Threshold signatures from different group actions

Giacomo Borin
2025.04.30 - SQIparty - Lleida SPAIN

Universität Zürich[UZH]
IBM

- Introduction of different *group actions*

- Introduction of different *group actions*
- N-out-of-N case

- Introduction of different *group actions*
- N-out-of-N case
- Active security

- Introduction of different *group actions*
- N-out-of-N case
- Active security
- T-out-of-N case

- Introduction of different *group actions*
- N-out-of-N case
- Active security
- T-out-of-N case
- Few words on open problems and DKG

(Threshold) Signatures

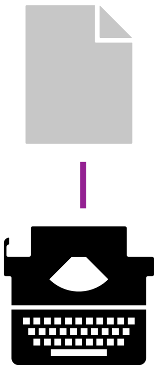
An (T,N)-threshold digital signature scheme is a protocol where any subset of at least T out of N key owners can sign an agreed message, but not one of less than T.

(Threshold) Signatures

An (T,N)-threshold digital signature scheme is a protocol where any subset of at least T out of N key owners can sign an agreed message, but not one of less than T.

(Threshold) Signatures

An (T,N)-threshold digital signature scheme is a protocol where any subset of at least T out of N key owners can sign an agreed message, but not one of less than T.



(Threshold) Signatures

An (T,N)-threshold digital signature scheme is a protocol where any subset of at least T out of N key owners can sign an agreed message, but not one of less than T.

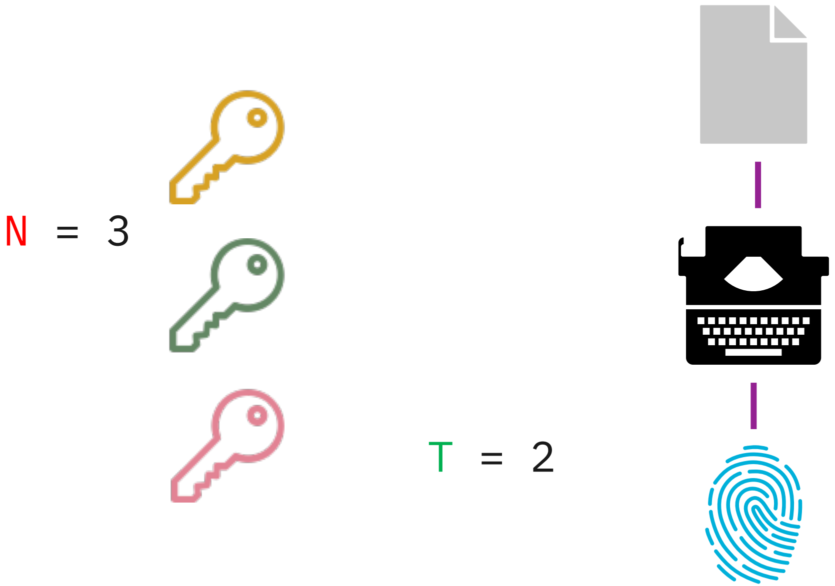
(Threshold) Signatures

An (T,N)-threshold digital signature scheme is a protocol where any subset of at least T out of N key owners can sign an agreed message, but not one of less than T.

N = 3

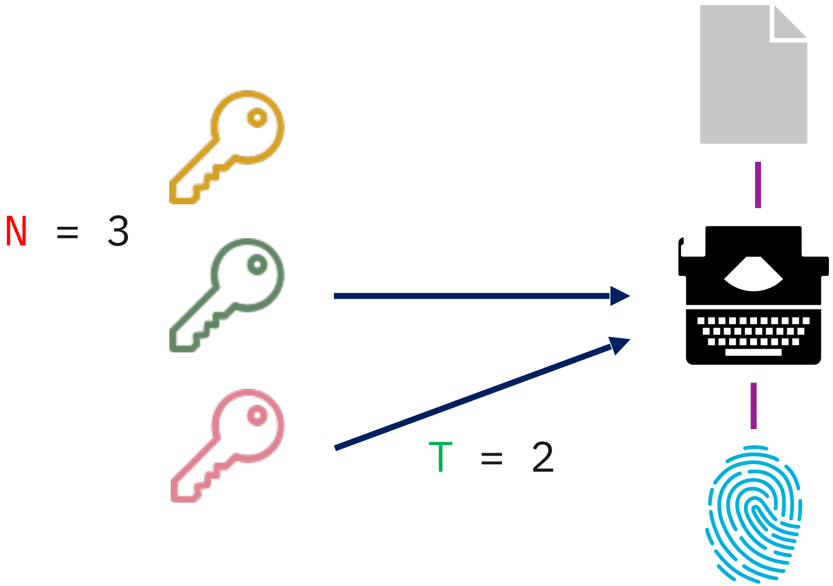
(Threshold) Signatures

An (T,N)-threshold digital signature scheme is a protocol where any subset of at least T out of N key owners can sign an agreed message, but not one of less than T.



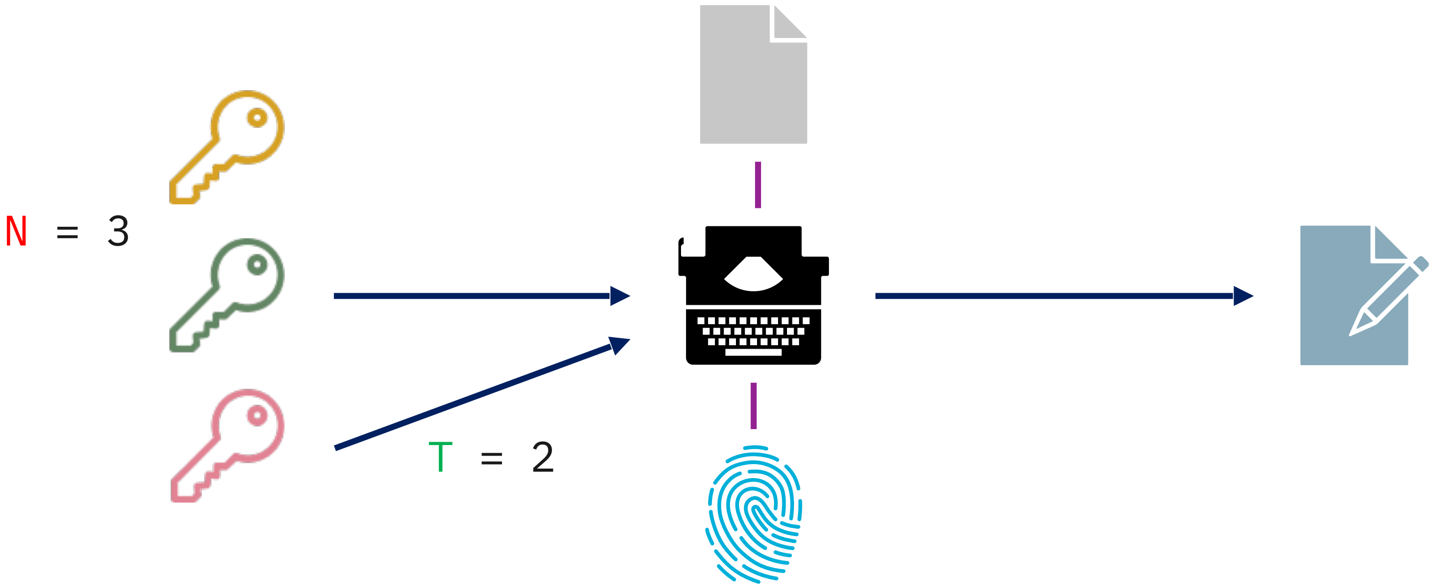
(Threshold) Signatures

An (T,N)-threshold digital signature scheme is a protocol where any subset of at least T out of N key owners can sign an agreed message, but not one of less than T.



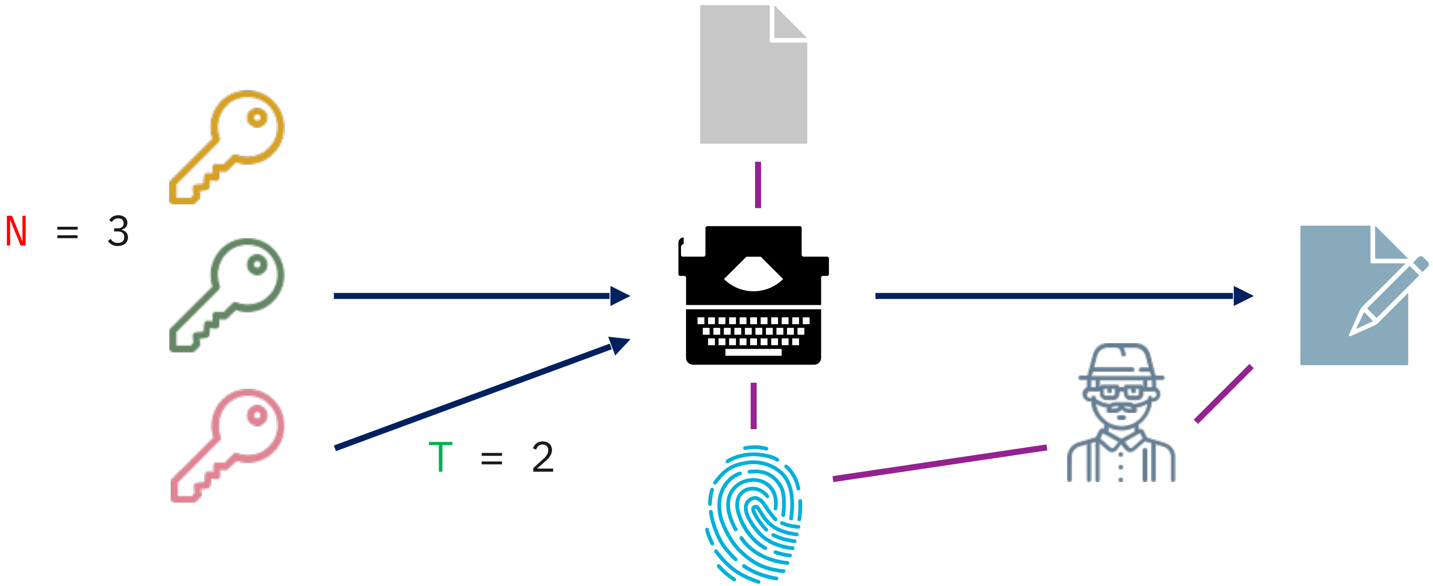
(Threshold) Signatures

An (T,N)-threshold digital signature scheme is a protocol where any subset of at least T out of N key owners can sign an agreed message, but not one of less than T.



(Threshold) Signatures

An (T,N)-threshold digital signature scheme is a protocol where any subset of at least T out of N key owners can sign an agreed message, but not one of less than T.



Cryptographic Group Actions - Definitions

Cryptographic Group Actions - Definitions

$$G \times X \to X$$
$$(g, x) \mapsto g \star x$$

Cryptographic Group Actions - Definitions

$$\boxed{G} \times X \to X$$
$$(g, x) \mapsto g \star x$$

Needs to be a group

Cryptographic Group Actions - Definitions

$$\boxed{G} \times X \to X$$

$$(g, x) \mapsto \boxed{g \star x}$$

Needs to be a group

Composable:
$g_1 \star (g_2 \star x) = (g_1 g_2) \star x$

Cryptographic Group Actions - Definitions

$$\boxed{G} \times X \to X$$

$$(g, x) \mapsto \boxed{g \star x}$$

Needs to be a group

Composable:
$g_1 \star (g_2 \star x) = (g_1 g_2) \star x$

- Effective, i.e. we can efficiently:

Cryptographic Group Actions - Definitions

$$\boxed{G} \times X \to X$$
$$(g, x) \mapsto \boxed{g \star x}$$

Needs to be a group

Composable:
$g_1 \star (g_2 \star x) = (g_1 g_2) \star x$

- Effective, i.e. we can efficiently:
  - compute, sample & canonically represent elements in G

Cryptographic Group Actions - Definitions

$$\boxed{G} \times X \to X$$

$$(g, x) \mapsto \boxed{g \star x}$$

Needs to be a group

Composable:
$g_1 \star (g_2 \star x) = (g_1 g_2) \star x$

- Effective, i.e. we can efficiently:
  - compute, <u>sample & canonically represent elements</u> in G
  - compute the action of <u>all the elements</u> of G

Cryptographic Group Actions - Definitions

$$\boxed{G} \times X \to X$$

$$(g, x) \mapsto \boxed{g \star x}$$

Needs to be a group

Composable:
$g_1 \star (g_2 \star x) = (g_1 g_2) \star x$

- Effective, i.e. we can efficiently:
  - compute, <u>sample & canonically represent elements</u> in G
  - compute the action of <u>all the elements</u> of G
- Cryptographic:

Cryptographic Group Actions - Definitions

$$\boxed{G} \times X \to X$$

$$(g, x) \mapsto \boxed{g \star x}$$

Needs to be a group

Composable:
$g_1 \star (g_2 \star x) = (g_1 g_2) \star x$

- Effective, i.e. we can efficiently:
  - compute, <u>sample & canonically represent elements</u> in G
  - compute the action of <u>all the elements</u> of G
- Cryptographic:
  - Vectorization: given x, y it is hard to find g s.t. $g \star x = y$

Cryptographic Group Actions - Definitions

$$\boxed{G} \times X \to X$$

$$(g, x) \mapsto \boxed{g \star x}$$

Needs to be a group

Composable:
$g_1 \star (g_2 \star x) = (g_1 g_2) \star x$

- Effective, i.e. we can efficiently:
  - compute, <u>sample & canonically represent elements</u> in G
  - compute the action of <u>all the elements</u> of G
- Cryptographic:
  - Vectorization: given x, y it is hard to find g s.t. $g \star x = y$
  - Parallelisation: given $x, y = g \star x, z = h \star x$ and w it is hard to say if $w = (gh) \star x$

Cryptographic Group Actions - Instantiations

$$G \times X \to X$$
$$(g, x) \mapsto g \star x$$

Cryptographic Group Actions - Instantiations

$$\boxed{G} \times X \to X$$
$$(g, x) \mapsto g \star x$$

Which kind of?

Cryptographic Group Actions - Instantiations

$$\boxed{G} \times X \to X$$
$$(g, x) \mapsto g \star x$$

Which kind of?

- Isomorphism problems from Tensors/Coding Theory (1)+ > **Non-Abelian**

(1) Barenghi A, Biasse JF, Persichetti E, Santini P. LESS-FM: fine-tuning signatures from the code equivalence problem.

Cryptographic Group Actions - Instantiations

Which kind of?

$$\boxed{G} \times X \to X$$
$$(g, x) \mapsto g \star x$$

- Isomorphism problems from Tensors/Coding Theory (1)+ > **Non-Abelian**
- Class Group acting on Oriented Supersingular Elliptic Curves:

(1) Barenghi A, Biasse JF, Persichetti E, Santini P. LESS-FM: fine-tuning signatures from the code equivalence problem.

Cryptographic Group Actions - Instantiations

$$\boxed{G} \times X \to X$$
$$(g, x) \mapsto g \star x$$

Which kind of?

- Isomorphism problems from Tensors/Coding Theory (1)+ > **Non-Abelian**
- Class Group acting on Oriented Supersingular Elliptic Curves:
  - CSI-FiSh (2) > **Cyclic** > We can work with $\mathbb{Z}/\#G\mathbb{Z}$

(1) Barenghi A, Biasse JF, Persichetti E, Santini P. LESS-FM: fine-tuning signatures from the code equivalence problem.
(2) Beullens W, Kleinjung T, Vercauteren F. CSI-FiSh: efficient isogeny based signatures through class group computations.

Cryptographic Group Actions - Instantiations

Which kind of?

$$\boxed{G} \times X \to X$$
$$(g, x) \mapsto g \star x$$

- Isomorphism problems from Tensors/Coding Theory (1)+ > **Non-Abelian**
- Class Group acting on Oriented Supersingular Elliptic Curves:
  - CSI-FiSh (2) > **Cyclic** > We can work with $\mathbb{Z}/\#G\mathbb{Z}$
  - PEGASIS (3) > **Abelian**

(1) Barenghi A, Biasse JF, Persichetti E, Santini P. LESS-FM: fine-tuning signatures from the code equivalence problem.
(2) Beullens W, Kleinjung T, Vercauteren F. CSI-FiSh: efficient isogeny based signatures through class group computations.
(3) Dartois P, Eriksen JK, Fouotsa TB, Le Merdy AH, Invernizzi R, Robert D, Rueger R, Vercauteren F, Wesolowski B. PEGASIS: Practical Effective Class Group Action using 4-Dimensional Isogenies.

