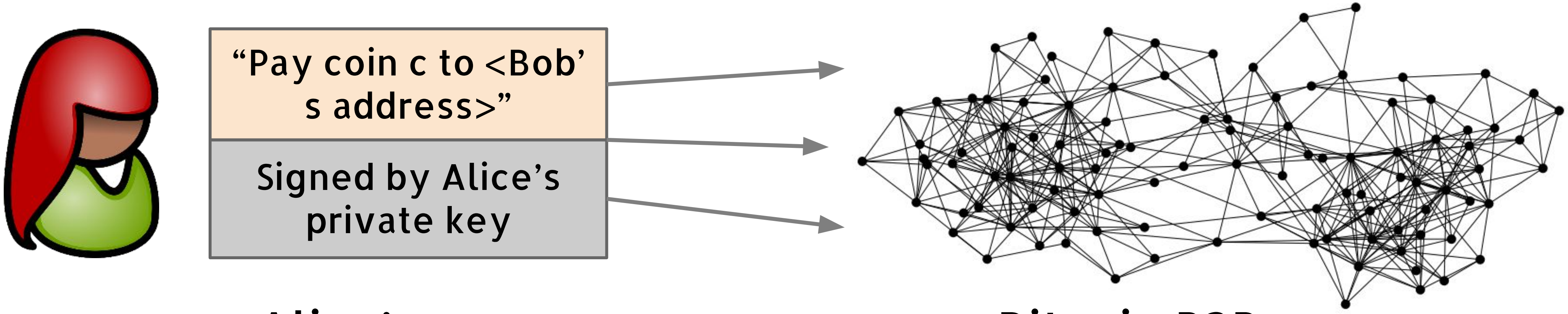
Securing Bitcoin wallets:

A new DSA threshold signature scheme that is usable in the real world

https://eprint.iacr.org/2016/013

Rosario Gennaro, **Steven Goldfeder**, Arvind Narayanan

Spending bitcoins is controlled by crypto



Alice’s device

Bitcoin P2P network

Alice’s device containing her key is a single point of failure

Your bitcoins are as secure as your private keys

Bitcoin hacks, thefts, losses

Rank	Name	Severity (January 2014 ฿)	USD Equivalent
1	2014 Mt. Gox Collapse	850000.000 ฿	700258171 $
2	Silk Road Seizure	32716.283 ฿	26867560 $
3	Sheep Marketplace Incident	4978.276 ฿	4070923 $
4	Silk Road 2 Incident	4400.000 ฿	3624866 $
5	GBL Scam	4185.734 ฿	3437446 $
6	MintPal Incident	3894.492 ฿	3208412 $
7	Bitcoin Savings and Trust	3700.408 ฿	2983473 $
8	PicoStocks Hack	3679.520 ฿	3009397 $
9	MyBitcoin Theft	1395.691 ฿	1072570 $
10	CryptoRush Theft	950.000 ฿	782641 $

Almost half of exchanges hacked

45% of Bitcoin exchanges shut down over three year observation period, many due to hacks

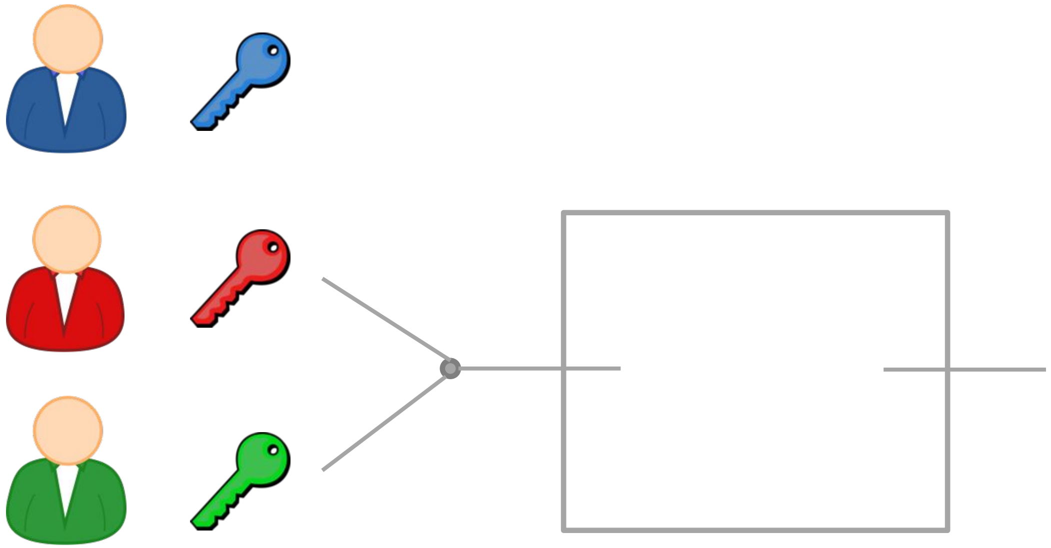
[Moore and Christin, 2013]

