SBA Research

COMET

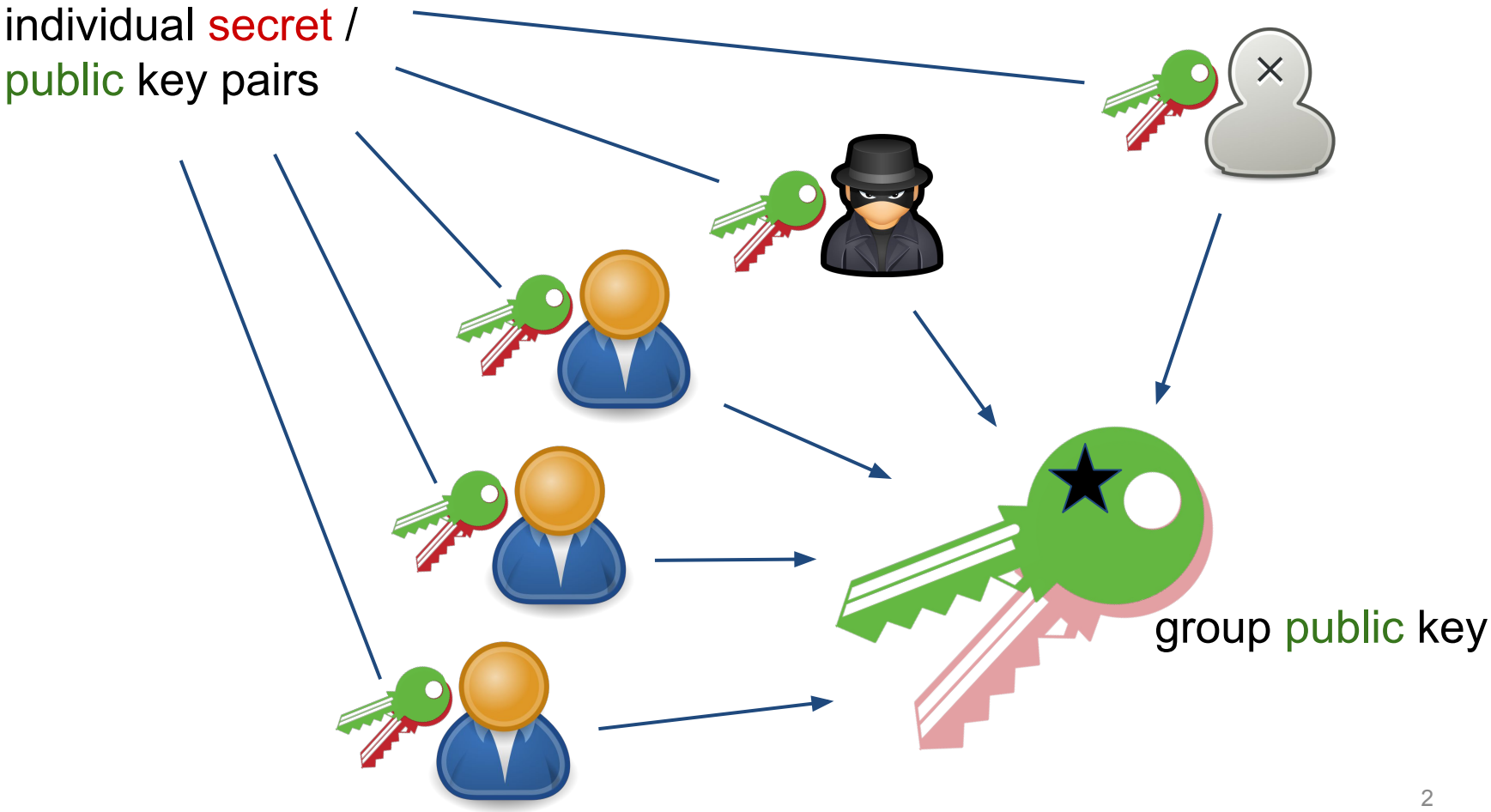
Competence Centers for Excellent Technologies