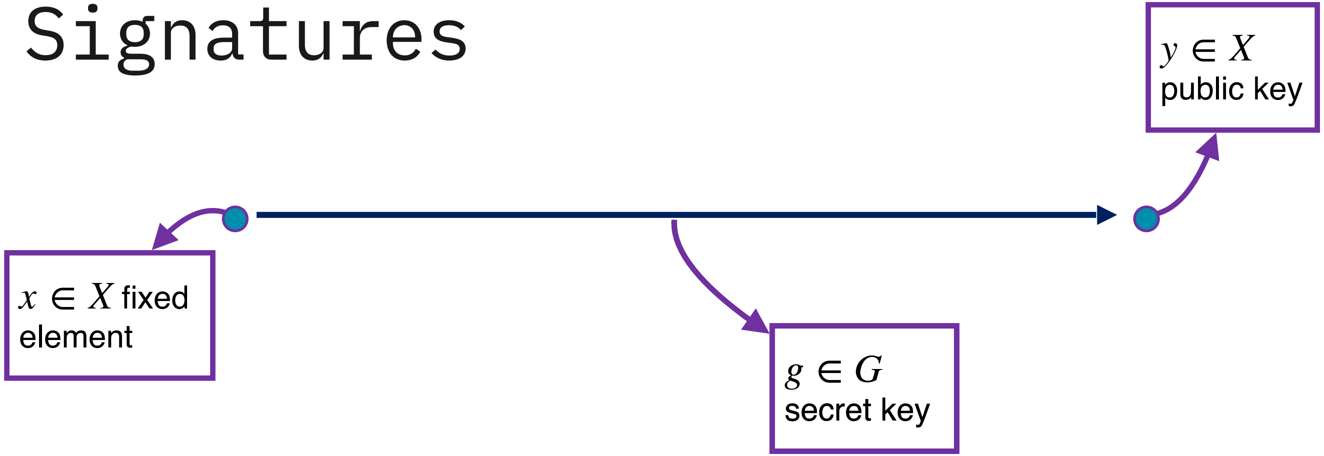
Signatures and Threshold Signatures

Signatures and Threshold Signatures

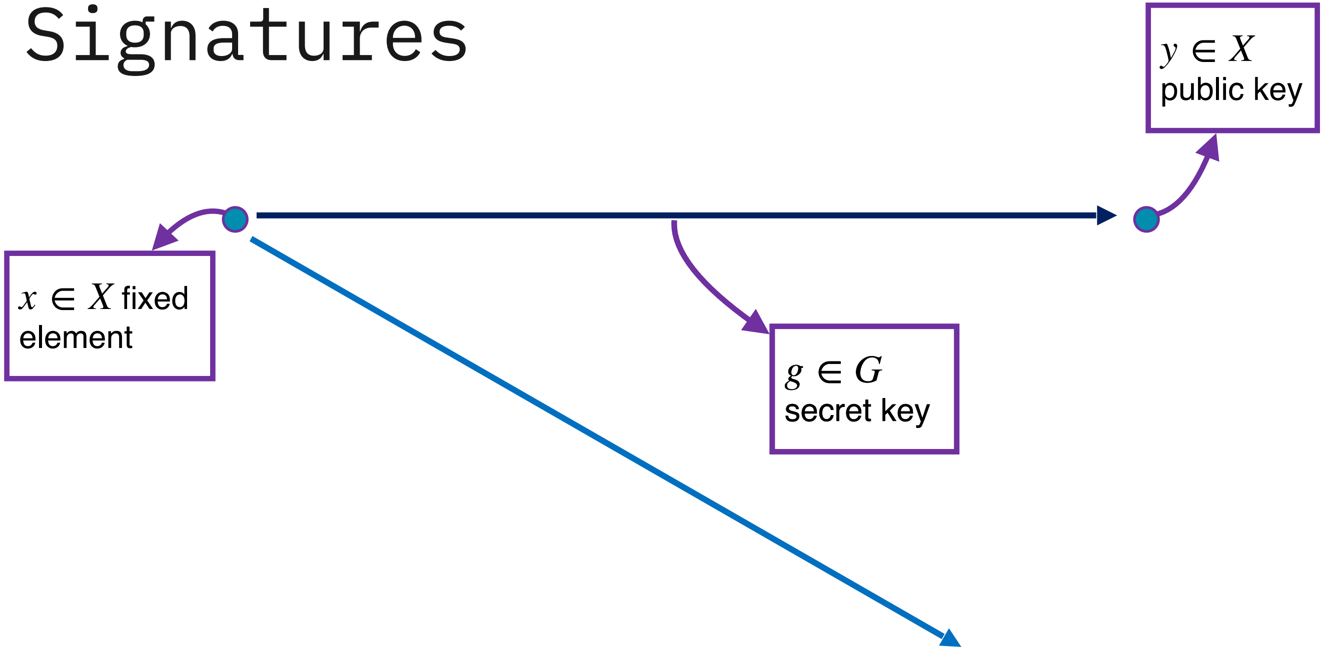
$x \in X$ fixed element

Signatures and Threshold Signatures

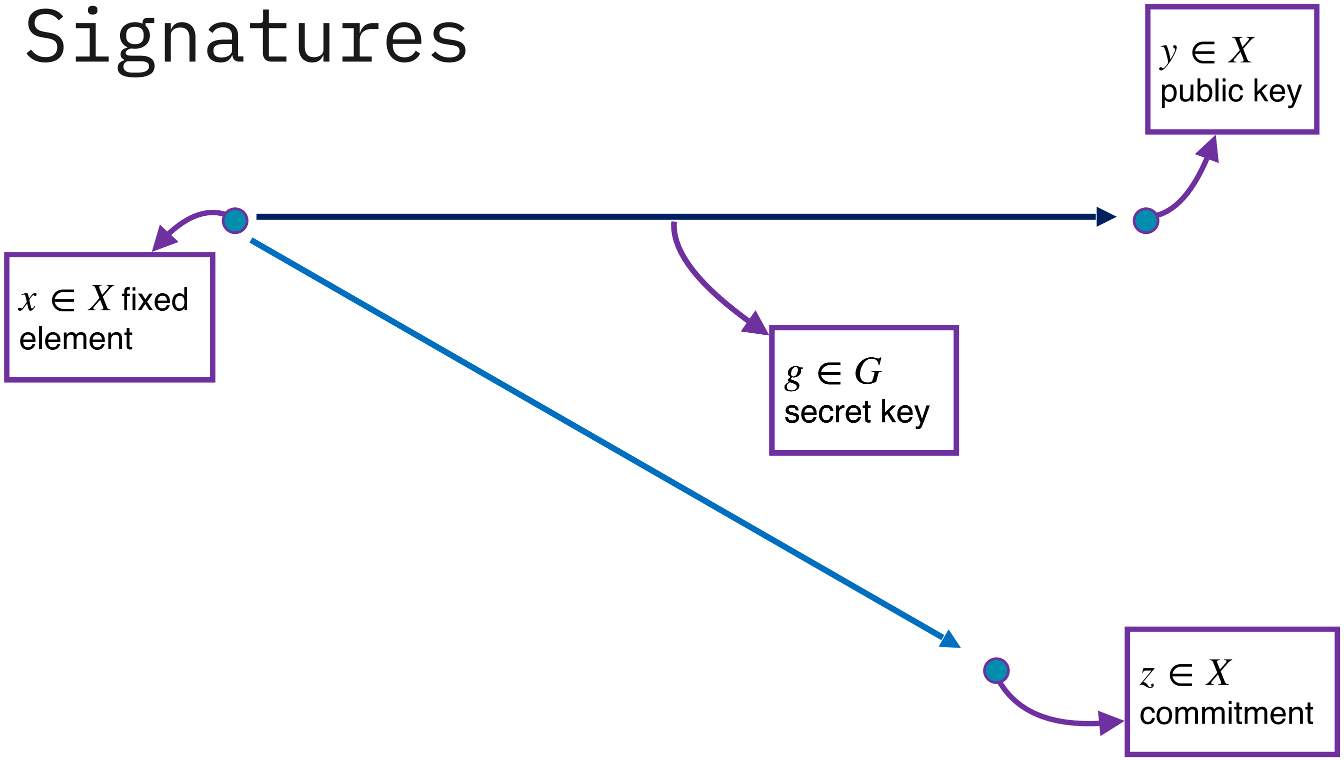
Signatures and Threshold Signatures



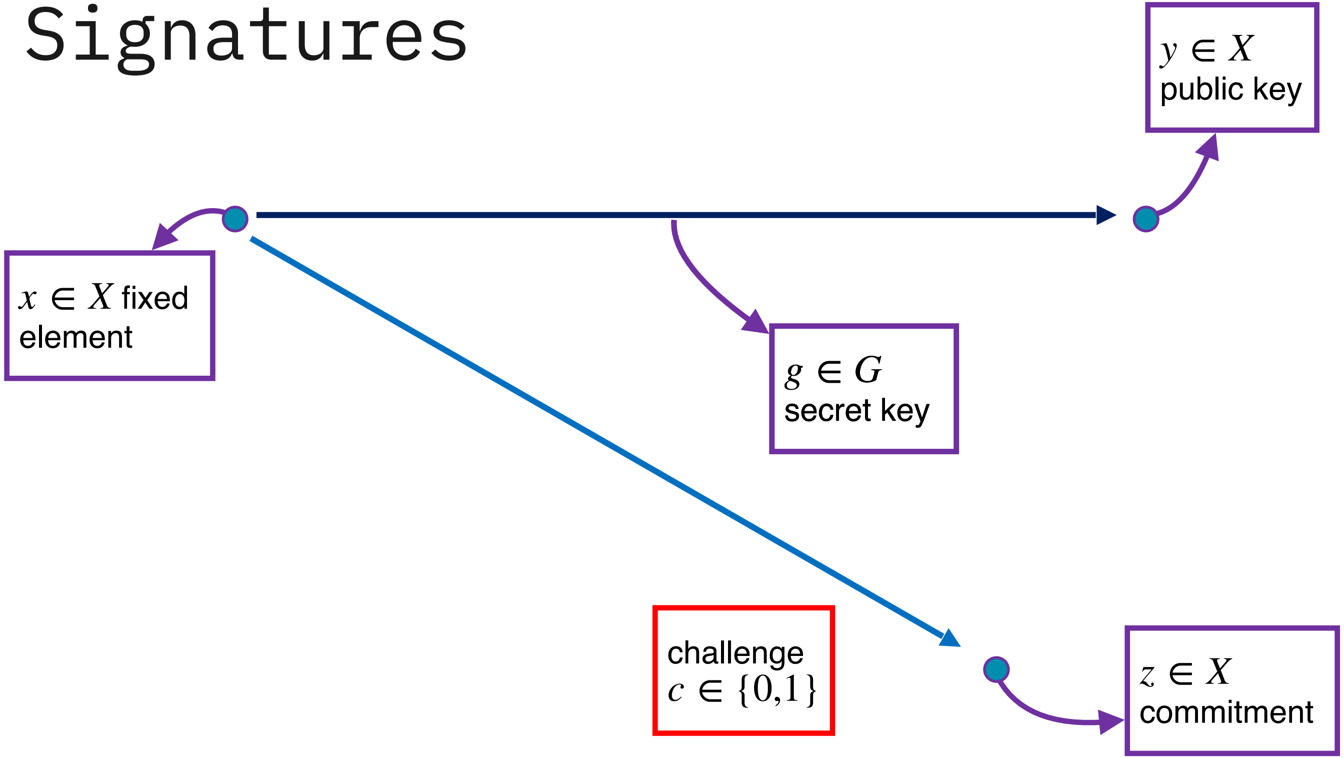
Signatures and Threshold Signatures



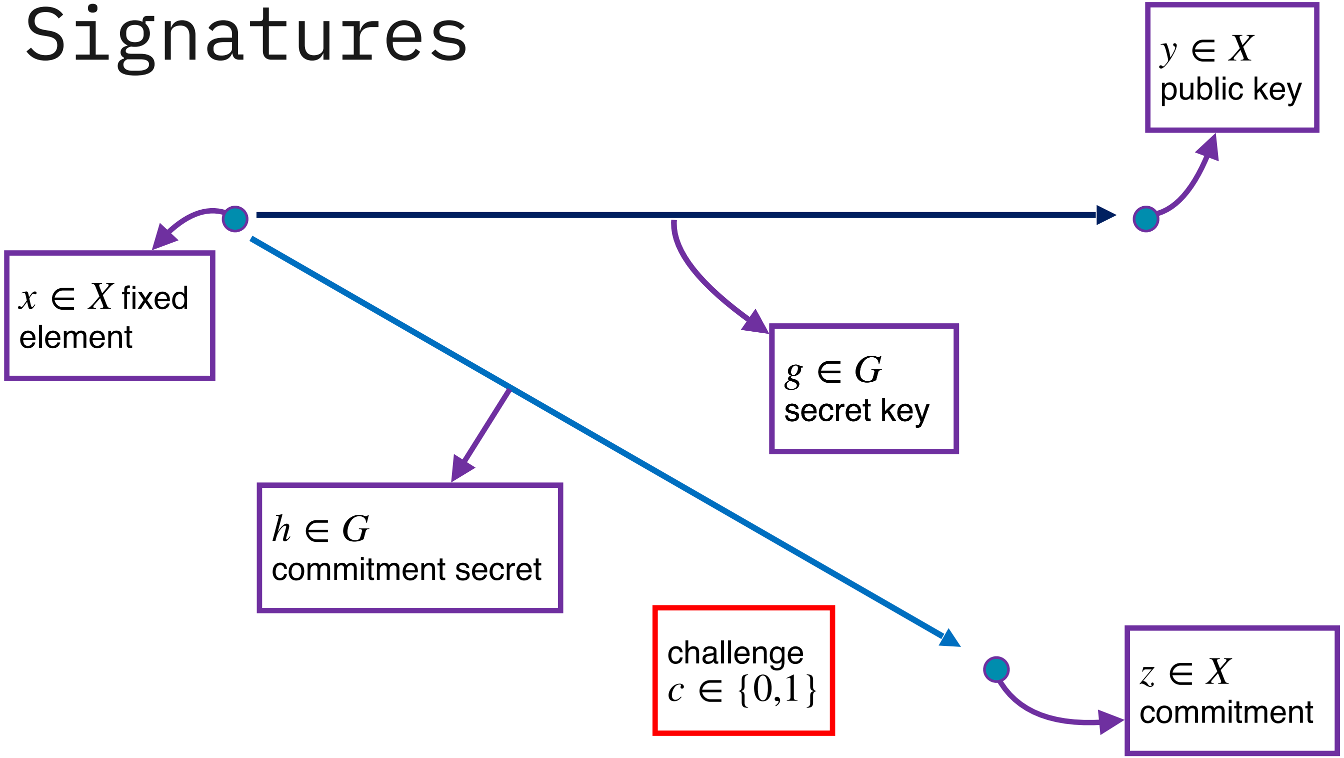
Signatures and Threshold Signatures



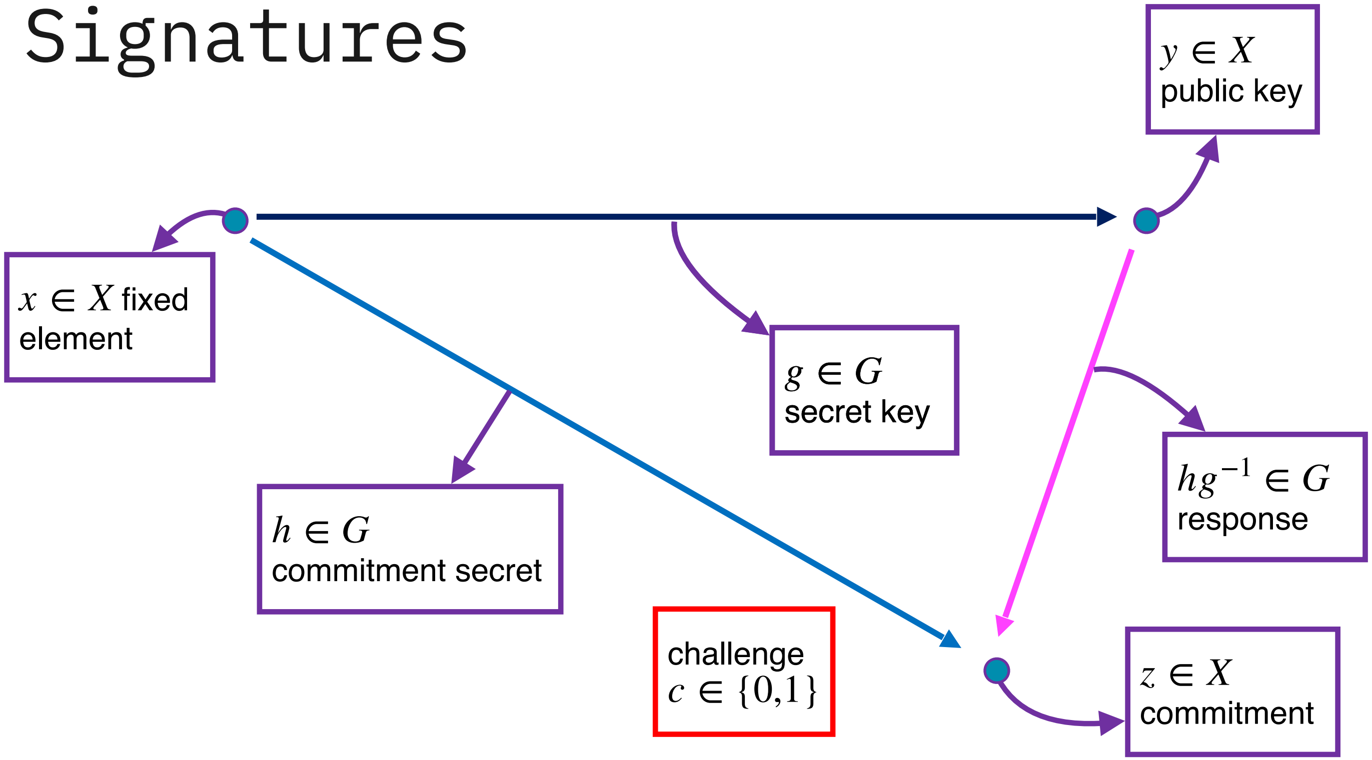
Signatures and Threshold Signatures



Signatures and Threshold Signatures

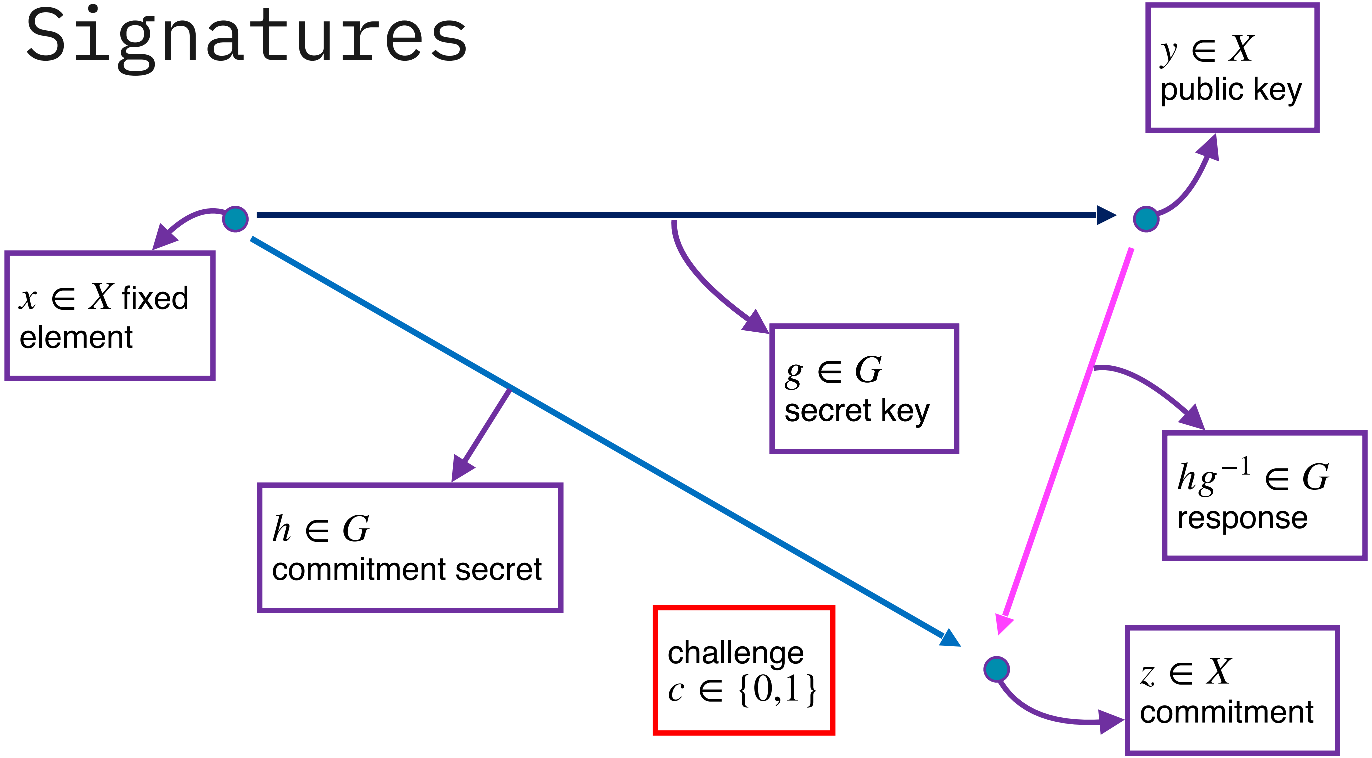


Signatures and Threshold Signatures

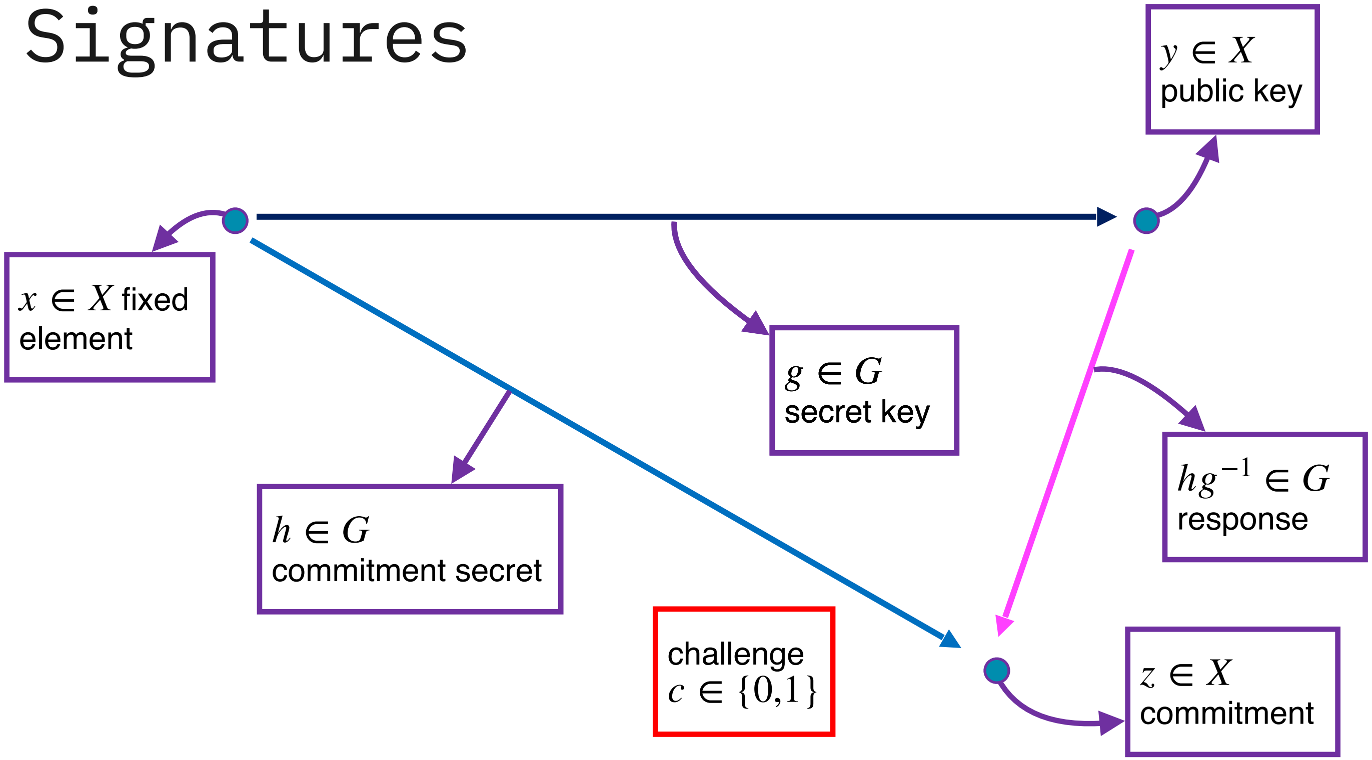


Signatures and Threshold Signatures

-Repeat λ times;



Signatures and Threshold Signatures

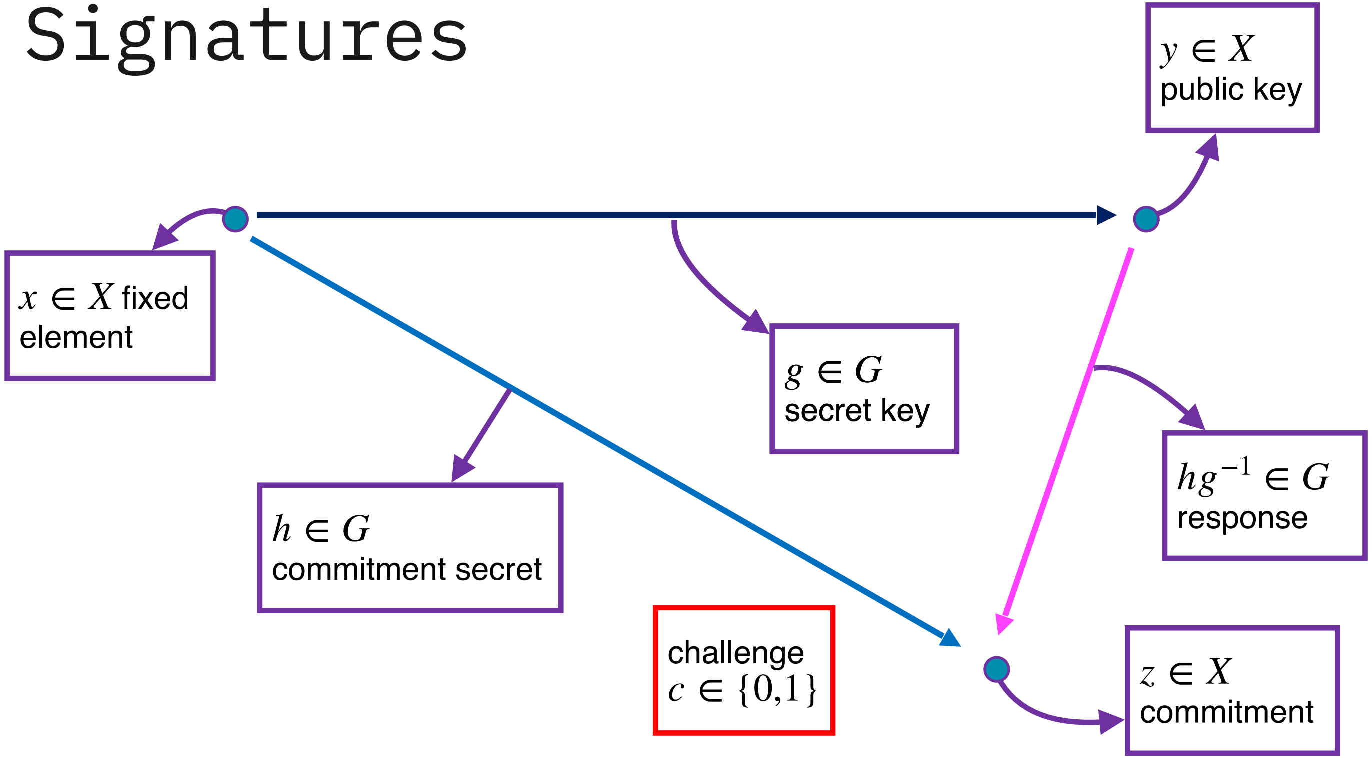


(1) De Feo L, Galbraith SD. SeaSign: compact isogeny signatures from class group actions

-Repeat λ times;