Avoiding a single point of failure

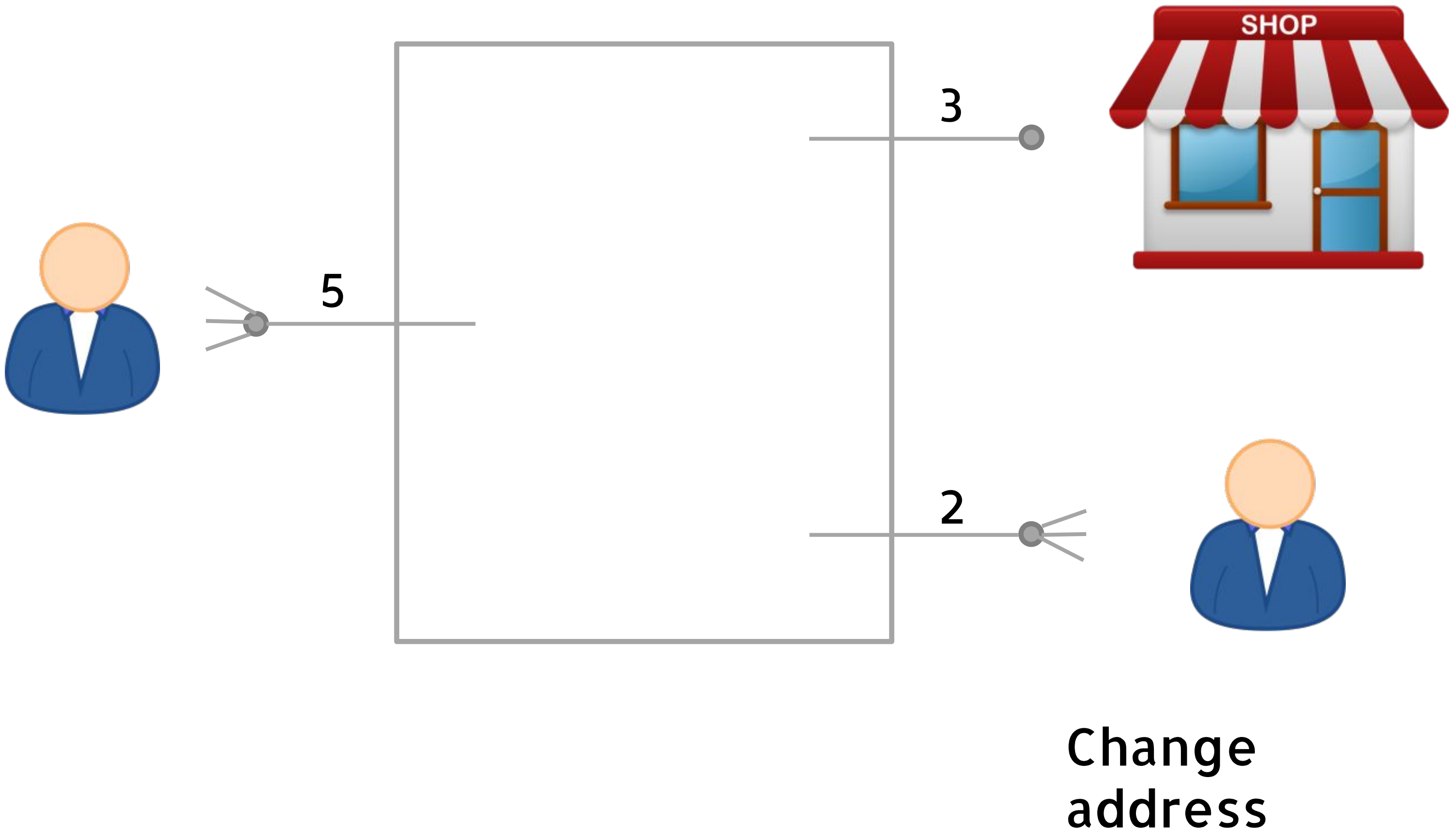
- Split your key between multiple devices
- Designate your address as protected by multiple keys