Distributed Key Generation with Ethereum Smart Contracts

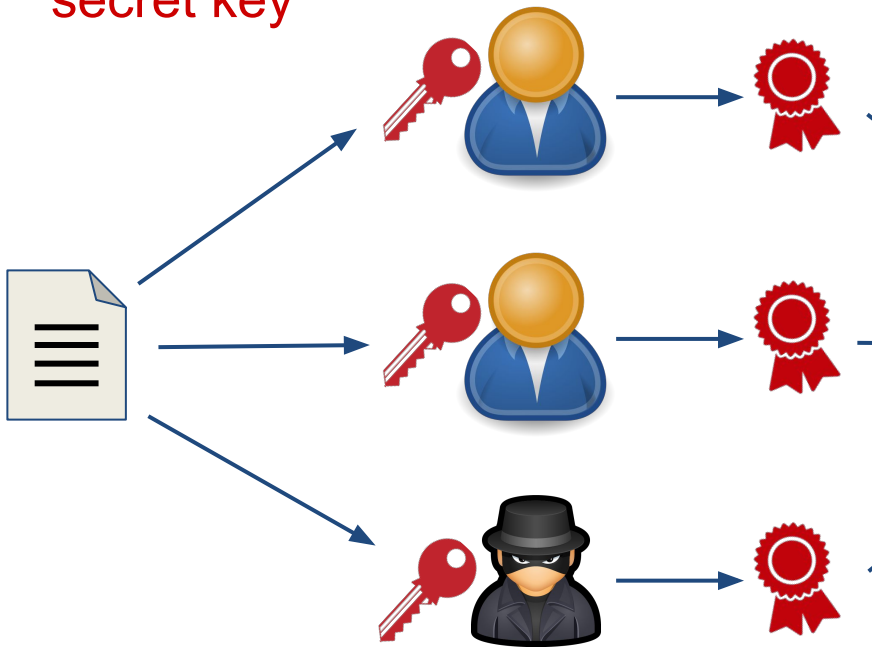
Philipp Schindler, Aljosha Judmayer, Nicholas Stifter