-With Fiat-Shamir transform you get a signature;

Signatures and Threshold Signatures



-Repeat λ times;

-With Fiat-Shamir transform you get a signature;

-Boneh et.al. (2): you need to do that at least λ group actions.

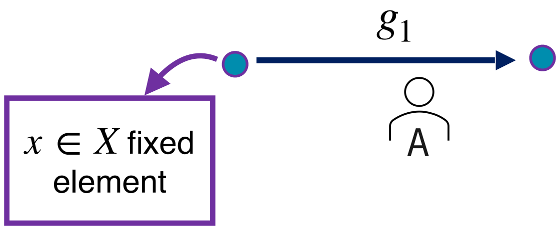
(1) De Feo L, Galbraith SD. SeaSign: compact isogeny signatures from class group actions
(2) Boneh D, Guan J, Zhandry M. A lower bound on the length of signatures based on group actions and generic isogenies.

Core intuition: N-out-of-N case

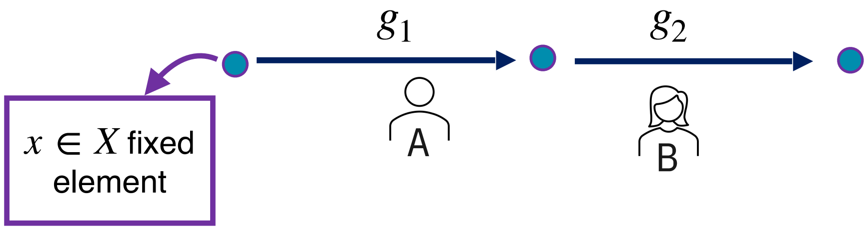
Core intuition: N-out-of-N case

$x \in X$ fixed element

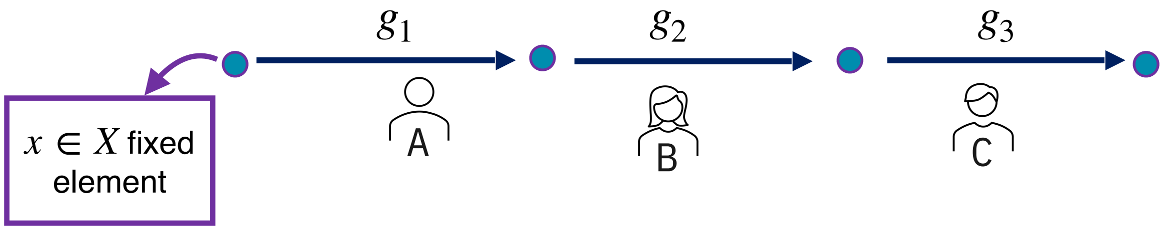
Core intuition: N-out-of-N case



Core intuition: N-out-of-N case

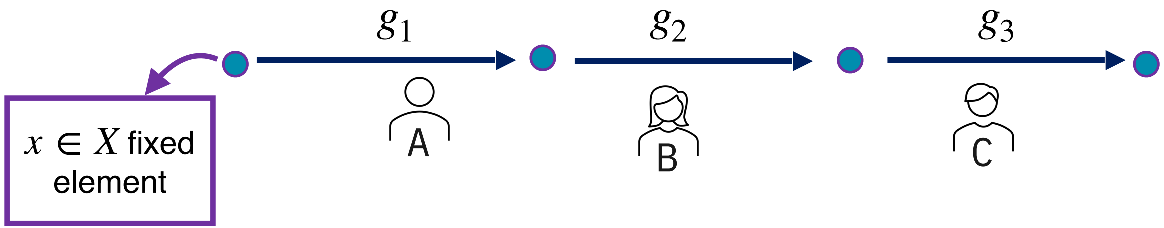


Core intuition: N-out-of-N case

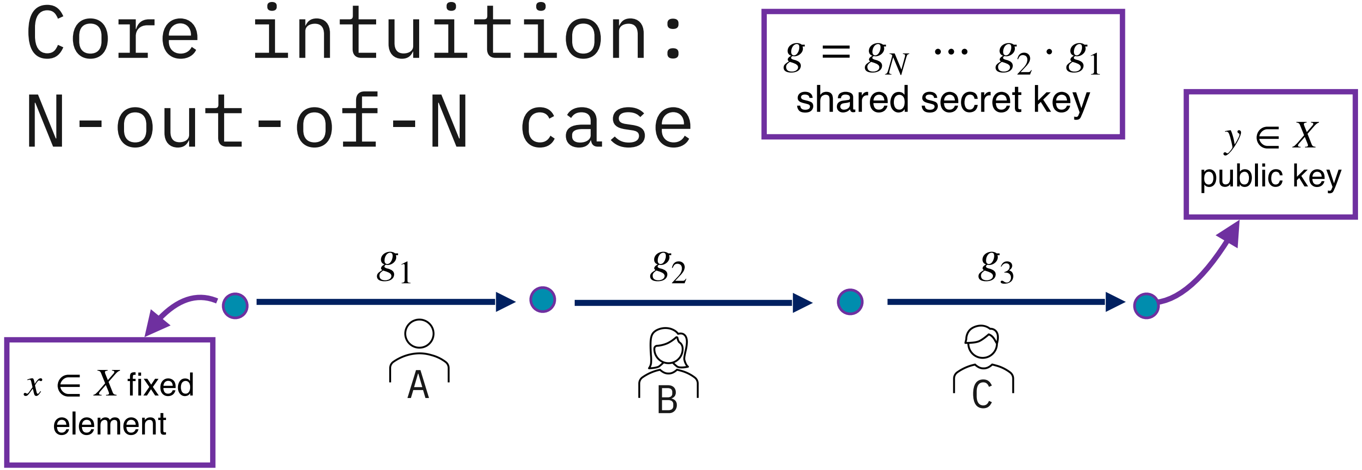


Core intuition: N-out-of-N case

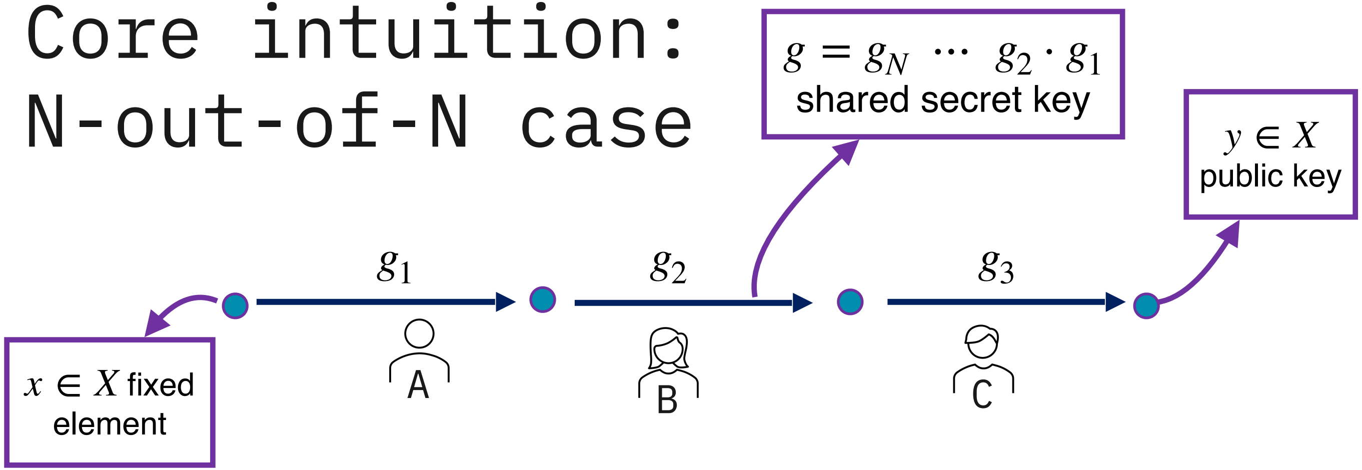
$g = g_N \cdots g_2 \cdot g_1$
shared secret key



