


Threshold Cryptography

1. Basic Maths
2. Lagrange Polynomial Interpolation
3. Shamir‘s Secret Sharing
4. Elgamal Encryption
5. Threshold Elgamal
6. **Threshold RSA**
7. E-Voting

RSA Threshold Signatures

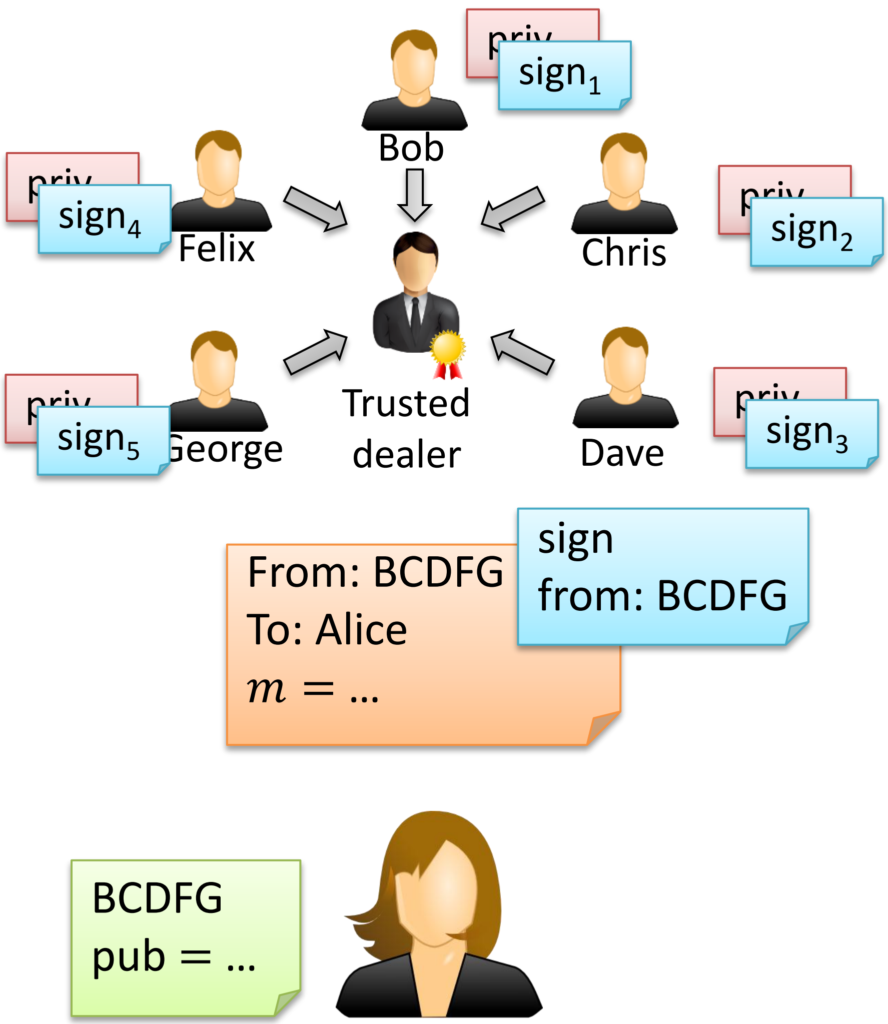
- Signatures



- Requires: Public / private key and hash function $H(x)$
- Sign a message:
  - Hash message m and encrypt with private key: $\text{sign} = \text{enc}_{\text{priv}}(H(m))$
- Verify signature
  - Decrypt signature with public key and check hash: $\text{dec}_{\text{pub}}(\text{sign}) \stackrel{?}{=} H(m)$

[Ca06]

RSA Threshold Signatures



- Every party signs with own private key
- Trusted dealer can compute global signature

Party i:

$$\mathrm{sign}_i = \mathrm{enc}_{\mathrm{priv}_i}(H(m))$$

Trusted dealer:

$$\mathrm{sign} = \mathrm{collect}(\mathrm{sign}_1, \ldots, \mathrm{sign}_n)$$

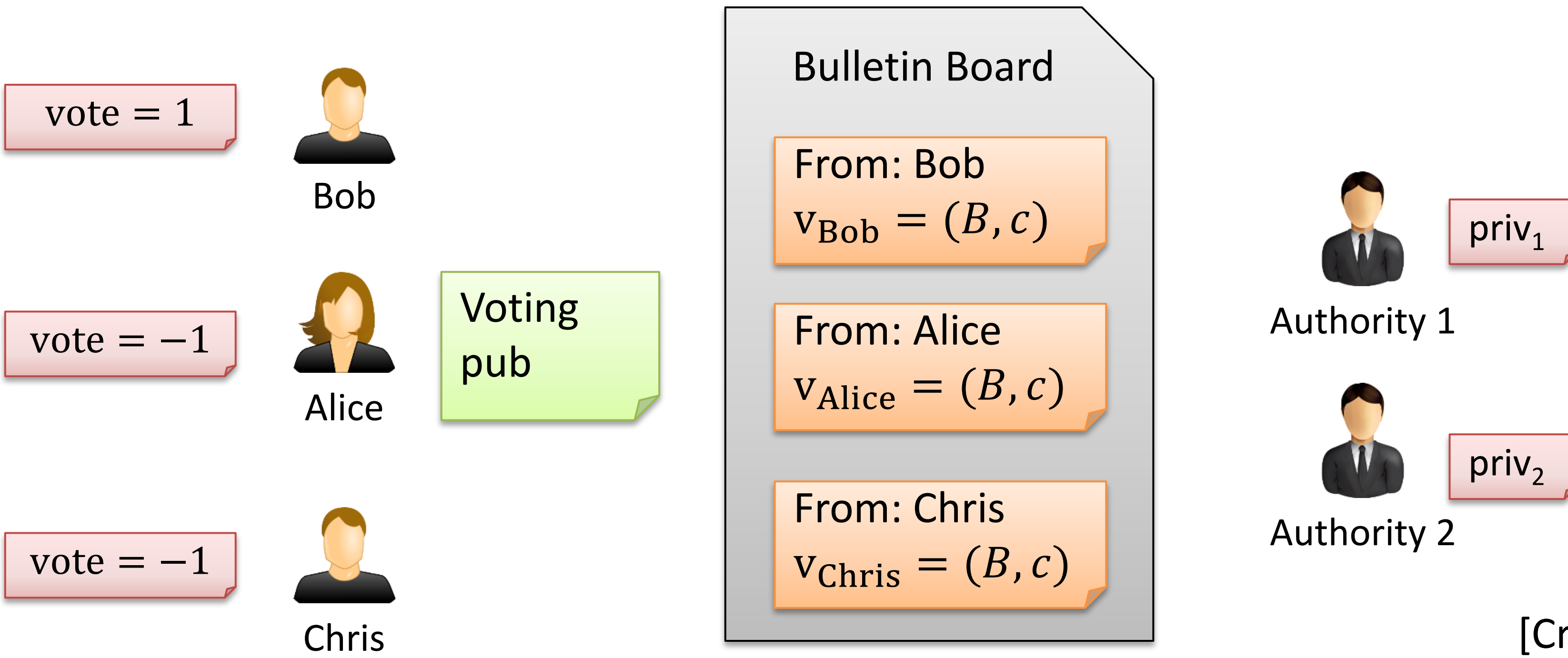
- V. Shoup: *"Practical threshold signatures"* shows threshold signature scheme with RSA [Sh]

Threshold Cryptography

1. Basic Maths
2. Lagrange Polynomial Interpolation
3. Shamir‘s Secret Sharing
4. Elgamal Encryption
5. Threshold Elgamal
6. Threshld RSA
7. **E-Voting**

E-Voting

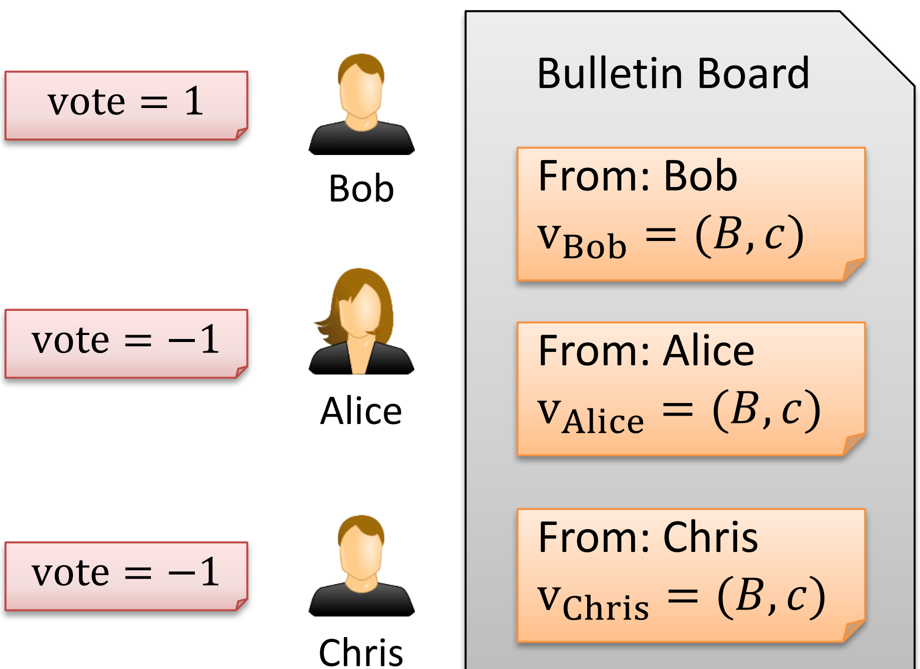
- Secret voting using Elgamal threshold encryption
- Voter encrypts vote with public key
- Private key is shared among voting authorities