pschindler@sba-research.org

https://github.com/PhilippSchindler/ethdkg


individual secret /
public key pairs
group public key

1. sign message using individual secret key



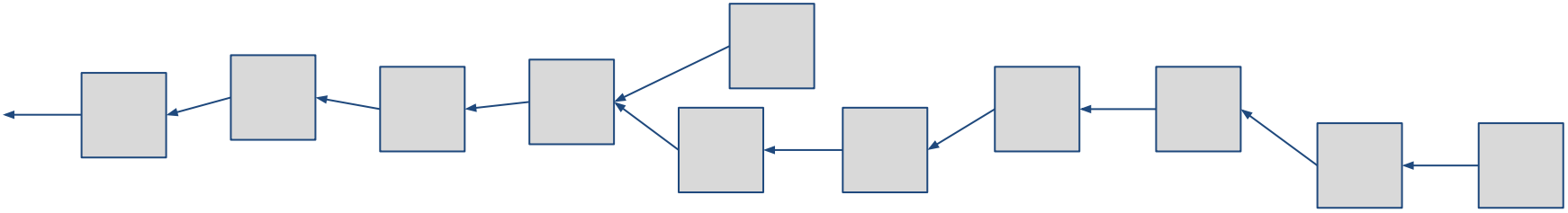
2. aggregate signatures

3. check signature via group public key

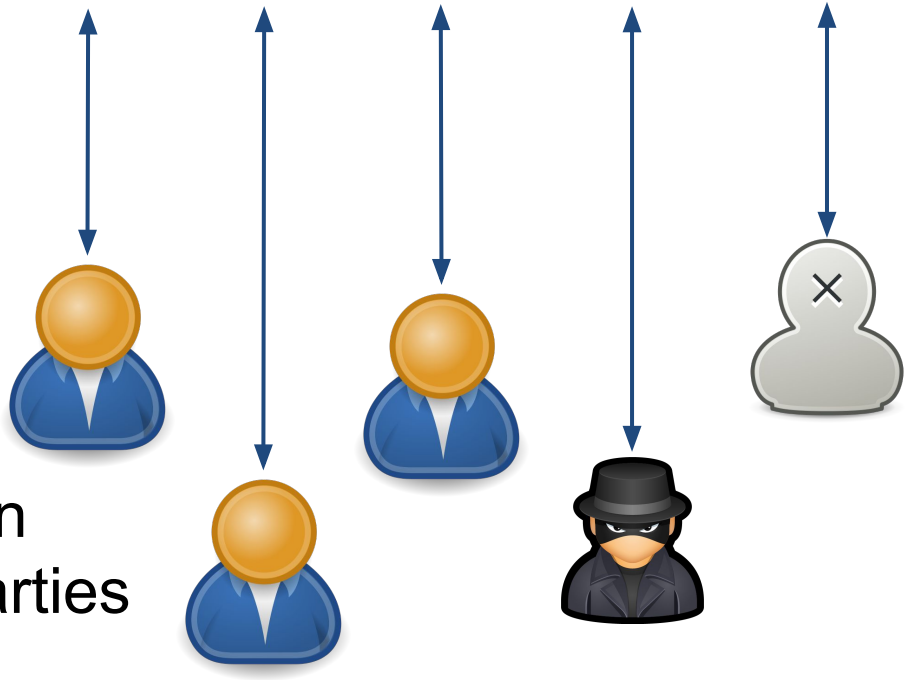
Previous Work

- Gennaro, R., Jarecki, S., Krawczyk, H., Rabin, T.:
  Secure distributed key generation
  for discrete-log based cryptosystems (1999)

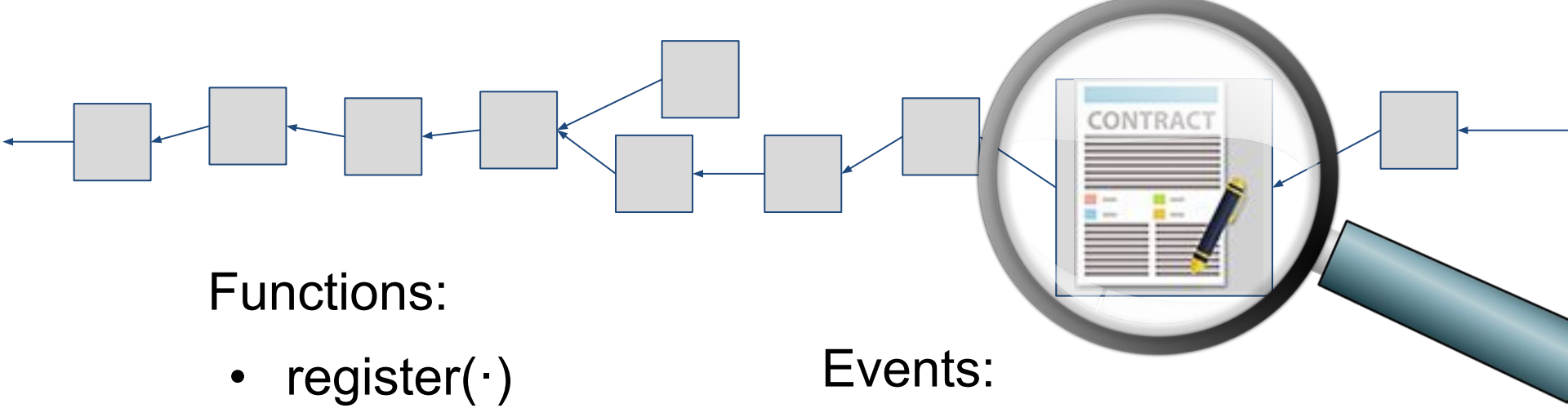
- Kate A., Goldberg I.:
  Distributed key generation for the internet (2009)
  Implementation available:
  https://crysp.uwaterloo.ca/software/DKG/intro.html



Smart Contract on the
Ethereum Blockchain



client application
run by all the parties



Functions:

- register(·)
- share_subkey(·)
- dispute_subkey(·)
- save_groupkey(·)
- verify_signature(·)

Events:

- registration received
- subkey shared
- dispute successful

Client Software

- communication and monitoring of the Smart Contract
- gen. & upload of subkey shares
- verification & submission of disputes
- aggregation & upload of group public key

Client:

- generate BLS keypair
- submit public key

Smart Contract:

- checks eligibility of client to register

Client:

- generate BLS keypair
- submit public key and NIZK proof

knowledge of secret key
challenge includes sender's address

Smart Contract:

- checks eligibility of client to register
- verifies proof

Client:

- run VSS protocol for all registered parties
- submit encrypted shares and verification vectors

Smart Contract:

- "basic" validity checks on the submitted data
- store hash of the submitted data

Client:

- verifies all of its shares received
- submits a dispute for all invalid shares

Smart Contract:

- checks if a claimed dispute is valid
- [withdraw security deposit on success]

verify that **all shares** are **valid**

check that a **single share** is indeed **invalid**

if a party claims that

At least one Client:

- compute and upload the group public key

only clients which successfully completed all previous phases

Smart Contract:

- verify and store the group public key

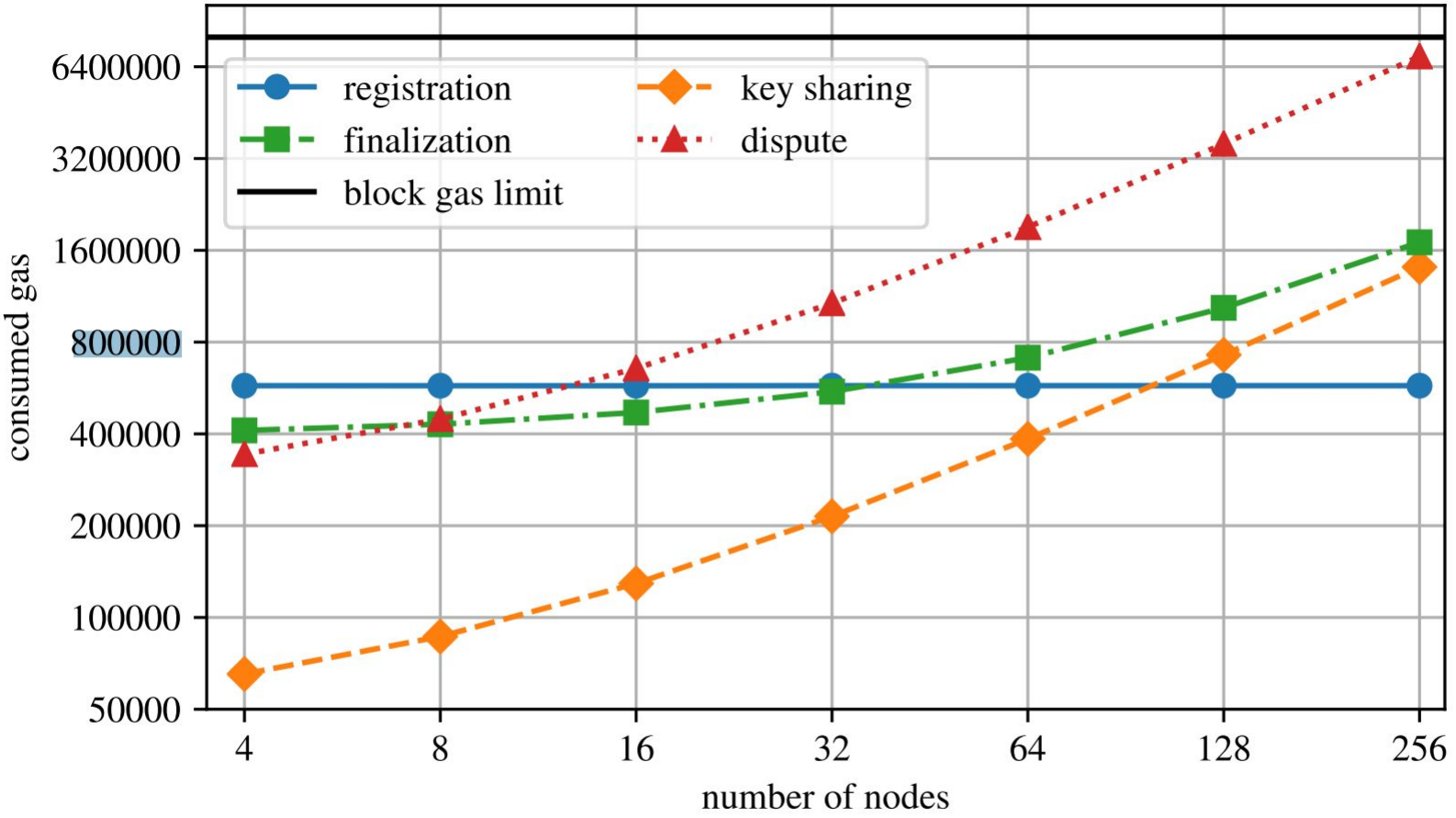


Fig. 1. Measured gas consumption per contract phase and participant

Live Demo

Source Code and additional resources available online:
https://github.com/PhilippSchindler/ethdkg