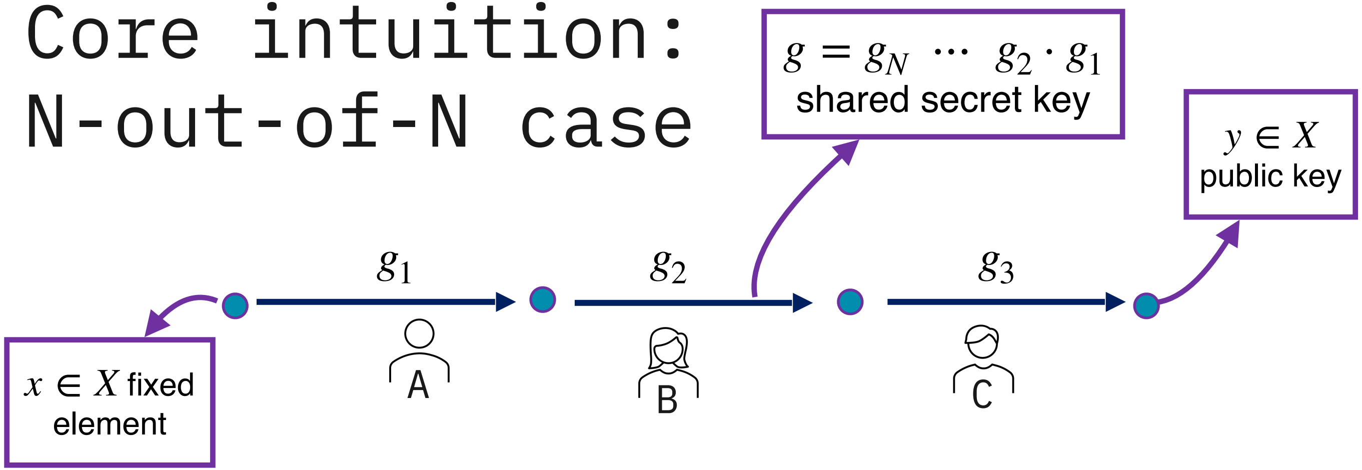
Core intuition: N-out-of-N case



Core intuition: N-out-of-N case

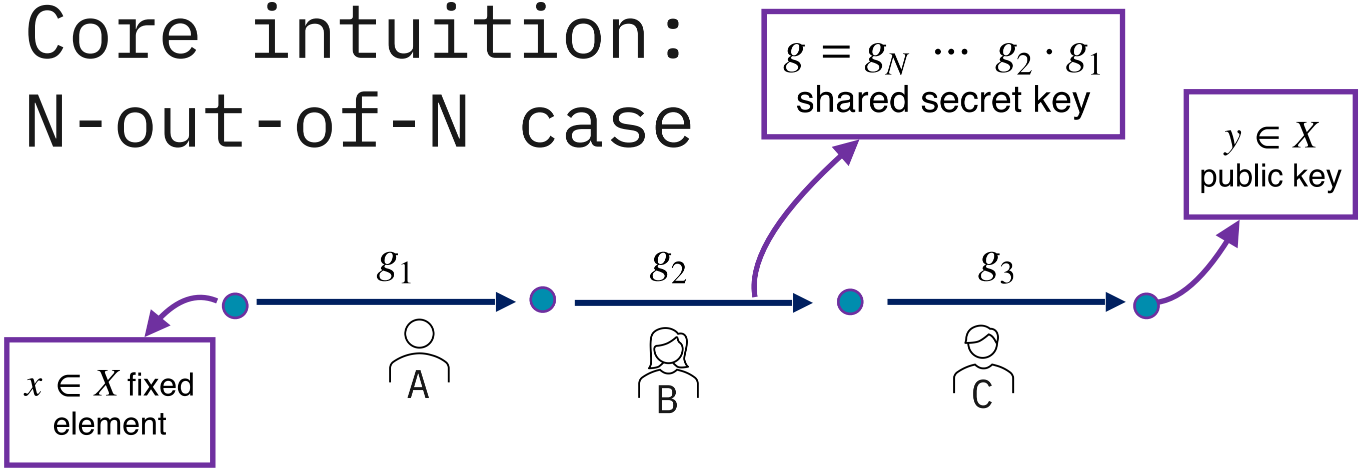


Core intuition: N-out-of-N case



- the intermediate pks are in relation given by:

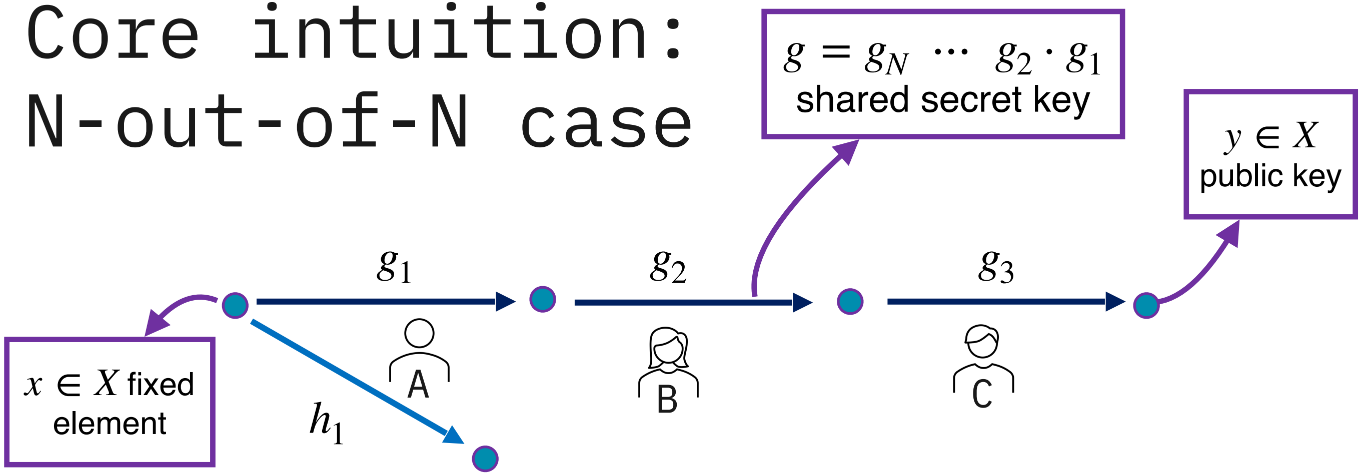
Core intuition: N-out-of-N case



- the intermediate pks are in relation given by:

$$y_{i+1} = g_{i+1} \star y_i$$

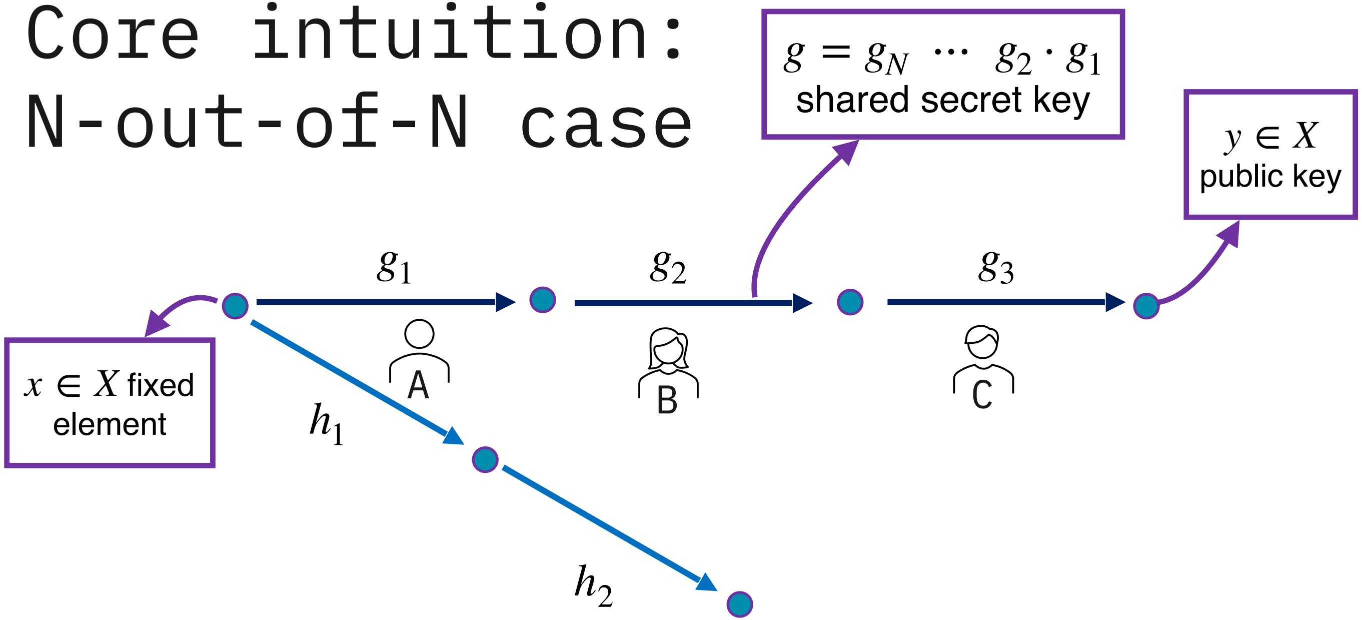
Core intuition: N-out-of-N case



- the intermediate pks are in relation given by:

$$y_{i+1} = g_{i+1} \star y_i$$

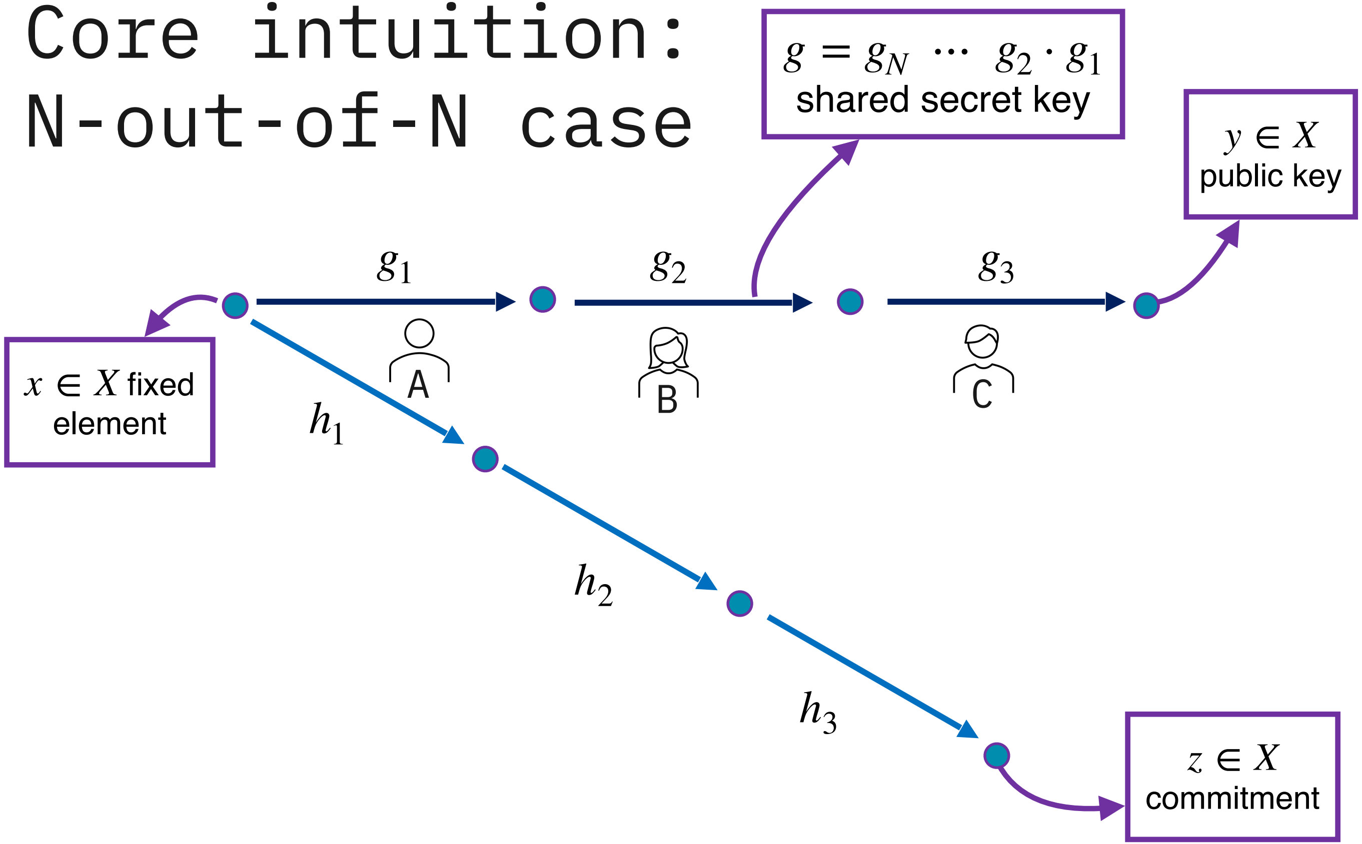
Core intuition: N-out-of-N case



- the intermediate pks are in relation given by:

$$y_{i+1} = g_{i+1} \star y_i$$

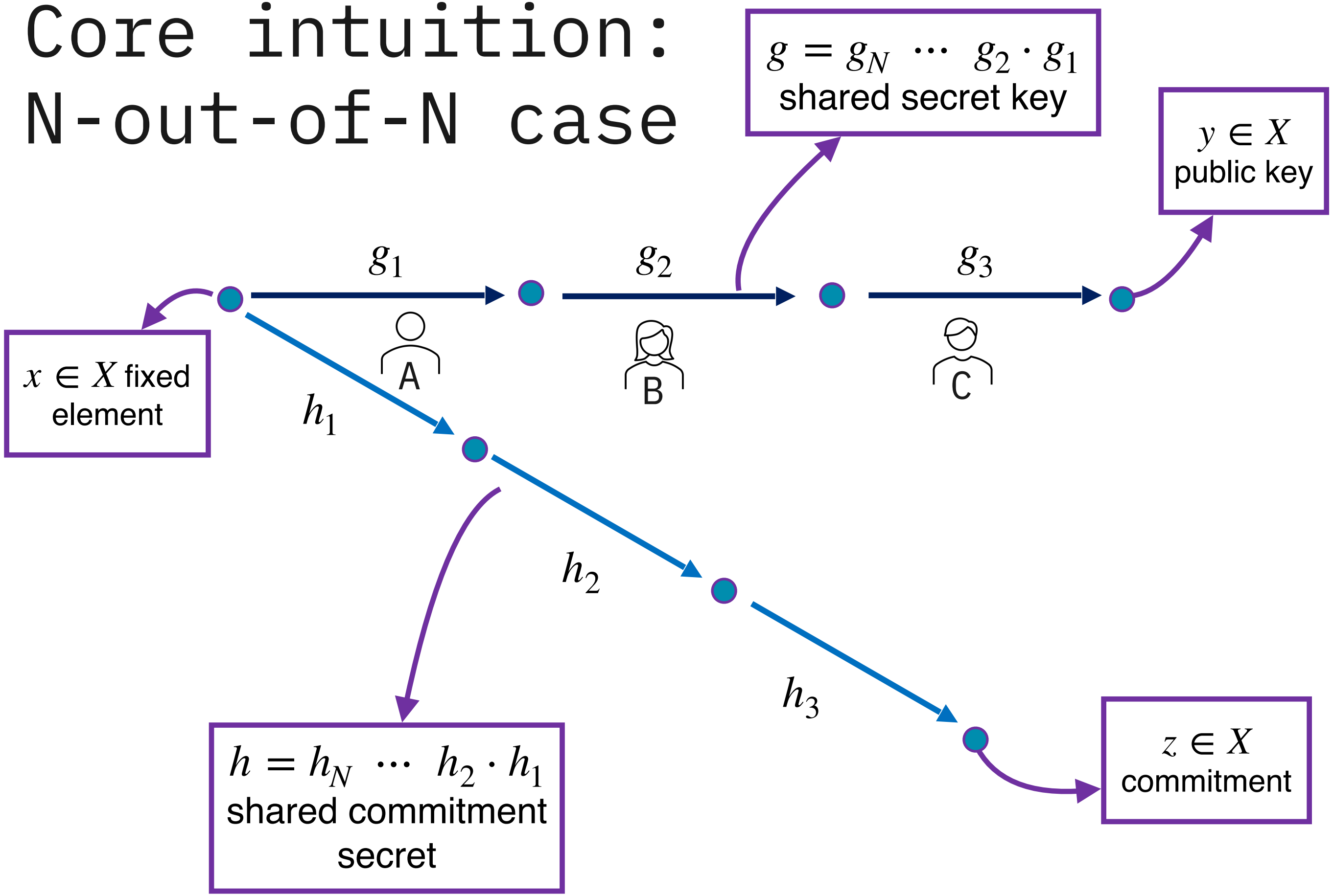
Core intuition: N-out-of-N case



- the intermediate pks are in relation given by:

$$y_{i+1} = g_{i+1} \star y_i$$

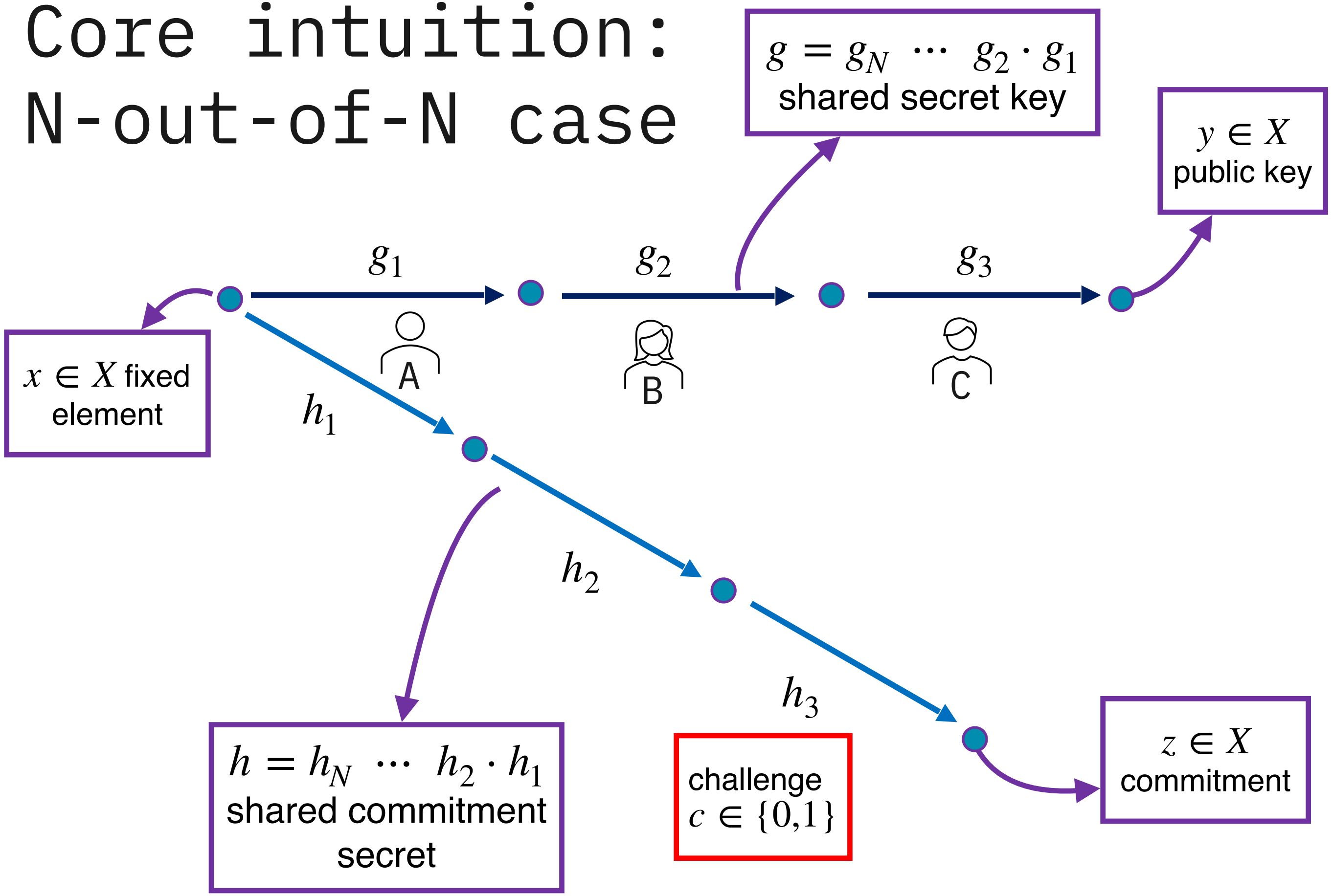
Core intuition: N-out-of-N case



- the intermediate pks are in relation given by:

$$y_{i+1} = g_{i+1} \star y_i$$

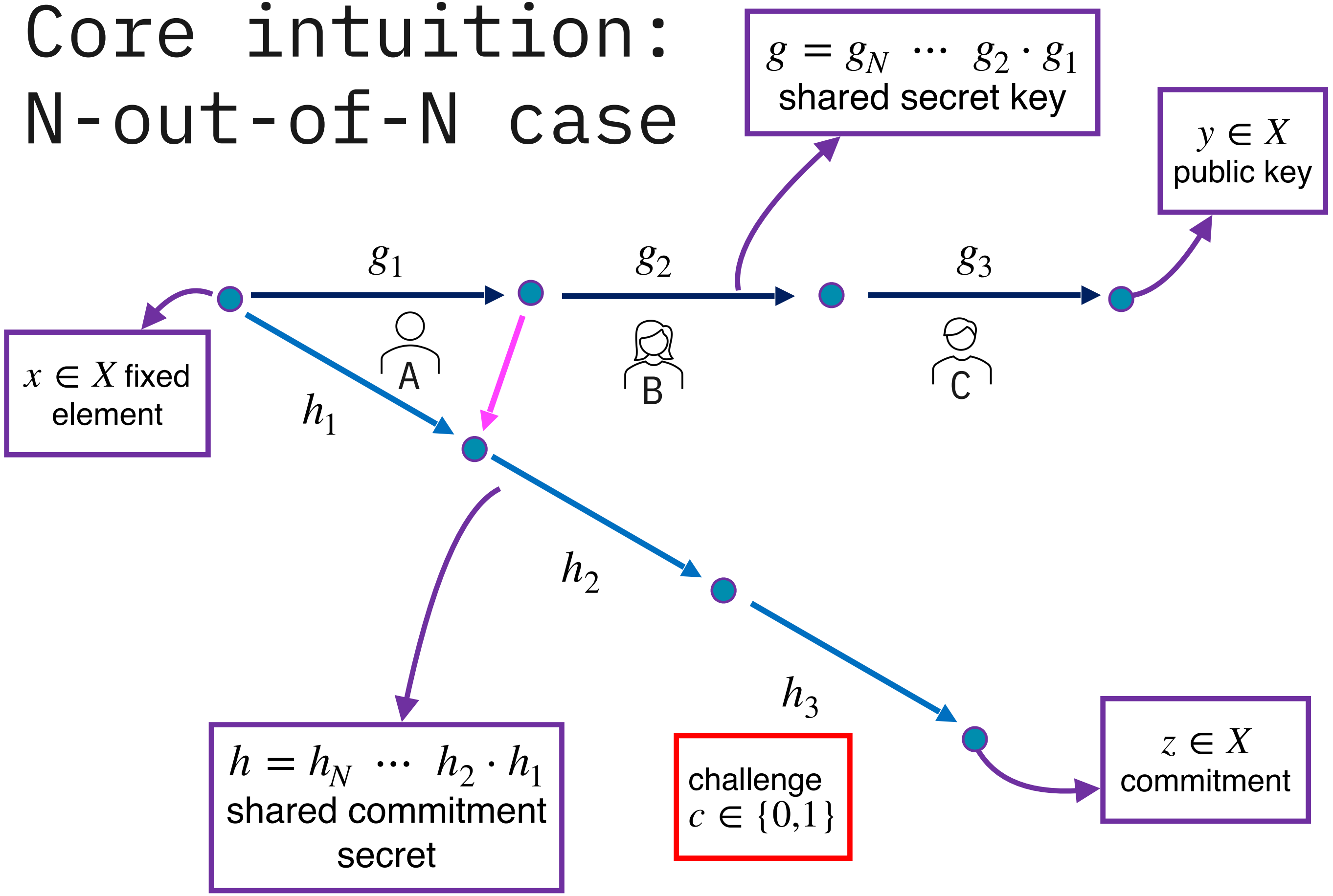
Core intuition: N-out-of-N case



- the intermediate pks are in relation given by:

$$y_{i+1} = g_{i+1} \star y_i$$

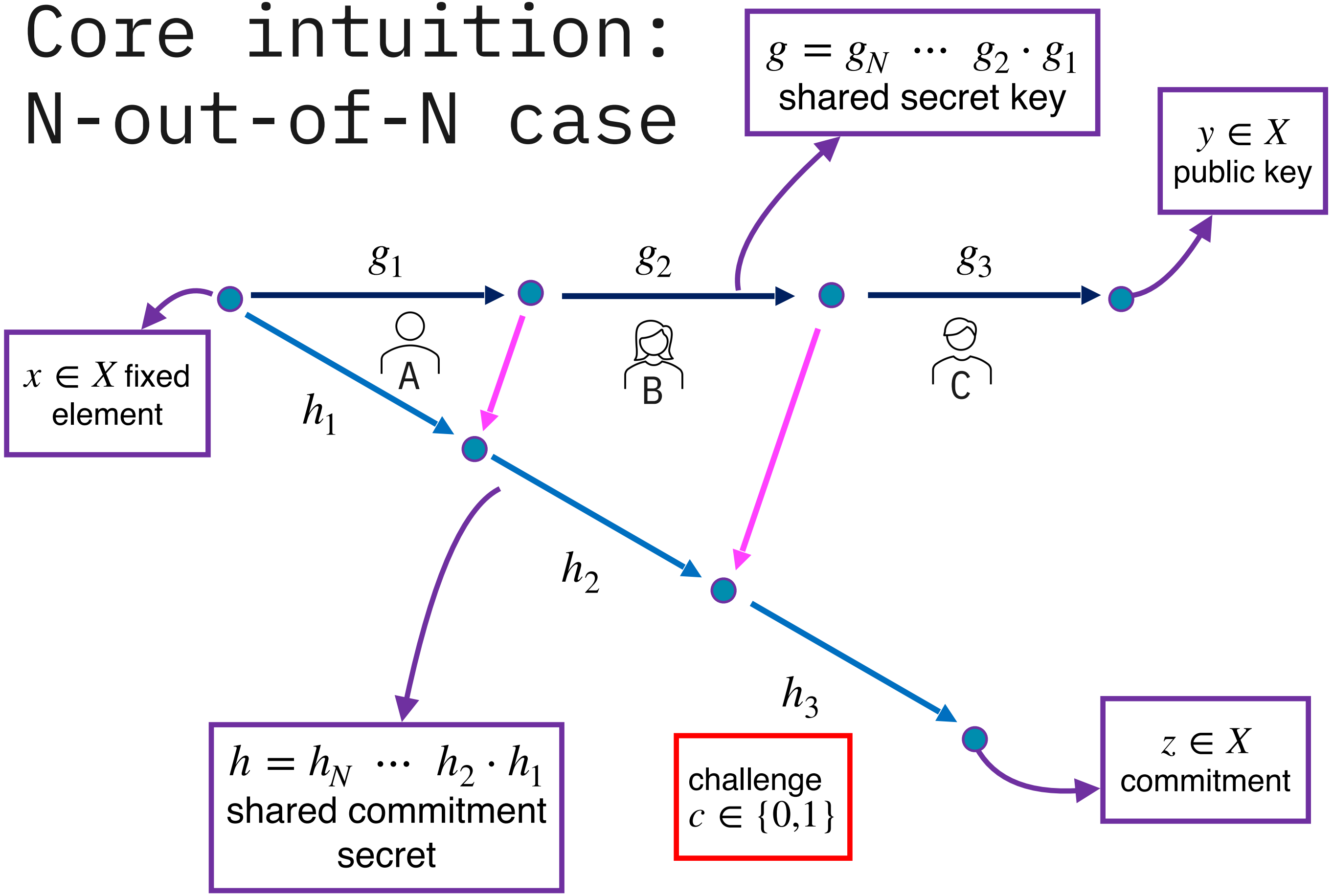
Core intuition: N-out-of-N case



- the intermediate pks are in relation given by:

$$y_{i+1} = g_{i+1} \star y_i$$

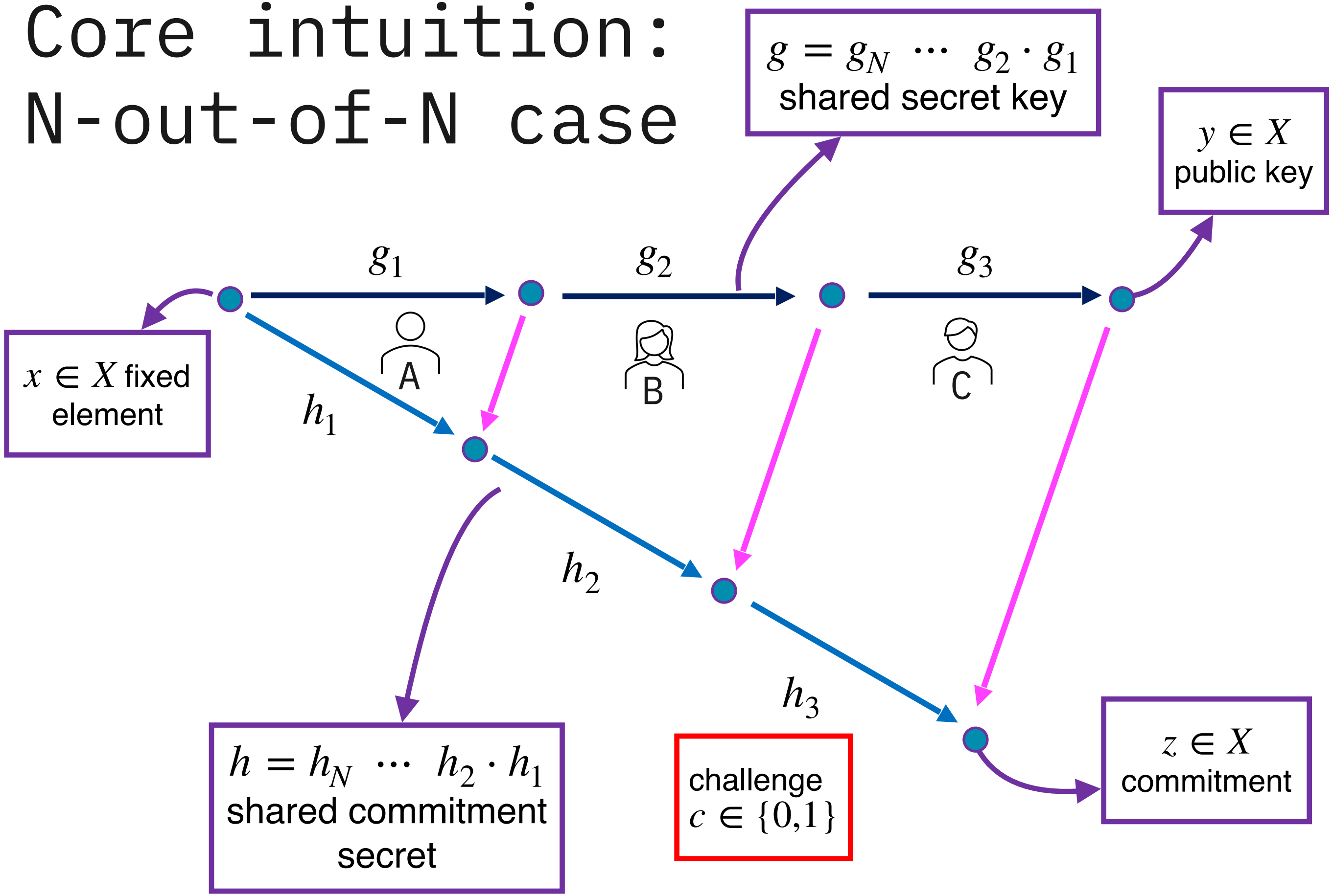
Core intuition: N-out-of-N case



- the intermediate pks are in relation given by:

$$y_{i+1} = g_{i+1} \star y_i$$

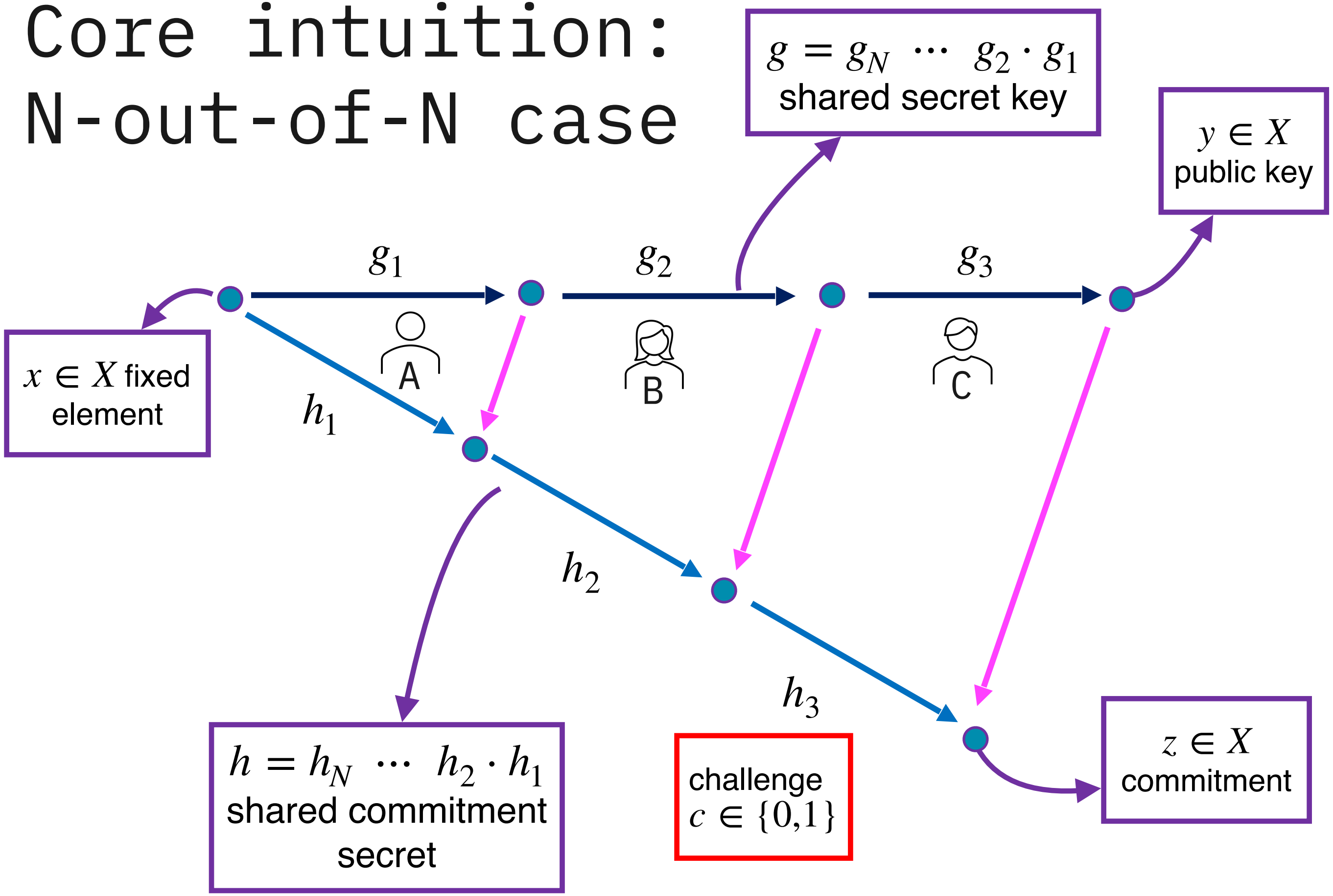
Core intuition: N-out-of-N case



- the intermediate pks are in relation given by:

$$y_{i+1} = g_{i+1} \star y_i$$

Core intuition: N-out-of-N case

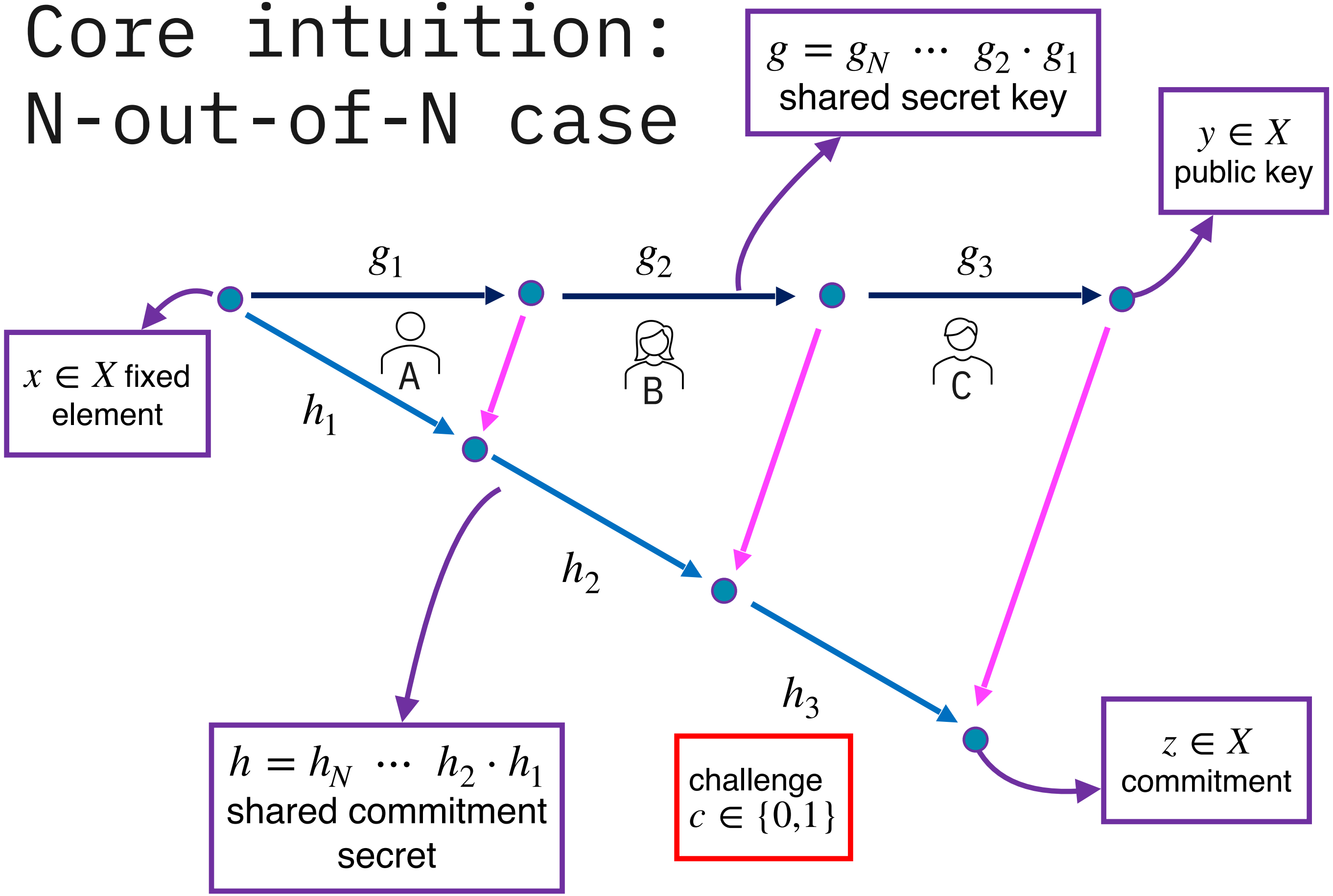


- the intermediate pks are in relation given by:

$$y_{i+1} = g_{i+1} \star y_i$$

- in the abelian case we can compress the response phase to one round

Core intuition: N-out-of-N case



- the intermediate pks are in relation given by:

$$y_{i+1} = g_{i+1} \star y_i$$

- in the abelian case we can compress the response phase to one round

- the hard part is the sharing of the secret, not the commitment

How to make this secure against active attackers?

How to make this secure against active attackers?

- In an active scenario the last user can always perform a basic version of the ROS attack;

How to make this secure against active attackers?

(1) Cozzo D, Smart NP. Sashimi: cutting up CSI-FiSh secret keys to produce an actively secure distributed signing protocol.

- In an active scenario the last user can always perform a basic version of the ROS attack;
- *Solution from (1)*:

How to make this secure against active attackers?

(1) Cozzo D, Smart NP. Sashimi: cutting up CSI-FiSh secret keys to produce an actively secure distributed signing protocol.

- In an active scenario the last user can always perform a basic version of the ROS attack;
- *Solution from (1)*:
  - Add a ZKPoK for <u>every action performed</u> in commitment generation;

How to make this secure against active attackers?

(1) Cozzo D, Smart NP. Sashimi: cutting up CSI-FiSh secret keys to produce an actively secure distributed signing protocol.

- In an active scenario the last user can always perform a basic version of the ROS attack;
- *Solution from (1)*:
  - Add a ZKPoK for <u>every action performed</u> in commitment generation;
  - **Con**: Very inefficient (*memo: Boneh et.al. result*);

How to make this secure against active attackers?

(1) Cozzo D, Smart NP. Sashimi: cutting up CSI-FiSh secret keys to produce an actively secure distributed signing protocol.

- In an active scenario the last user can always perform a basic version of the ROS attack;
- *Solution from (1)*:
  - Add a ZKPoK for <u>every action performed</u> in commitment generation;
  - **Con**: Very inefficient (*memo: Boneh et.al. result*);
  - **Pro**: Simple and imply adaptive security.

How to make this secure against active attackers?

(1) Cozzo D, Smart NP. Sashimi: cutting up CSI-FiSh secret keys to produce an actively secure distributed signing protocol.
(2) Battagliola M, Borin G, Meneghetti A, Persichetti E. Cutting the grass: Threshold group action signature schemes.

- In an active scenario the last user can always perform a basic version of the ROS attack;
- *Solution from (1)*:
  - Add a ZKPoK for <u>every action performed</u> in commitment generation;
  - **Con**: Very inefficient (*memo: Boneh et.al. result*);
  - **Pro**: Simple and imply adaptive security.
- *Solution from (2)*:

How to make this secure against active attackers?

(1) Cozzo D, Smart NP. Sashimi: cutting up CSI-FiSh secret keys to produce an actively secure distributed signing protocol.
(2) Battagliola M, Borin G, Meneghetti A, Persichetti E. Cutting the grass: Threshold group action signature schemes.

- In an active scenario the last user can always perform a basic version of the ROS attack;
- *Solution from (1)*:
  - Add a ZKPoK for <u>every action performed</u> in commitment generation;
  - **Con**: Very inefficient (*memo: Boneh et.al. result*);
  - **Pro**: Simple and imply adaptive security.
- *Solution from (2)*:
  - use secure randomness + verify all intermediate signatures

Universität Zürich[UZH]

How to make this secure against active attackers?

(1) Cozzo D, Smart NP. Sashimi: cutting up CSI-FiSh secret keys to produce an actively secure distributed signing protocol.
(2) Battagliola M, Borin G, Meneghetti A, Persichetti E. Cutting the grass: Threshold group action signature schemes.

- In an active scenario the last user can always perform a basic version of the ROS attack;
- *Solution from (1)*:
  - Add a ZKPoK for <u>every action performed</u> in commitment generation;
  - **Con**: Very inefficient (*memo: Boneh et.al. result*);
  - **Pro**: Simple and imply adaptive security.
- *Solution from (2)*:
  - use secure randomness + verify all intermediate signatures
  - **Pro**: Much more efficient;

Universität Zürich[UZH]

How to make this secure against active attackers?

(1) Cozzo D, Smart NP. Sashimi: cutting up CSI-FiSh secret keys to produce an actively secure distributed signing protocol.
(2) Battagliola M, Borin G, Meneghetti A, Persichetti E. Cutting the grass: Threshold group action signature schemes.

- In an active scenario the last user can always perform a basic version of the ROS attack;
- *Solution from (1)*:
  - Add a ZKPoK for <u>every action performed</u> in commitment generation;
  - **Con**: Very inefficient (*memo: Boneh et.al. result*);
  - **Pro**: Simple and imply adaptive security.
- *Solution from (2)*:
  - use secure randomness + verify all intermediate signatures
  - **Pro**: Much more efficient;
  - **Con**: Requires to know all intermediate public keys.

Cozzo D, Giunta E. Round-robin is optimal: lower bounds for group action based protocols.

Cozzo D, Giunta E. Round-robin is optimal: lower bounds for group action based protocols.

# Rounds				
Complexity				
Share size				

Cozzo D, Giunta E. Round-robin is optimal: lower bounds for group action based protocols.

	Passive, Non-Abelian			
# Rounds	N + N			
Complexity	O(N λ)			
Share size	O(λ)			

Cozzo D, Giunta E. Round-robin is optimal: lower bounds for group action based protocols.

	Passive, Non-Abelian	Passive, Abelian		
# Rounds	N + N	N + 1		
Complexity	O(N λ)	O(N λ)		
Share size	O(λ)	O(λ)		

Cozzo D, Giunta E. Round-robin is optimal: lower bounds for group action based protocols.

	Passive, Non-Abelian	Passive, Abelian	Active, with ZKPs	
# Rounds	N + N	N + 1	N + 1 + 1	
Complexity	$O(N \lambda)$	$O(N \lambda)$	$O(N \lambda^2)$	
Share size	$O(\lambda)$	$O(\lambda)$	$O(\lambda)$	

Cozzo D, Giunta E. Round-robin is optimal: lower bounds for group action based protocols.

	Passive, Non-Abelian	Passive, Abelian	Active, with ZKPs	Active, with Secure Randomness
# Rounds	N + N	N + 1	N + 1 + 1	N + N + 1
Complexity	$O(N \lambda)$	$O(N \lambda)$	$O(N \lambda^2)$	$O(N \lambda)$
Share size	$O(\lambda)$	$O(\lambda)$	$O(\lambda)$	$O(N \lambda)$

Cozzo D, Giunta E. Round-robin is optimal: lower bounds for group action based protocols.

De Feo L, Meyer M. Threshold schemes from isogeny assumptions

How to make this for T-out-of-N ?

Cyclic Case

How to make this for T-out-of-N ? Cyclic Case

De Feo L, Meyer M. Threshold schemes from isogeny assumptions

Shamir Secret Sharing

How to make this for T-out-of-N ?
Cyclic Case

De Feo L, Meyer M. Threshold schemes from isogeny assumptions

Shamir Secret Sharing

- **Idea**: each authorised subset of parties L can write the secret as a linear combination of their shares $s = \lambda_{S,1}s_1 + \cdots + \lambda_{S,T}s_T$, then $y = [\lambda_{S,1}s_1] \cdots [\lambda_{S,T}s_T]\ x$

How to make this for T-out-of-N ? Cyclic Case

De Feo L, Meyer M. Threshold schemes from isogeny assumptions

Shamir Secret Sharing

- **Idea**: each authorised subset of parties L can write the secret as a linear combination of their shares $s = \lambda_{S,1}s_1 + \cdots + \lambda_{S,T}s_T$, then $y = [\lambda_{S,1}s_1]\cdots[\lambda_{S,T}s_T]\, x$
- **Problem 1**: requires G to be a ring with division, but $\#G$ is composite,

How to make this for T-out-of-N ? Cyclic Case

De Feo L, Meyer M. Threshold schemes from isogeny assumptions

Shamir Secret Sharing

- **Idea**: each authorised subset of parties L can write the secret as a linear combination of their shares $s = \lambda_{S,1} s_1 + \cdots + \lambda_{S,T} s_T$, then $y = [\lambda_{S,1} s_1] \cdots [\lambda_{S,T} s_T]\, x$
- **Problem 1**: requires G to be a ring with division, but $\#G$ is composite,
  - **Remark**: the denominator abs is bounded by N

How to make this for T-out-of-N ? Cyclic Case

De Feo L, Meyer M. Threshold schemes from isogeny assumptions

Shamir Secret Sharing

- **Idea**: each authorised subset of parties L can write the secret as a linear combination of their shares $s = \lambda_{S,1} s_1 + \cdots + \lambda_{S,T} s_T$, then $y = [\lambda_{S,1} s_1] \cdots [\lambda_{S,T} s_T]\, x$
- **Problem 1**: requires G to be a ring with division, but $\#G$ is composite,
  - **Remark**: the denominator abs is bounded by N
  - **Solution**: modify the generator so that $N \leq$ all prime factors of $\#G$;

How to make this for T-out-of-N ? Cyclic Case

De Feo L, Meyer M. Threshold schemes from isogeny assumptions

Shamir Secret Sharing

- **Idea**: each authorised subset of parties L can write the secret as a linear combination of their shares $s = \lambda_{S,1}s_1 + \cdots + \lambda_{S,T}s_T$, then $y = [\lambda_{S,1}s_1]\cdots[\lambda_{S,T}s_T]\, x$
- **Problem 1**: requires G to be a ring with division, but $\#G$ is composite,
  - **Remark**: the denominator abs is bounded by N
  - **Solution**: modify the generator so that $N \leq$ all prime factors of $\#G$;
- **Problem 2**: still requires T rounds.