Bitcoin multisignatures

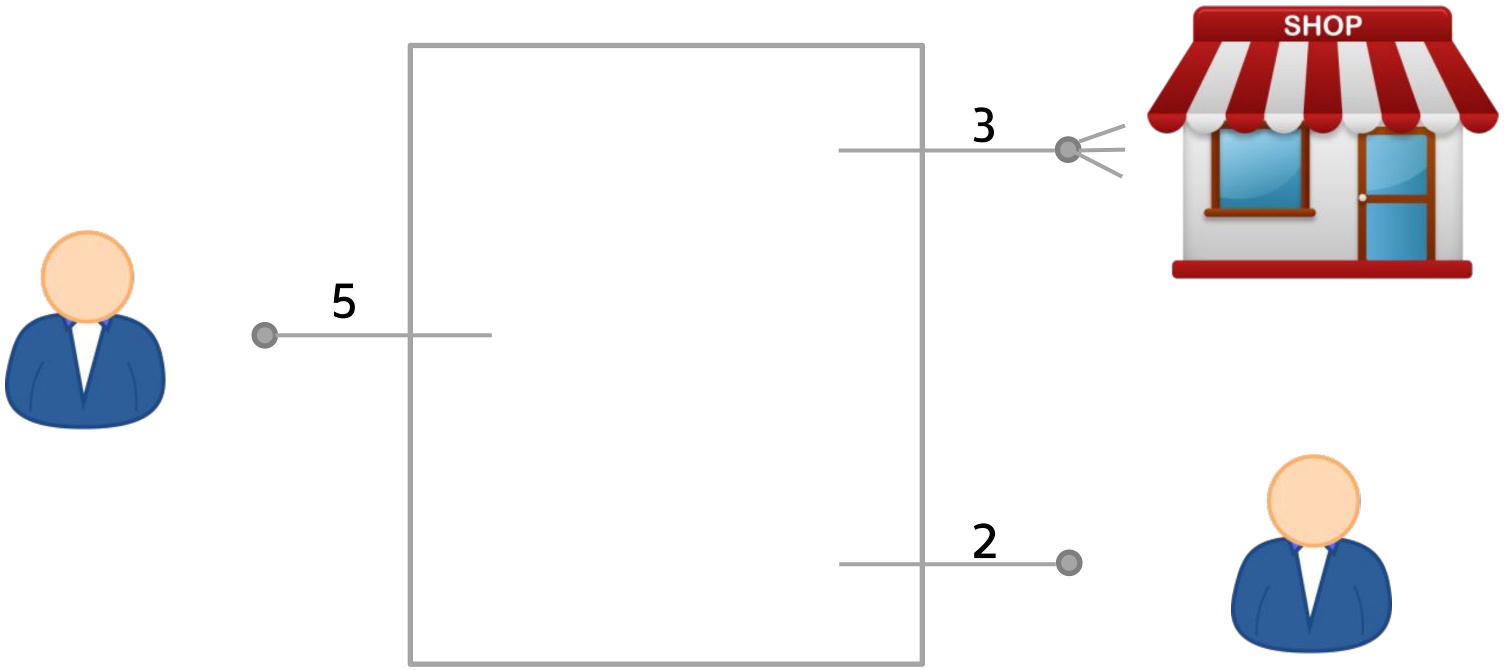
- Associate *n* keys with an address
- Specify a threshold *t* of keys that must sign to spend from that address



Multi-sig ruins anonymity



Multi-sig ruins anonymity



Change address

Multisig at a company

Joint control between 3 employees

Imagine one employee is replaced

Money must be moved on the block chain!