SBA Research

COMET

Competence Centers for Excellent Technologies

Distributed Key Generation with Ethereum Smart Contracts

Philipp Schindler, Aljosha Judmayer, Nicholas Stifter

pschindler@sba-research.org

https://github.com/PhilippSchindler/ethdkg

Backup Demo

```
D$ # docker build -t ethdkg .
D$ docker run -p 127.0.0.1:8545:8545 -it ethdkg /bin/bash
user@589f269b6e40:/ethdkg$ ganache-cli --host 0.0.0.0 --block
Time 1800
```

```
D$ docker run -p 127.0.0.1:8545:8545 -it ethdkg /bin/bash
user@589f269b6e40:/ethdkg$ ganache-cli --host 0.0.0.0 --block
Time 1800
Ganache CLI v6.3.0 (ganache-core: 2.4.0)

Available Accounts
==================
(0) 0xbb998b91cc2e3ed1e81b2320f7c517cfb75d4574 (~100 ETH)
(1) 0xd91a6fcb86c0c3dd249098e9e9348ec84fa3061e (~100 ETH)
(2) 0x1a6d85d6381da48d5dc1da560584980694c48721 (~100 ETH)
(3) 0x8d2772a7a543b6240187616560 8a2a295f9d5463 (~100 ETH)
(4) 0x02ead56140d9ec2b210836204d9ab99d201e48b5 (~100 ETH)
(5) 0x7758f5e304c519592a5815bfecb62d19821f77ea (~100 ETH)
(6) 0x961bda612dd03dfa647dc02818184a82cc86cd1b (~100 ETH)
(7) 0x23c5a6261d2213112606e74149fd2d47d27e9fd9 (~100 ETH)
(8) 0xef12d02373a1d4cafb54e363617d188eb72daf84 (~100 ETH)
(9) 0xadd97e4cb925c164521473356d7c099907a81987 (~100 ETH)
```

```
HD Wallet
==================
Mnemonic:      december champion present casino clip method l
ight essay van cover erode amount
Base HD Path:  m/44'/60'/0'/0/{account_index}

Gas Price
==================
20000000000

Gas Limit
==================
6721975

Listening on 0.0.0.0:8545
```

```
M$ python dkg.py deploy
deploying DKG contract from Ethereum account 0xaDD97e4cB925C1
645214733356d7c09...
```

```
M$ python dkg.py deploy
deploying DKG contract from Ethereum account 0xaDD97e4cB925C1
645214733356d7c09...

deployment transaction confirmed:
    transaction hash: 0x82fb22d18ac75ab7fe83b2ac5b2f80ecd71dd
4587b321354a779d...
    block number:     1
    gas used:         3492253

contract deployed at: 0x07d9dA9bafbBF8c415Ebf757ac9a09f84D549
7Ba
M$
```

```
1$ python dkg.py --line-length 60 \
> run 1 0x07d9dA9bafbBF8c415Ebf757ac9a09f84D5497Ba
```

```
1$ python dkg.py --line-length 60 \
> run 1 0x07d9dA9bafbBF8c415Ebf757ac9a09f84D5497Ba

RUNNING DKG CLIENT
    NODE ACCOUNT ADDRESS: 0xd91A6FCB86C0C3DD249098E9e9348...
    CONTRACT ADDRESS:     0x07d9dA9bafbBF8c415Ebf757ac9a0...

generating keypair
    sk:     56803439183270452450667550707818684291372659 4...
    pk:     (20007646910153135720701450738772418066599344...
    bls_pk: (20720237403821609377185732448261487207011389...

REGISTRATION PHASE STARTED

sending registration transaction...
```