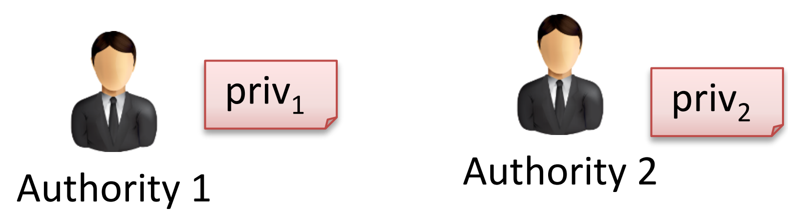
[Cr97]

E-Voting

- Voting authorities “counting” encrypted votes
- Decrypt result of “counting” with shared secrets



$$v_{\text{Result}} = \text{count}(v_{\text{Bob}}, v_{\text{Alice}}, v_{\text{Chris}})$$
$$v_{\text{Result}} = (B, c)$$



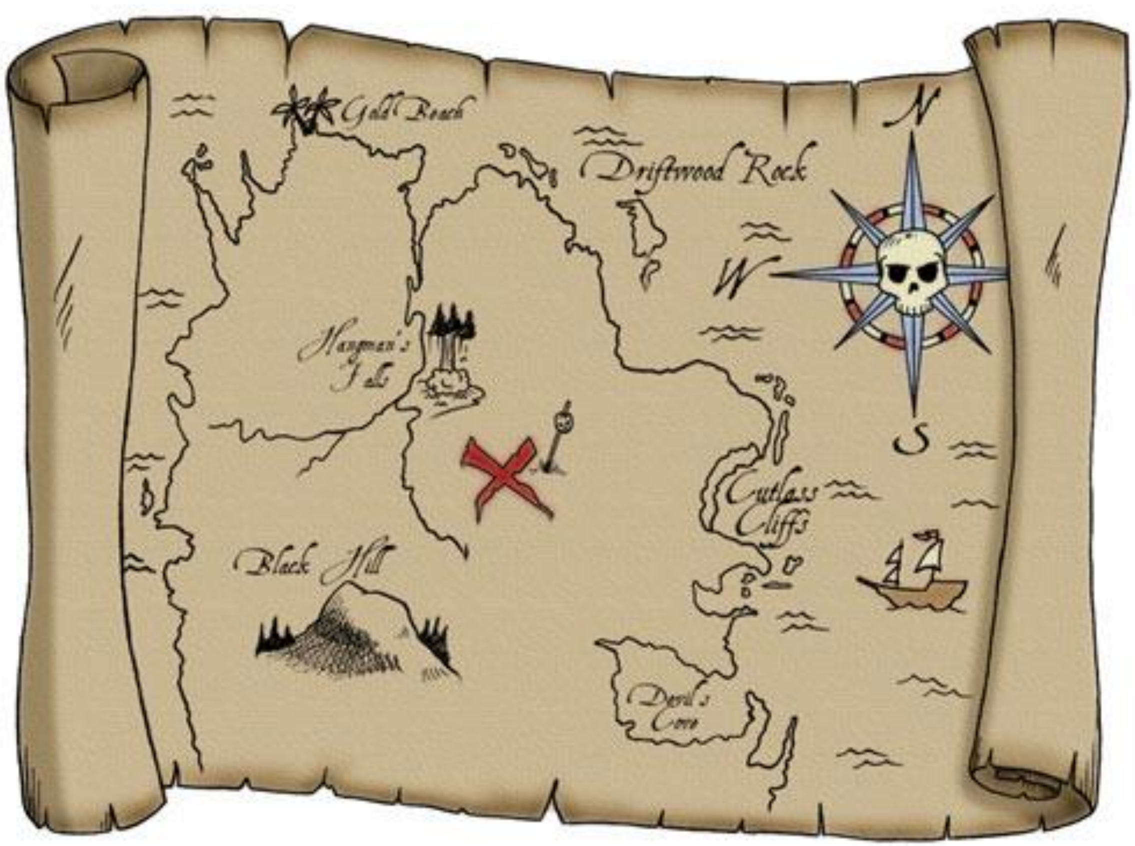
$$\text{Result} = \text{dec}_{\text{priv}}(v_{\text{Result}})$$
$$\text{Result} = -1$$

- Cramer, et. al.: "*A secure and optimally efficient multi-authority election scheme.*" [Cr97]