Access structure is public

Multisignatures could reveal too much about security internals to external world

And they're bad for user privacy

Splitting your key: threshold signatures

Informal definition: t-out-of-n threshold signature scheme

The secret key is shared among n players s.t.

- **Correctness**:
  any $t + 1$ of them can jointly sign any given message
- **Security**:
  no t colluding players can forge signature

Advantages of splitting your key

Threshold signature is indistinguishable from a regular signature

Address looks like a standard address

Digital Signature Algorithm (DSA)

Given

- a group G of order N
- a generator g
- a private key $\mathbf{x}$

To sign a message m:

- pick a nonce k s.t. $1 \leq k \leq N - 1$
- $r = g^k$
- $s = k^{-1}(m + \mathbf{x} \cdot r) \bmod N$

Signature is (r,s)

Bitcoin uses an instantiation of DSA on elliptic curves

Nonce k must be kept secret

$r = g^k$
$s = k^{-1}(m + x \cdot r) \bmod N$

Knowledge of k together with signature leaks the key

$$s = k^{-1}(m + x \cdot r) \bmod N$$

$$\rightarrow$$

$$x \equiv_N (s \cdot k - m)\, r^{-1}$$

$$r = g^{k}$$
$$s = k^{-1}(m + x \cdot r) \bmod N$$

GJKR Threshold DSA[1]

Includes multiplication of Shamir shares

1. R. Gennaro, S. Jarecki, H. Krawczyk and T. Rabin. *Threshold DSS Signatures*. EUROCRYPT’96, LNCS Vol.1070, pp. 354–371.

$r = g^{k}$
$s = k^{-1}(m + \mathbf{x} \cdot r) \bmod N$

Problem: Multiplication

If a and b are shared on degree-t polynomials

$a \cdot b$ will be shared on a degree-$2t$ polynomial

→ Need $2t + 1$ players to sign

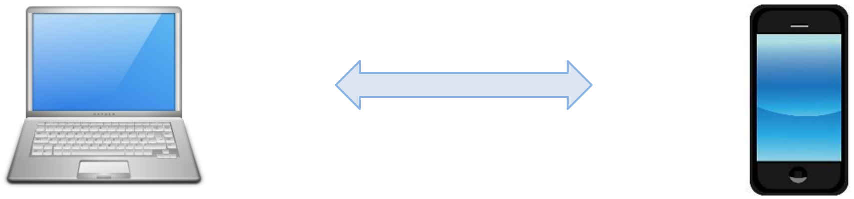
BUT $t + 1$ corrupted players can compromise security!

Not useful for Bitcoin

Need $2t + 1$ players to sign

BUT $t + 1$ corrupted players can compromise security

2-out-of-2 threshold not possible



Mackenzie and Reiter Threshold DSA[2]

Specifically for the two party case (which was unrealizable with GJKR)

We show how this can be extended to t-of-n, but the resulting scheme is very inefficient: key size and computation time grow exponentially with t

2. P. MacKenzie and M. Reiter. Two-party Generation of DSA Signatures. Int. J. Inf. Secur. (2004)

$r = g^{k}$
$s = k^{-1}(m + x \cdot r) \bmod N$

Our scheme

Threshold homomorphic encryption → threshold sig

(Threshold Paillier)

Secret sharing — each player gets:

- a share of a Paillier decryption key
- Enc(x) encrypted under corresponding public key