How to make this for T-out-of-N ? Cyclic Case

De Feo L, Meyer M. Threshold schemes from isogeny assumptions

Shamir Secret Sharing

- **Idea**: each authorised subset of parties L can write the secret as a linear combination of their shares $s = \lambda_{S,1}s_1 + \cdots + \lambda_{S,T}s_T$, then $y = [\lambda_{S,1}s_1]\cdots[\lambda_{S,T}s_T]\, x$
- **Problem 1**: requires G to be a ring with division, but $\#G$ is composite,
  - **Remark**: the denominator abs is bounded by N
  - **Solution**: modify the generator so that $N \leq$ all prime factors of $\#G$;
- **Problem 2**: still requires T rounds.
- **Problem 3**: ZKPs becomes much more complicated (PVP)

How to make this for T-out-of-N ?
<u>Non-Abelian Case</u>

Battagliola M, Borin G, Meneghetti A, Persichetti E. Cutting the grass: Threshold group action signature schemes.

How to make this for T-out-of-N ? Non-Abelian Case

Replicated Secret Sharing

Battagliola M, Borin G, Meneghetti A, Persichetti E. Cutting the grass: Threshold group action signature schemes.

How to make this for T-out-of-N ? Non-Abelian Case

Replicated Secret Sharing

- Idea: increase (exponentially) the number of secrets and assign the knowledge to multiple parties;

Battagliola M, Borin G, Meneghetti A, Persichetti E. Cutting the grass: Threshold group action signature schemes.

How to make this for T-out-of-N ? Non-Abelian Case

Replicated Secret Sharing

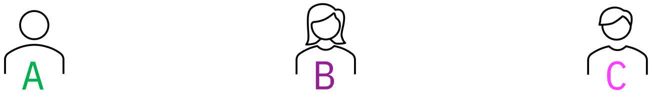
- Idea: increase (exponentially) the number of secrets and assign the knowledge to multiple parties;
- Example: 2-out-of-3 users:

Battagliola M, Borin G, Meneghetti A, Persichetti E. Cutting the grass: Threshold group action signature schemes.

How to make this for T-out-of-N ? <u>Non-Abelian Case</u>

Replicated Secret Sharing

- Idea: increase (<u>exponentially</u>) the number of secrets and assign the knowledge to multiple parties;
- Example: 2-out-of-3 users:

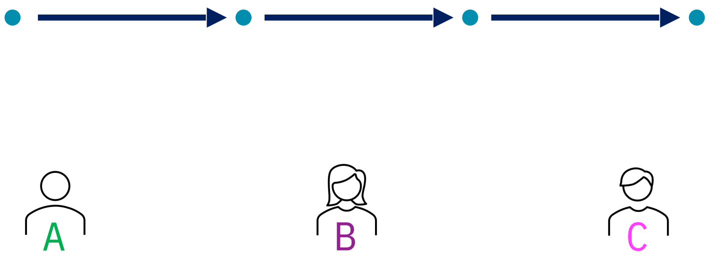


Battagliola M, Borin G, Meneghetti A, Persichetti E. Cutting the grass: Threshold group action signature schemes.

How to make this for T-out-of-N ? <u>Non-Abelian Case</u>

Replicated Secret Sharing

- Idea: increase (<u>exponentially</u>) the number of secrets and assign the knowledge to multiple parties;
- Example: 2-out-of-3 users:

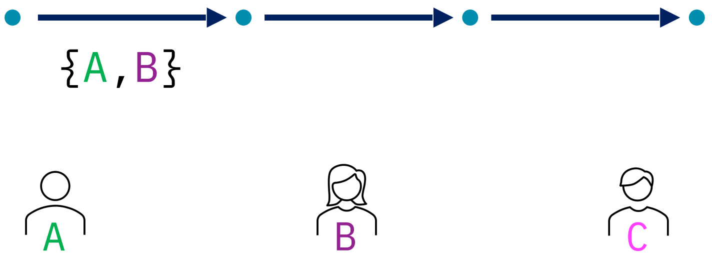


Battagliola M, Borin G, Meneghetti A, Persichetti E. Cutting the grass: Threshold group action signature schemes.

How to make this for T-out-of-N ? <u>Non-Abelian Case</u>

Replicated Secret Sharing

- Idea: increase (<u>exponentially</u>) the number of secrets and assign the knowledge to multiple parties;
- Example: 2-out-of-3 users:

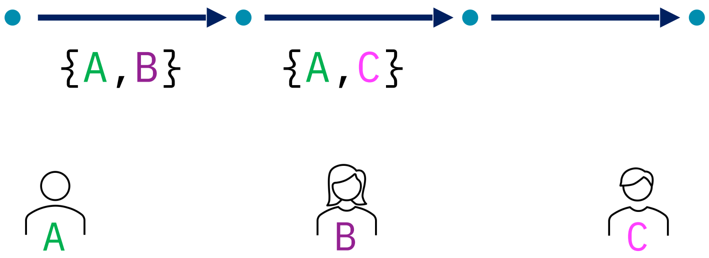


Battagliola M, Borin G, Meneghetti A, Persichetti E. Cutting the grass: Threshold group action signature schemes.

How to make this for T-out-of-N ? Non-Abelian Case

Replicated Secret Sharing

- Idea: increase (exponentially) the number of secrets and assign the knowledge to multiple parties;
- Example: 2-out-of-3 users:

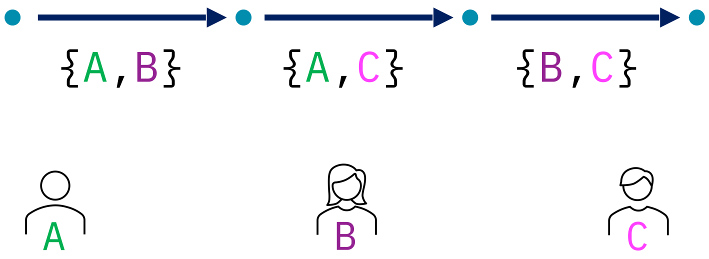


Battagliola M, Borin G, Meneghetti A, Persichetti E. Cutting the grass: Threshold group action signature schemes.

How to make this for T-out-of-N ? Non-Abelian Case

Replicated Secret Sharing

- Idea: increase (exponentially) the number of secrets and assign the knowledge to multiple parties;
- Example: 2-out-of-3 users:

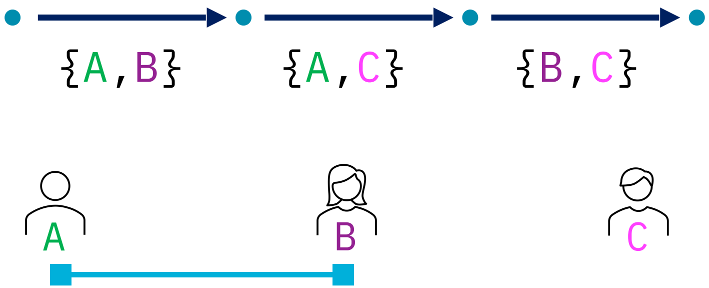


Battagliola M, Borin G, Meneghetti A, Persichetti E. Cutting the grass: Threshold group action signature schemes.

How to make this for T-out-of-N ? Non-Abelian Case

Replicated Secret Sharing

- Idea: increase (exponentially) the number of secrets and assign the knowledge to multiple parties;
- Example: 2-out-of-3 users:

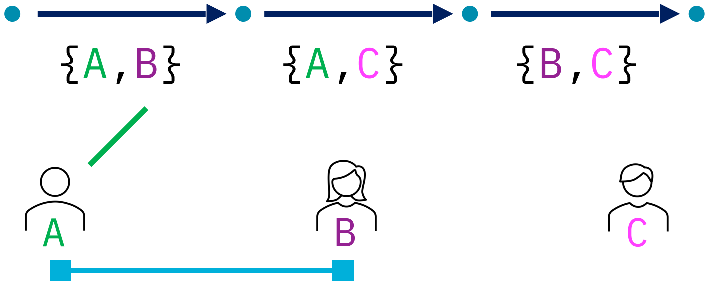


Battagliola M, Borin G, Meneghetti A, Persichetti E. Cutting the grass: Threshold group action signature schemes.

How to make this for T-out-of-N ? Non-Abelian Case

Replicated Secret Sharing

- Idea: increase (exponentially) the number of secrets and assign the knowledge to multiple parties;
- Example: 2-out-of-3 users:

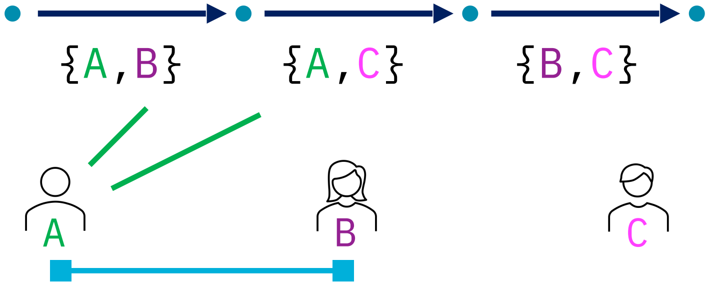


Battagliola M, Borin G, Meneghetti A, Persichetti E. Cutting the grass: Threshold group action signature schemes.

How to make this for T-out-of-N ? Non-Abelian Case

Replicated Secret Sharing

- Idea: increase (exponentially) the number of secrets and assign the knowledge to multiple parties;
- Example: 2-out-of-3 users:

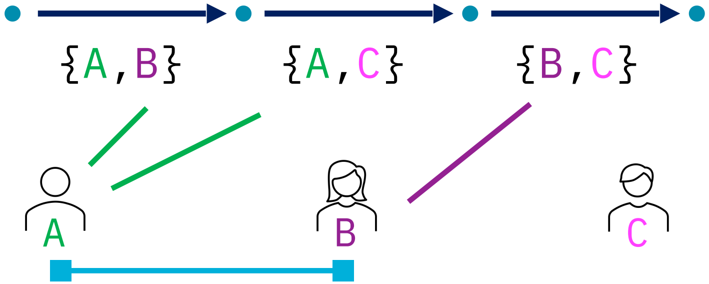


Battagliola M, Borin G, Meneghetti A, Persichetti E. Cutting the grass: Threshold group action signature schemes.

How to make this for T-out-of-N ?
<u>Non-Abelian Case</u>

Replicated Secret Sharing

- Idea: increase (<u>exponentially</u>) the number of secrets and assign the knowledge to multiple parties;
- Example: 2-out-of-3 users:



Battagliola M, Borin G, Meneghetti A, Persichetti E. Cutting the grass: Threshold group action signature schemes.

How to make this for T-out-of-N ? Non-Abelian Case

(1) Desmedt Y, Di Crescenzo G, Burmester M. Multiplicative non-abelian sharing schemes and their application to threshold cryptography.
(2) Battagliola M, Borin G, Di Crescenzo G, Meneghetti A, Persichetti E. Enhancing Threshold Group Action Signature Schemes: Adaptive Security and Scalability Improvements.

Universität Zürich[UZH]

How to make this for T-out-of-N ? Non-Abelian Case

(1) Desmedt Y, Di Crescenzo G, Burmester M. Multiplicative non-abelian sharing schemes and their application to threshold cryptography.
(2) Battagliola M, Borin G, Di Crescenzo G, Meneghetti A, Persichetti E. Enhancing Threshold Group Action Signature Schemes: Adaptive Security and Scalability Improvements.

‘Vandermonde’ Secret Sharing

Universität Zürich[UZH]

How to make this for T-out-of-N ? Non-Abelian Case

(1) Desmedt Y, Di Crescenzo G, Burmester M. Multiplicative non-abelian sharing schemes and their application to threshold cryptography.
(2) Battagliola M, Borin G, Di Crescenzo G, Meneghetti A, Persichetti E. Enhancing Threshold Group Action Signature Schemes: Adaptive Security and Scalability Improvements.

'Vandermonde' Secret Sharing

- Recursive idea, use algorithmically the Vandermonde inequality:

How to make this for T-out-of-N ? Non-Abelian Case

(1) Desmedt Y, Di Crescenzo G, Burmester M. Multiplicative non-abelian sharing schemes and their application to threshold cryptography.
(2) Battagliola M, Borin G, Di Crescenzo G, Meneghetti A, Persichetti E. Enhancing Threshold Group Action Signature Schemes: Adaptive Security and Scalability Improvements.

‘Vandermonde’ Secret Sharing

- Recursive idea, use algorithmically the Vandermonde inequality:

$$\binom{N}{T} = \sum_{k=0}^{T} \binom{b}{k} \cdot \binom{N-b}{T-k}$$

How to make this for T-out-of-N ? Non-Abelian Case

(1) Desmedt Y, Di Crescenzo G, Burmester M. Multiplicative non-abelian sharing schemes and their application to threshold cryptography.
(2) Battagliola M, Borin G, Di Crescenzo G, Meneghetti A, Persichetti E. Enhancing Threshold Group Action Signature Schemes: Adaptive Security and Scalability Improvements.

'Vandermonde' Secret Sharing

- Recursive idea, use algorithmically the Vandermonde inequality:

$$\binom{N}{T} = \sum_{k=0}^{T} \binom{b}{k} \cdot \binom{N-b}{T-k}$$

- Recursive evaluation of T-out-of-N:

How to make this for T-out-of-N ? Non-Abelian Case

(1) Desmedt Y, Di Crescenzo G, Burmester M. Multiplicative non-abelian sharing schemes and their application to threshold cryptography.
(2) Battagliola M, Borin G, Di Crescenzo G, Meneghetti A, Persichetti E. Enhancing Threshold Group Action Signature Schemes: Adaptive Security and Scalability Improvements.

‘Vandermonde’ Secret Sharing

- Recursive idea, use algorithmically the Vandermonde inequality:

$$\binom{N}{T} = \sum_{k=0}^{T} \binom{b}{k} \cdot \binom{N-b}{T-k}$$

- Recursive evaluation of T-out-of-N:
  - If $T = 1$ or $T = N$ share the secret in the ‘obvious way’

Universität Zürich[UZH]

How to make this for T-out-of-N ? Non-Abelian Case

(1) Desmedt Y, Di Crescenzo G, Burmester M. Multiplicative non-abelian sharing schemes and their application to threshold cryptography.
(2) Battagliola M, Borin G, Di Crescenzo G, Meneghetti A, Persichetti E. Enhancing Threshold Group Action Signature Schemes: Adaptive Security and Scalability Improvements.

‘Vandermonde’ Secret Sharing

- Recursive idea, use algorithmically the Vandermonde inequality:

$$\binom{N}{T} = \sum_{k=0}^{T} \binom{b}{k} \cdot \binom{N-b}{T-k}$$

- Recursive evaluation of T-out-of-N:
  - If $T = 1$ or $T = N$ share the secret in the ‘obvious way’
  - If $T \leq 0$ or $T > N$ ignore the sharing

How to make this for T-out-of-N ? Non-Abelian Case

(1) Desmedt Y, Di Crescenzo G, Burmester M. Multiplicative non-abelian sharing schemes and their application to threshold cryptography.
(2) Battagliola M, Borin G, Di Crescenzo G, Meneghetti A, Persichetti E. Enhancing Threshold Group Action Signature Schemes: Adaptive Security and Scalability Improvements.

'Vandermonde' Secret Sharing

- Recursive idea, use algorithmically the Vandermonde inequality:

$$\binom{N}{T} = \sum_{k=0}^{T} \binom{b}{k} \cdot \binom{N-b}{T-k}$$

- Recursive evaluation of T-out-of-N:
  - If $T = 1$ or $T = N$ share the secret in the 'obvious way'
  - If $T \leq 0$ or $T > N$ ignore the sharing
  - Otherwise:

How to make this for T-out-of-N ? Non-Abelian Case

(1) Desmedt Y, Di Crescenzo G, Burmester M. Multiplicative non-abelian sharing schemes and their application to threshold cryptography.
(2) Battagliola M, Borin G, Di Crescenzo G, Meneghetti A, Persichetti E. Enhancing Threshold Group Action Signature Schemes: Adaptive Security and Scalability Improvements.