1

```
sending registration transaction...
registration transaction confirmed
    transaction hash: 0x83059d07a8867629ed5b64e843e2b1395...
    block number:     2
    gas used:         570984

waiting for Registration events until key sharing phase s...

Registration event received
    block number: 2
    assigned id:  1
    address:      0xd91A6FCB86C0C3DD249098E9e9348Ec84Fa3061e
    pk:           [200076469101531357207014507387724180 66...
    bls_pk:       [207202374038216093771857324826148 7207...

waiting for 23 blocks until key sharing phase starts...
```

```
2$ python dkg.py --line-length 60 \
> run 2 0x07d9dA9bafbBF8c415Ebf757ac9a09f84D5497Ba

RUNNING DKG CLIENT
    NODE ACCOUNT ADDRESS: 0x1A6D85d6381dA48d5DC1DA5605849...
    CONTRACT ADDRESS:     0x07d9dA9bafbBF8c415Ebf757ac9a0...

generating keypair
    sk:     58457989218458810284444256454806765015164143 5...
    pk:     (12992611144931392683287343744643391194366988...
    bls_pk: (10695117677974010824572742610685353456719931...

REGISTRATION PHASE STARTED

sending registration transaction...
```

2

```
3$ python dkg.py --line-length 60 \
> run 3 0x07d9dA9bafbBF8c415Ebf757ac9a09f84D5497Ba

RUNNING DKG CLIENT
    NODE ACCOUNT ADDRESS: 0x8D2772A7A543b6240187616560 8A2...
    CONTRACT ADDRESS:     0x07d9dA9bafbBF8c415Ebf757ac9a0...

generating keypair
    sk:     67066110604074748268913247054839760556136 7075...
    pk:     (13637050393881574085094728672185517168804550...
    bls_pk: (17589509483216842594536604223290579702366312...

REGISTRATION PHASE STARTED

sending registration transaction...
```

3

```
    assigned id:  2
    address:      0x1A6D85d6381dA48d5DC1DA560584980694c48721
    pk:           [12992611144931392683287343744643391194...
    bls_pk:       [10695117677974010824572742610685353456...

Registration event received
    block number: 3
    assigned id:  3
    address:      0x8D2772A7A543b6240187616560 8A2a295F9d5463
    pk:           [13637050393881574085094728672185517168...
    bls_pk:       [17589509483216842594536604223290579702...

waiting for 21 blocks until key sharing phase starts...
waiting for 20 blocks until key sharing phase starts...
waiting for 19 blocks until key sharing phase starts...
waiting for 18 blocks until key sharing phase starts...
```

3

```
4$ python dkg.py --line-length 60 \
> run 4 0x07d9dA9bafbBF8c415Ebf757ac9a09f84D5497Ba \
> --send-invalid-shares 1
```

4

```
5$ python dkg.py --line-length 60 \
> run 5 0x07d9dA9bafbBF8c415Ebf757ac9a09f84D5497Ba \
> --abort-after-registration
```

5

```
5$ python dkg.py --line-length 60 \
> run 5 0x07d9dA9bafbBF8c415Ebf757ac9a09f84D5497Ba \
> --abort-after-registration

RUNNING DKG CLIENT
    NODE ACCOUNT ADDRESS: 0x7758f5E304c519592A5815bfECb62...
    CONTRACT ADDRESS:     0x07d9dA9bafbBF8c415Ebf757ac9a0...

generating keypair
    sk:     71410894306899892414952496905558080131372265 7...
    pk:     (156303575375187764172310426636173959583340 70...
    bls_pk: (179906084071703986363231307448847131075121 53...

REGISTRATION PHASE STARTED

sending registration transaction...
```

5

```
    block number: 8
    assigned id:  4
    address:      0x02ead56140d9eC2B210836204d9ab99D201E48B5
    pk:           [339794350155894833000763907862546824 86...
    bls_pk:       [34919766757066468225510797687043770629...

Registration event received
    block number: 8
    assigned id:  5
    address:      0x7758f5E304c519592A5815bfECb62D19821f77ea
    pk:           [156303575375187764172310426361739595 8...
    bls_pk:       [17990608407170398636323130744884713107...

waiting for 16 blocks until key sharing phase starts...
waiting for 14 blocks until key sharing phase starts...
waiting for 13 blocks until key sharing phase starts...
```