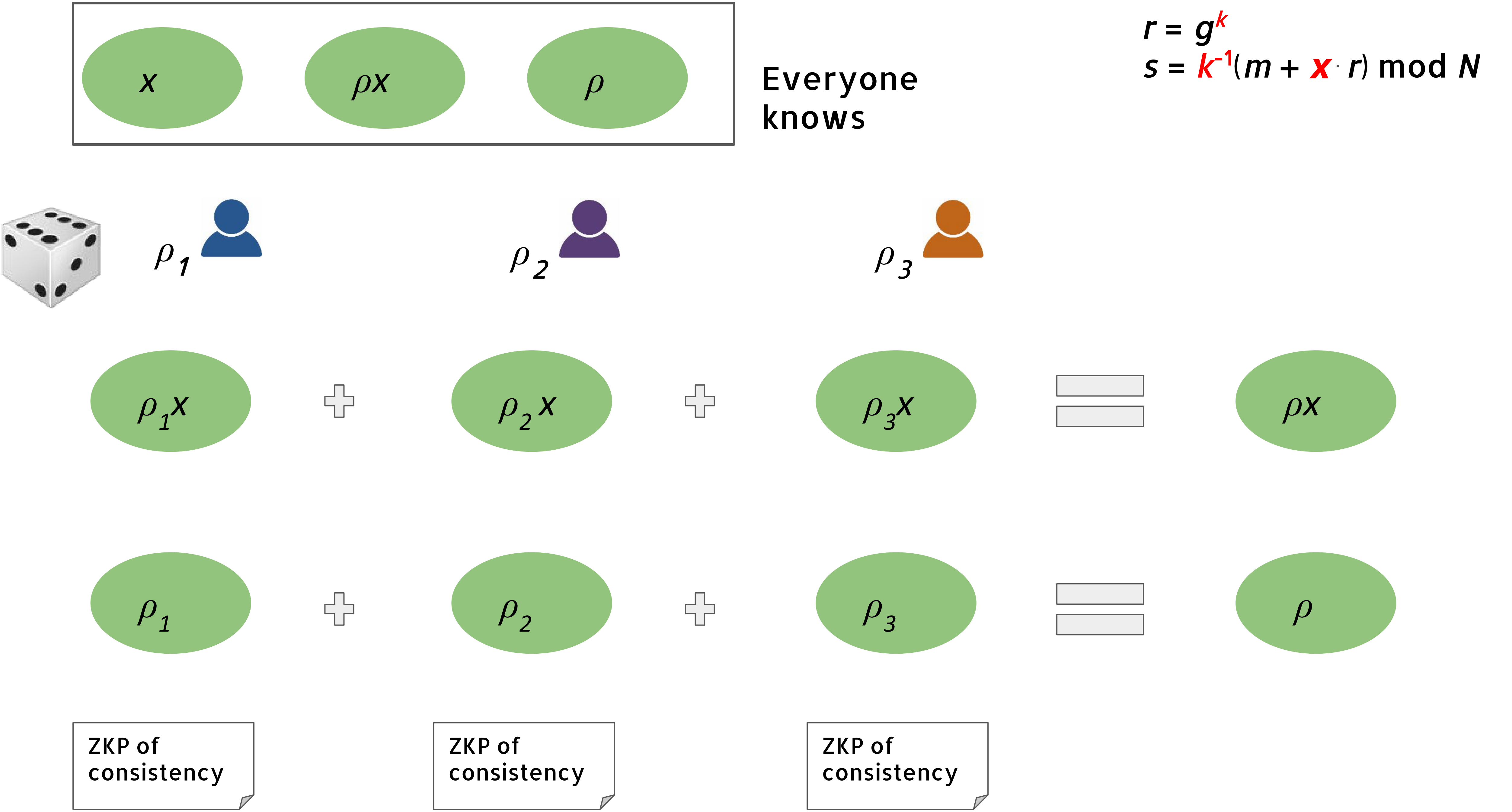
Signature: compute Enc(s), then threshold decrypt