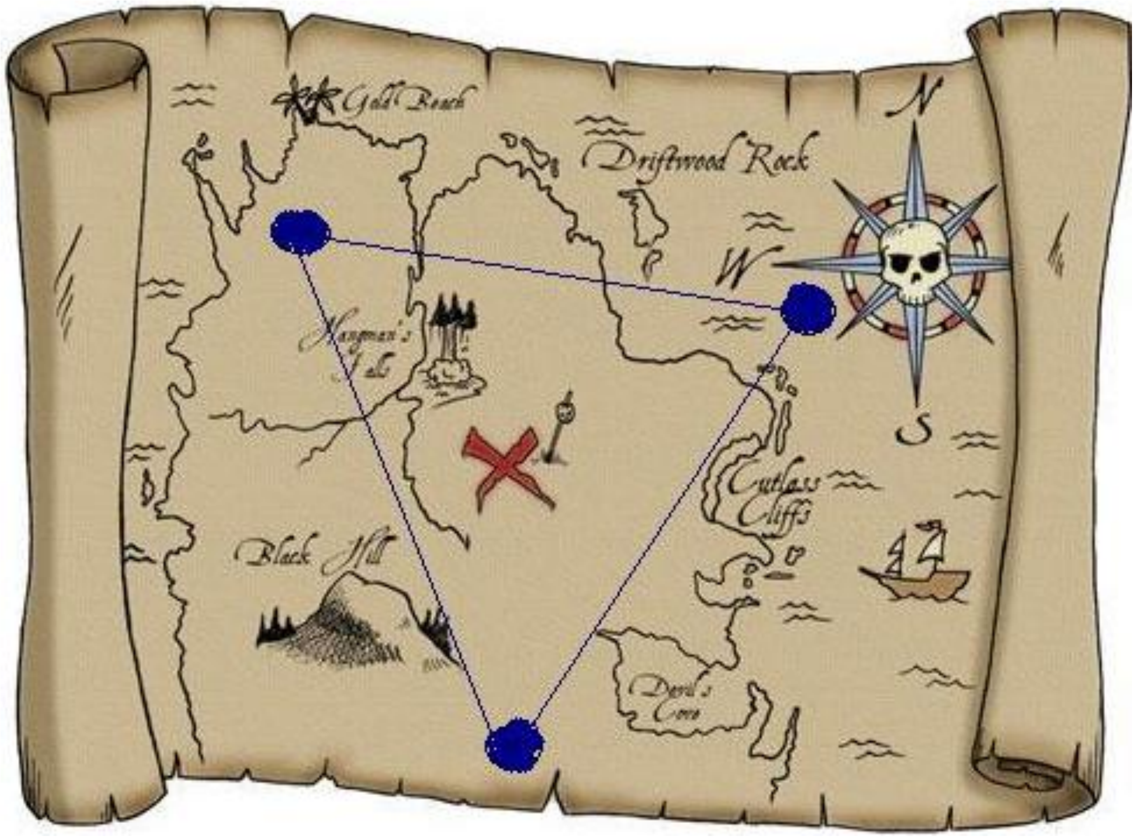
Summary Threshold Cryptography

- Sharing Secrets
- Threshold Encryption
- Threshold Signatures
- E-Voting

- General Problem: Trusted Dealer
- → Secret sharing schemes without trusted dealer


Gold Beach
Driftwood Rock
N
W
S
Hangman's Falls
Black Hill
Cutlass Cliffs
Devil's Cove

?


Gold Beach
Driftwood Rock
N
W
S
Black Hill
Cutlass Cliffs

!

References

[La13] Lagrange polynomial. (2013, May 22). In Wikipedia, The Free Encyclopedia. Retrieved 06:22, June 24, 2013, from http://en.wikipedia.org/w/index.php?title=Lagrange_polynomial&oldid=556301912

[El85] ElGamal, T. (1985, January). A public key cryptosystem and a signature scheme based on discrete logarithms. In Advances in Cryptology (pp. 10-18). Springer Berlin Heidelberg.

[Sho00] V. Shoup, Practical threshold signatures, Advances in Cryptology: EUROCRYPT 2000 (B. Preneel, ed.), Lecture Notes in Computer Science, vol. 1087, Springer, 2000, pp. 207–220.

[Sha79] Shamir, Adi. "How to share a secret." Communications of the ACM 22.11 (1979): 612-613.

[Cr97] Cramer, Ronald, Rosario Gennaro, and Berry Schoenmakers. "A secure and optimally efficient multi-authority election scheme." *European transactions on Telecommunications* 8.5 (1997): 481-490.

[Li04] T-79.159 Cryptography and Data Security, 24.03.2004 Lecture 9: Secret Sharing, Threshold Cryptography, MPC, Helger Lipmaa

[Ca06] Security and Fault-tolerance in Distributed Systems, Winter 2006/07, 7 Distributed Cryptography, Christian Cachin, IBM Zurich Research Lab