5

```
Registration event received
    block number: 8
    assigned id:  5
    address:      0x7758f5E304c519592A5815bfECb62D19821f77ea
    pk:           [156303575375187764172310426636173959580...
    bls_pk:       [179906084071703986363231307448847131070...

waiting for 16 blocks until key sharing phase starts...
waiting for 14 blocks until key sharing phase starts...
waiting for 13 blocks until key sharing phase starts...
waiting for 8 blocks until key sharing phase starts...
waiting for 3 blocks until key sharing phase starts...
waiting for 2 blocks until key sharing phase starts...
waiting for 1 blocks until key sharing phase starts...

ABORTING PROTOCOL
5$
```

5

```
waiting for 3 blocks until key sharing phase starts...
waiting for 2 blocks until key sharing phase starts...
waiting for 1 blocks until key sharing phase starts...

KEY SHARING PHASE STARTED

loading registration data...
    assigned id for this node:            1
    number of register nodes (n):         5
    signing / key recovery threshold (t): 3

generating key shares...
    node 1: 87449974140615897194090710522811838054338131 1...
    node 2: 12880710823070075341987559662233986070503565 3...
    node 3: 18087484145352502112802220900640275224346522 7...
    node 4: 24770745090695948096066490222427761784143208 5...
    node 5: 98259676578999038768936555175560391098036884 3...
```

1

```
generating key shares...
    node 1: 87449974140615897194090710522811838054338131 1...
    node 2: 12880710823070075341987559662233986070503565 3...
    node 3: 18087484145352502112802220900640275224346522 7...
    node 4: 24770745090695948096066490222427761784143208 5...
    node 5: 98259676578999038768936555175560391098036884 3...

encrypting key shares...
    node 1: <no encrypted share for oneself>
    node 2: 87449974140615897194090710522811838054338131 1...
    node 3: 12880710823070075341987559662233986070503565 3...
    node 4: 18087484145352502112802220900640275224346522 7...
    node 5: 24770745090695948096066490222427761784143208 5...

sending key sharing transaction...
```

```
sending key sharing transaction...
key sharing transaction confirmed
    transaction hash: 0xa6cbe2866d39bbfc274055a2669be0c73...
    block number:     26
    gas used:         67918

waiting for KeySharing events until dispute phase starts...

KeySharing event received
    block number:         26
    issuing node id:      3
    encrypted_shares:     [402329867499690582360705580017 4...
    verification vector: [965270570163235371626535595010 6...

KeySharing event received
    block number:         26
    issuing node id:      1
```

1

```
    encrypted_shares:    [701260697827231844577874341208 3...
    verification vector: [120519310534523413844153423456 7...

KeySharing event received
    block number:        26
    issuing node id:     2
    encrypted_shares:    [458696148033909350751893092767 5...
    verification vector: [977216781842871886624345783177 1...

KeySharing event received
    block number:        26
    issuing node id:     4
    encrypted_shares:    [104588908671329356224785834046 2...
    verification vector: [139267272813893904003399086473 0...

waiting for 21 blocks until dispute phase starts...
```

1

```
    block number:         26
    issuing node id:      2
    encrypted_shares:     [458696148033909350751893092767 5...
    verification vector:  [977216781842871886624345783177 1...

KeySharing event received
    block number:         26
    issuing node id:      4
    encrypted_shares:     [104588908671329356224785834046 2...
    verification vector:  [139267272813893904003399086473 0...

waiting for 21 blocks until dispute phase starts...
waiting for 11 blocks until dispute phase starts...
waiting for 6 blocks until dispute phase starts...
waiting for 3 blocks until dispute phase starts...
waiting for 1 blocks until dispute phase starts...
```

```
waiting for 1 blocks until dispute phase starts...

decrypting received shares...

verifying received shares...
INVALID SHARES RECEIVED:
    node 1: this node
        share: 87449974140615897194090710522811838054338 1...
    node 2:
        valid share: 223896351848864080452025206071635653...
        verification vector valid
    node 3:
        valid share: 648815564832075505596087007448737107...
        verification vector valid
    node 4:
        INVALID share: 761318532184983909968992943260675 5...
        verification vector valid
    node 5: <no share received>
```

```
    node 1: this node
        share: 87449974140615897194090710522811838054338 1...
    node 2:
        valid share: 223896351848864080452025206071635653...
        verification vector valid
    node 3:
        valid share: 648815564832075505596087007448737107...
        verification vector valid
    node 4:
        INVALID share: 761318532184983909968992943260675 5...
        verification vector valid
    node 5: <no share received>

DISPUTE PHASE STARTED

submitting dispute for node 4
```

```
verifying received shares...
all received shares are valid:
    node 1:
        valid share: 128807108230700753419875596622339860...
        verification vector valid
    node 2: this node
        share: 873730571627910045170395597293120628369986...
    node 3:
        valid share: 363175372862901941178013434083362733...
        verification vector valid
    node 4:
        valid share: 901786409170900132727406678798802270...
        verification vector valid
    node 5: <no share received>

DISPUTE PHASE STARTED
```