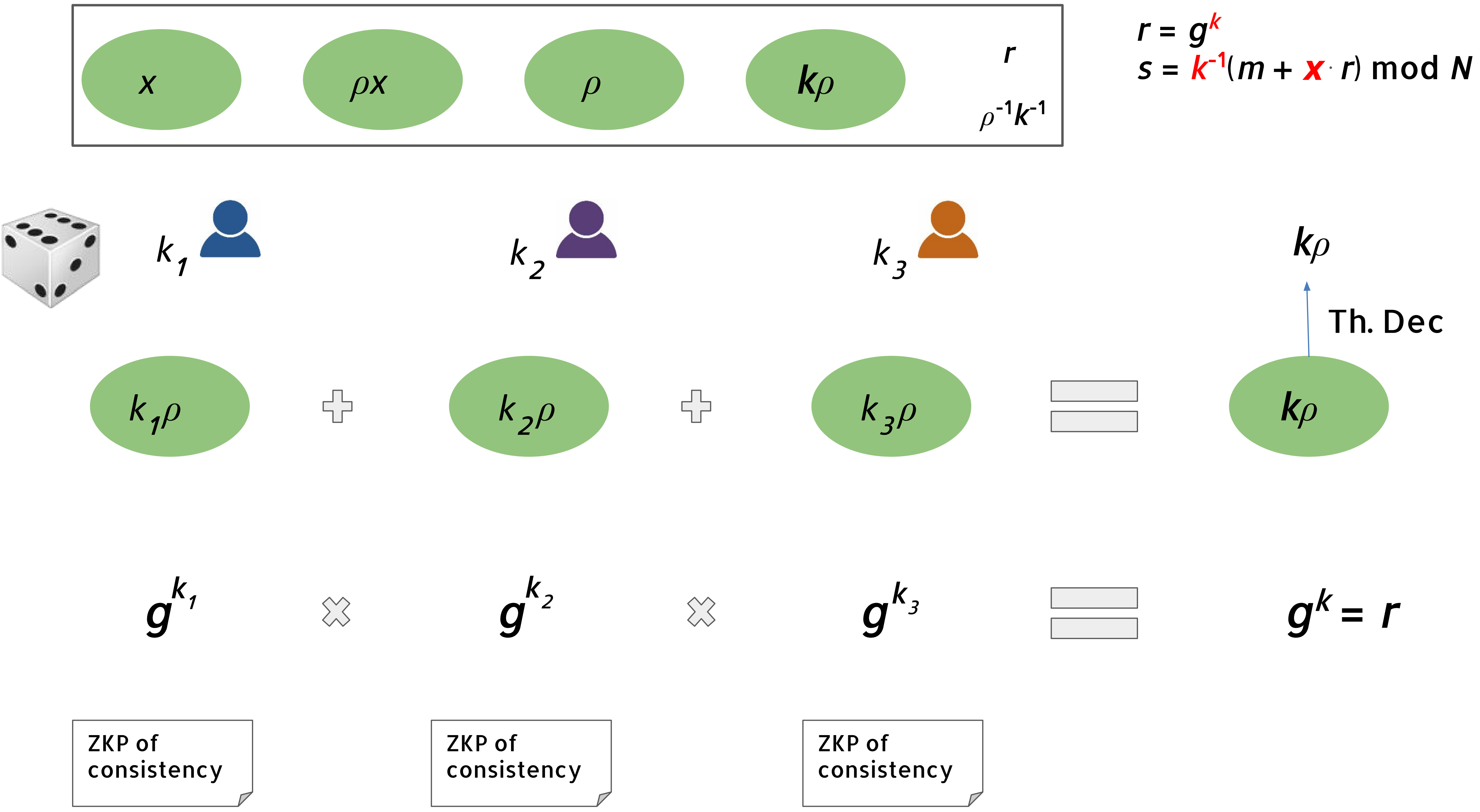
Recall: multiplication by scalar inside additively homomorphic encryption

$r = g^{k}$
$s = k^{-1}(m + \mathbf{x} \cdot r) \bmod N$

Additively homomorphic encryption E

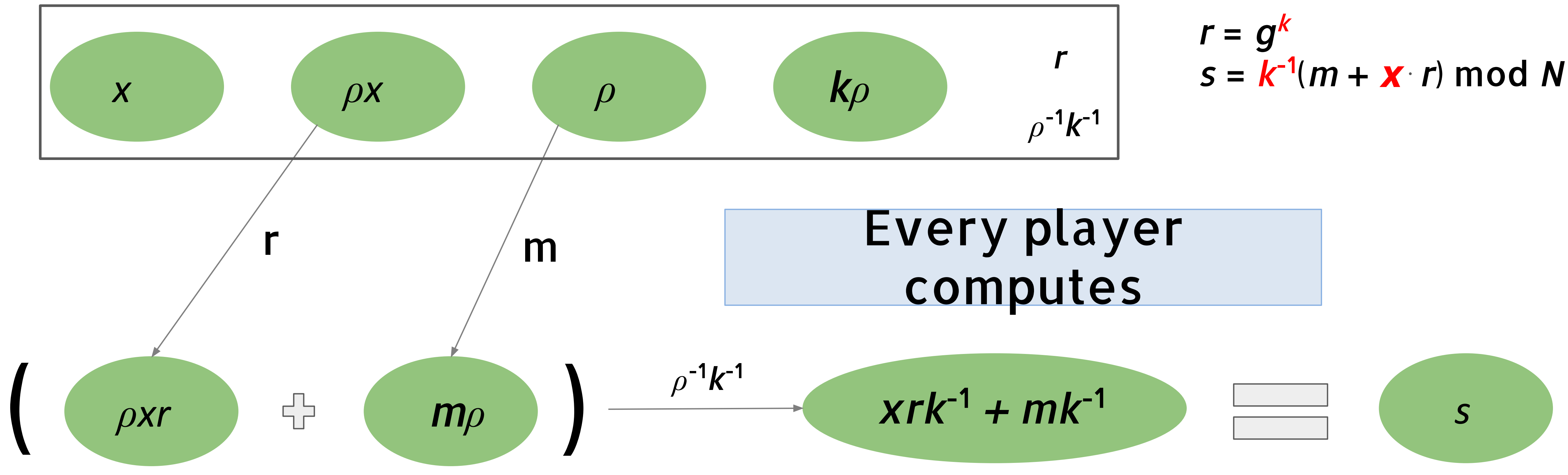
$$E(cx) = E(\underbrace{x + \ldots + x}_{c}) = E(\underbrace{x + \ldots + x}_{c/2}) \Diamond E(\underbrace{x + \ldots + x}_{c/2})$$


$r = g^k$
$s = k^{-1}(m + x \cdot r) \bmod N$
x
ρx
ρ
Everyone knows
ρ_1
ρ_2
ρ_3
$\rho_1 x$ + $\rho_2 x$ + $\rho_3 x$ = ρx
ρ_1 + ρ_2 + ρ_3 = ρ
ZKP of consistency
ZKP of consistency
ZKP of consistency


x
ρx
ρ
kρ
r
ρ^-1 k^-1
r = g^k
s = k^-1(m + x · r) mod N
k1
k2
k3
kρ
Th. Dec
k1ρ
k2ρ
k3ρ
kρ
g^k1
g^k2
g^k3
g^k = r
ZKP of consistency
ZKP of consistency
ZKP of consistency

$r = g^k$

$s = k^{-1}(m + x \cdot r) \bmod N$



Final step: players threshold decrypt s

Why is our scheme different?

GJKR

$t+1$ players can sign. There is no Shamir multiplication, so no need for $2t+1$ participants

MR

works for t-of-n with constant storage, fixed number of rounds and constant per-player computation time

Dealerless protocol

How does each party initially get their share of x?

- Existing key: a trusted dealer who knows x distributes shares
- Fresh key: Each party can independently generate their key share (for both the Paillier and DSA keys)

What if a player is replaced by another?

Still no need to reconstruct the secret!

Future Research: Building Secure protocols

How can we use threshold signatures to build secure protocols?

Building Secure Protocols: Coinbase

Could use threshold signatures for cold storage

Currently use secret-sharing based protocol that requires key reconstruction

Building Secure Protocols: Trezor

Can combine threshold signatures with secure hardware

Building Secure Protocols: OpenBazaar

- Escrow: Buyer and seller jointly choose mediator set
- Buyer ***and*** seller can jointly move money out of escrow
- buyer ***or*** seller together with a majority of mediators can move money out of escrow