2

```
submitting dispute for node 4
dispute transaction confirmed
    transaction hash: 0x2343b2fb78cf277ab69cd8f58cae7d367...
    block number:     48
    gas used:         345254

no more dispute to file

waiting for DisputeSuccessful events until finalization p...

DisputeSuccessful event received
    block number:           48
    disputed node id:       4
    disputed node address: 0x02ead56140d9eC2B210836204d9a...

waiting for 21 blocks until finalization phase starts...
```

```
    block number:     48
    gas used:         345254

no more dispute to file

waiting for DisputeSuccessful events until finalization p...

DisputeSuccessful event received
    block number:          48
    disputed node id:      4
    disputed node address: 0x02ead56140d9eC2B210836204d9a...

waiting for 21 blocks until finalization phase starts...
waiting for 11 blocks until finalization phase starts...
waiting for 6 blocks until finalization phase starts...
waiting for 1 blocks until finalization phase starts...
```

1

```
FINALIZATION PHASE STARTED

checking which nodes should contribute to the master key
    node 1: YES
    node 2: YES
    node 3: YES
    node 4: NO
    node 5: NO

deriving group and master key...
    BLS group secret key for this node: 17472116580870985...
    BLS group public key for this node: (179132563249514 4...
    BLS master public key:              (370942193450697 9...

sending BLS master public key to contract...
```

```
FINALIZATION PHASE STARTED

checking which nodes should contribute to the master key
    node 1: YES
    node 2: YES
    node 3: YES
    node 4: NO
    node 5: NO

deriving group and master key...
    BLS group secret key for this node: 2198119191221207 7...
    BLS group public key for this node: (468262500281087 9...
    BLS master public key:              (370942193450697 9...

sending BLS master public key to contract...
```

2

```
FINALIZATION PHASE STARTED

checking which nodes should contribute to the master key
    node 1: YES
    node 2: YES
    node 3: YES
    node 4: NO
    node 5: NO

deriving group and master key...
    BLS group secret key for this node: 3175997894603677...
    BLS group public key for this node: (1230866056610669...
    BLS master public key:              (3709421934506979...

sending BLS master public key to contract...
```

3

```
    node 3: YES
    node 4: NO
    node 5: NO

deriving group and master key...
    BLS group secret key for this node: 1747211658087098 5...
    BLS group public key for this node: (179132563249514 4...
    BLS master public key:              (370942193450697 9...

sending BLS master public key to contract...
transaction confirmed
    transaction hash: 0x61703101a3752ce3f70ab41d5921dfd31...
    block number:     74
    gas used:         31907


1$
```

1

```
    node 3: YES
    node 4: NO
    node 5: NO

deriving group and master key...
    BLS group secret key for this node: 2198119191221207 7...
    BLS group public key for this node: (468262500281087 9...
    BLS master public key:              (370942193450697 9...

sending BLS master public key to contract...
transaction confirmed
    transaction hash: 0x7b064ad3440c4a9339f704cfbe492babd...
    block number:     74
    gas used:         413914


2$
```

2

SBA Research

COMET

Competence Centers for Excellent Technologies

Distributed Key Generation with Ethereum Smart Contracts

Philipp Schindler, Aljosha Judmayer, Nicholas Stifter

pschindler@sba-research.org

https://github.com/PhilippSchindler/ethdkg