'Vandermonde' Secret Sharing

- Recursive idea, use algorithmically the Vandermonde inequality:

$$\binom{N}{T} = \sum_{k=0}^{T} \binom{b}{k} \cdot \binom{N-b}{T-k}$$

- Recursive evaluation of T-out-of-N:
  - If $T = 1$ or $T = N$ share the secret in the 'obvious way'
  - If $T \leq 0$ or $T > N$ ignore the sharing
  - Otherwise:
    - divide in two groups of size $\approx N/2$

How to make this for T-out-of-N ? Non-Abelian Case

(1) Desmedt Y, Di Crescenzo G, Burmester M. Multiplicative non-abelian sharing schemes and their application to threshold cryptography.
(2) Battagliola M, Borin G, Di Crescenzo G, Meneghetti A, Persichetti E. Enhancing Threshold Group Action Signature Schemes: Adaptive Security and Scalability Improvements.

'Vandermonde' Secret Sharing

- Recursive idea, use algorithmically the Vandermonde inequality:

$$\binom{N}{T} = \sum_{k=0}^{T} \binom{b}{k} \cdot \binom{N-b}{T-k}$$

- Recursive evaluation of T-out-of-N:
  - If $T = 1$ or $T = N$ share the secret in the 'obvious way'
  - If $T \leq 0$ or $T > N$ ignore the sharing
  - Otherwise:
    - divide in two groups of size $\approx N/2$
    - for each k do a k-out-of-$N/2$ and $T - k$-out-of-$N/2$ sharing

Replicated Secret Sharing

- Less efficient, but simpler

Replicated Secret Sharing

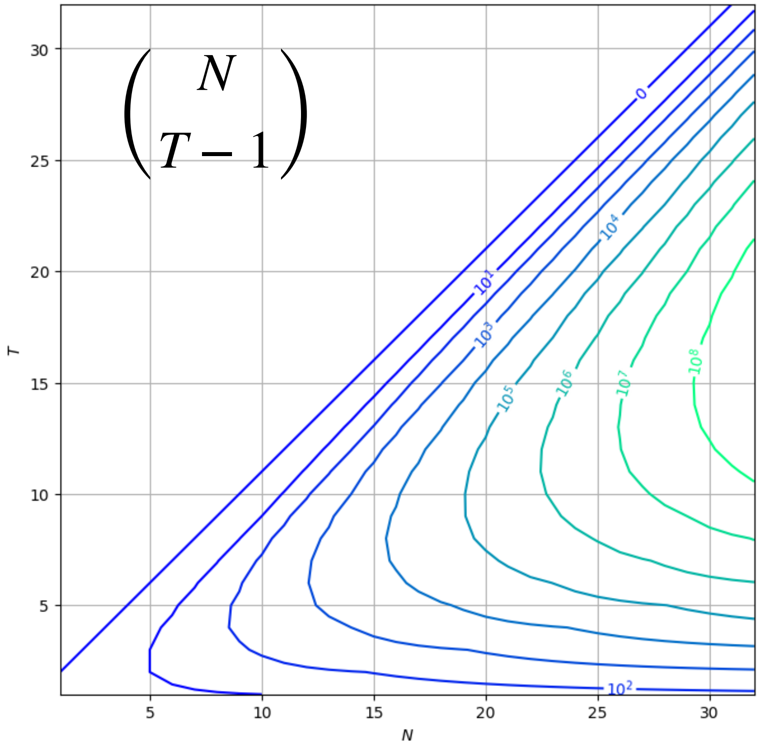
- Less efficient, but simpler

‘Vandermonde’ Secret Sharing

- More complicated, but efficient

Replicated Secret Sharing

- Less efficient, but simpler

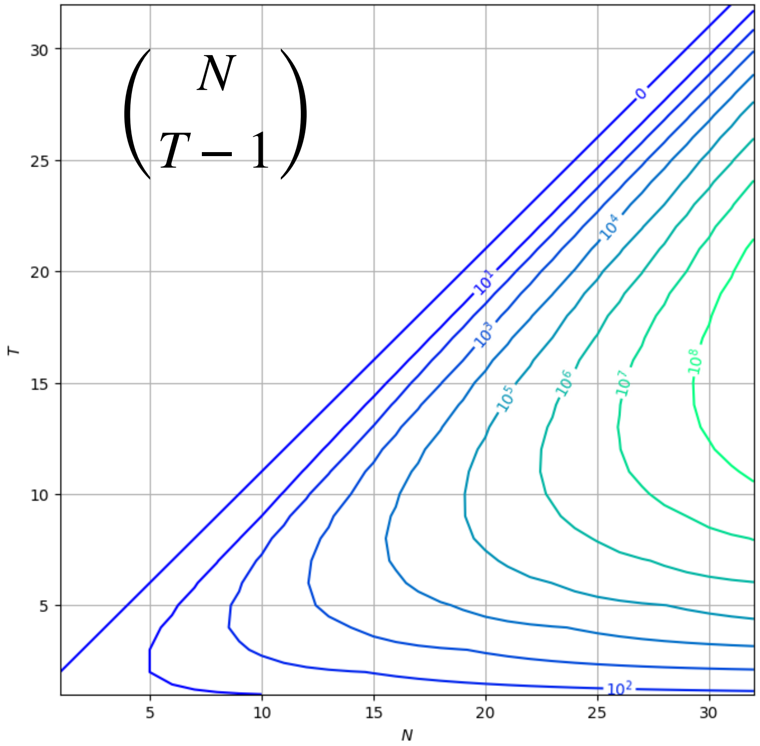


'Vandermonde' Secret Sharing

- More complicated, but efficient

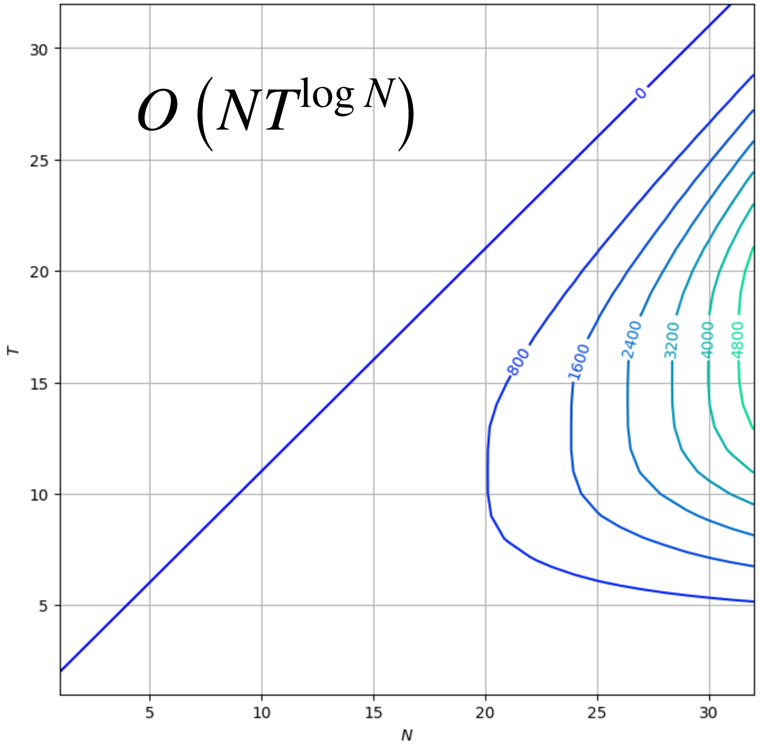
Replicated Secret Sharing

- Less efficient, but simpler



‘Vandermonde’ Secret Sharing

- More complicated, but efficient



How to make this for T-out-of-N ? Abelian Case (open)

How to make this for T-out-of-N ? Abelian Case (open)

- **Problem:** no field like structure (since $\#G$ is unknown):

How to make this for T-out-of-N ? Abelian Case (open)

- **Problem**: no field like structure (since $\#G$ is unknown):

$$\lambda_{S,i} = \frac{\prod_{j \in S} j}{\prod_{j \in S} (j - i)}$$

How to make this for T-out-of-N ? Abelian Case (open)

- **Problem**: no field like structure (since $\#G$ is unknown):

$$\lambda_{S,i} = \frac{\prod_{j \in S} j}{\boxed{\prod_{j \in S} (j - i)}}$$

I need to make sense of this division!

Universität Zürich UZH

How to make this for T-out-of-N ? Abelian Case (open)

- **Problem**: no field like structure (since $\#G$ is unknown):

$$\lambda_{S,i} = \frac{\prod_{j \in S} j}{\boxed{\prod_{j \in S} (j - i)}}$$

I need to make sense of this division!

- **Note**: this is the same problem they had with RSA.

How to make this for T-out-of-N ? Abelian Case (open)

- **Problem**: no field like structure (since $\#G$ is unknown):

$$\lambda_{S,i} = \frac{\prod_{j \in S} j}{\boxed{\prod_{j \in S} (j - i)}}$$

I need to make sense of this division!

- **Note**: this is the same problem they had with RSA.
- **Solution 1a**: work on $\mathbb{Z}$ and use LISS, not compatible with PVP.

Universität Zürich[UZH]

How to make this for T-out-of-N ? <u>Abelian Case</u> (open)

- **Problem**: no field like structure (since $\#G$ is unknown):

$$\lambda_{S,i} = \frac{\prod_{j\in S} j}{\boxed{\prod_{j\in S}(j-i)}}$$

I need to make sense of this division!

- **Note**: this is the same problem they had with RSA.
- **Solution 1a**: work on $\mathbb{Z}$ and use LISS, not compatible with PVP.
- **Solution 1b**: multiply by $N!$ so we are in $\mathbb{Z}$ (compatible with PVP?)

How to make this for T-out-of-N ? Abelian Case (open)

- **Problem**: no field like structure (since $\#G$ is unknown):

$$\lambda_{S,i} = \frac{\prod_{j \in S} j}{\boxed{\prod_{j \in S} (j - i)}}$$

I need to make sense of this division!

- **Note**: this is the same problem they had with RSA.
- **Solution 1a**: work on $\mathbb{Z}$ and use LISS, not compatible with PVP.
- **Solution 1b**: multiply by $N!$ so we are in $\mathbb{Z}$ (compatible with PVP?)
- **Solution 2**: use previous Vandermonde Sharing:

How to make this for T-out-of-N ? Abelian Case (open)

- **Problem**: no field like structure (since $\#G$ is unknown):

$$\lambda_{S,i} = \frac{\prod_{j \in S} j}{\boxed{\prod_{j \in S} (j-i)}}$$

I need to make sense of this division!

- **Note**: this is the same problem they had with RSA.
- **Solution 1a**: work on $\mathbb{Z}$ and use LISS, not compatible with PVP.
- **Solution 1b**: multiply by $N!$ so we are in $\mathbb{Z}$ (compatible with PVP?)
- **Solution 2**: use previous Vandermonde Sharing:
  - Active security with ZKPs or with Secure Randomness

# Rounds				
Signing Complexity				
Share size				

	Shamir			
	Cyclic			
# Rounds	$2T + 1$			
Signing Complexity	$O(N\ \lambda^2)$			
Share size	$O(1)$			

	Shamir	Replicated		
	Cyclic	Non-Abelian		
# Rounds	2T + 1	$2\binom{N}{T-1}+1$		
Signing Complexity	O(N λ^2)	$O\left(\binom{N}{T-1}\lambda\right)$		
Share size	O(1)	$O\left(\binom{N}{T-1}\lambda\right)$		

	Shamir	Replicated	Vandermonde
	Cyclic	Non-Abelian	Non-Abelian
# Rounds	2T + 1	$2\binom{N}{T-1}+1$	2T + 1
Signing Complexity	O(N λ^2)	$O\left(\binom{N}{T-1}\lambda\right)$	O(T λ)
Share size	O(1)	$O\left(\binom{N}{T-1}\lambda\right)$	$O(NT^{\log N}\lambda)$

	Shamir	Replicated	Vandermonde	Vandermonde
	Cyclic	Non-Abelian	Non-Abelian	Abelian
# Rounds	2T + 1	$2\binom{N}{T-1}+1$	2T + 1	T + 2
Signing Complexity	O(N λ^2)	$O\left(\binom{N}{T-1}\lambda\right)$	O(T λ)	O(T λ)
Share size	O(1)	$O\left(\binom{N}{T-1}\lambda\right)$	$O(NT^{\log N}\lambda)$	$O(NT^{\log N}\lambda)$

How to distribute the generation of the key? (open)

(1) Atapoor S, Baghery K, Cozzo D, Pedersen R. CSI-SharK: CSI-FiSh with sharing-friendly keys.
(2) Frixons P, Gilchrist V, Kutas P, Merz SP, Petit C. Another Look at the Quantum Security of the Vectorization Problem with Shifted Inputs.
(3) Cozzo D, Smart NP. Sashimi: cutting up CSI-FiSh secret keys to produce an actively secure distributed signing protocol.

How to distribute the generation of the key? (open)

(1) Atapoor S, Baghery K, Cozzo D, Pedersen R. CSI-SharK: CSI-FiSh with sharing-friendly keys.
(2) Frixons P, Gilchrist V, Kutas P, Merz SP, Petit C. Another Look at the Quantum Security of the Vectorization Problem with Shifted Inputs.
(3) Cozzo D, Smart NP. Sashimi: cutting up CSI-FiSh secret keys to produce an actively secure distributed signing protocol.

- **Option 1**: CSI-SharK (1) introduces PVP,

Universität Zürich UZH

How to distribute the generation of the key? (open)

(1) Atapoor S, Baghery K, Cozzo D, Pedersen R. CSI-SharK: CSI-FiSh with sharing-friendly keys.
(2) Frixons P, Gilchrist V, Kutas P, Merz SP, Petit C. Another Look at the Quantum Security of the Vectorization Problem with Shifted Inputs.
(3) Cozzo D, Smart NP. Sashimi: cutting up CSI-FiSh secret keys to produce an actively secure distributed signing protocol.

- **Option 1**: CSI-SharK (1) introduces PVP,
  - Requires an assumption that is *quantumly broken* (2).

How to distribute the generation of the key? (open)

(1) Atapoor S, Baghery K, Cozzo D, Pedersen R. CSI-SharK: CSI-FiSh with sharing-friendly keys.
(2) Frixons P, Gilchrist V, Kutas P, Merz SP, Petit C. Another Look at the Quantum Security of the Vectorization Problem with Shifted Inputs.
(3) Cozzo D, Smart NP. Sashimi: cutting up CSI-FiSh secret keys to produce an actively secure distributed signing protocol.

- **Option 1**: CSI-SharK (1) introduces PVP,
  - Requires an assumption that is *quantumly broken* (2).
- **Option 2**: Sashimi (3) approach in the KeyGen,

How to distribute the generation of the key? (open)

(1) Atapoor S, Baghery K, Cozzo D, Pedersen R. CSI-SharK: CSI-FiSh with sharing-friendly keys.
(2) Frixons P, Gilchrist V, Kutas P, Merz SP, Petit C. Another Look at the Quantum Security of the Vectorization Problem with Shifted Inputs.
(3) Cozzo D, Smart NP. Sashimi: cutting up CSI-FiSh secret keys to produce an actively secure distributed signing protocol.

- **Option 1**: CSI-SharK (1) introduces PVP,
  - Requires an assumption that is *quantumly broken* (2).
- **Option 2**: Sashimi (3) approach in the KeyGen,
  - Working, but the number of iterations is $\binom{N}{T-1}$,

How to distribute the generation of the key? (open)

(1) Atapoor S, Baghery K, Cozzo D, Pedersen R. CSI-SharK: CSI-FiSh with sharing-friendly keys.
(2) Frixons P, Gilchrist V, Kutas P, Merz SP, Petit C. Another Look at the Quantum Security of the Vectorization Problem with Shifted Inputs.
(3) Cozzo D, Smart NP. Sashimi: cutting up CSI-FiSh secret keys to produce an actively secure distributed signing protocol.

- **Option 1**: CSI-SharK (1) introduces PVP,
  - Requires an assumption that is *quantumly broken* (2).
- **Option 2**: Sashimi (3) approach in the KeyGen,
  - Working, but the number of iterations is $\binom{N}{T-1}$,
  - Assumption *not secure for the non-abelian case*,

How to distribute the generation of the key? (open)

(1) Atapoor S, Baghery K, Cozzo D, Pedersen R. CSI-SharK: CSI-FiSh with sharing-friendly keys.
(2) Frixons P, Gilchrist V, Kutas P, Merz SP, Petit C. Another Look at the Quantum Security of the Vectorization Problem with Shifted Inputs.
(3) Cozzo D, Smart NP. Sashimi: cutting up CSI-FiSh secret keys to produce an actively secure distributed signing protocol.

- **Option 1**: CSI-SharK (1) introduces PVP,
  - Requires an assumption that is *quantumly broken* (2).
- **Option 2**: Sashimi (3) approach in the KeyGen,
  - Working, but the number of iterations is $\binom{N}{T-1}$,
  - Assumption *not secure for the non-abelian case*,
  - **Option 2b**: Extractable ZKPoK with commitment at the start to the secret.

How to distribute the generation of the key? (open)

(1) Atapoor S, Baghery K, Cozzo D, Pedersen R. CSI-SharK: CSI-FiSh with sharing-friendly keys.
(2) Frixons P, Gilchrist V, Kutas P, Merz SP, Petit C. Another Look at the Quantum Security of the Vectorization Problem with Shifted Inputs.
(3) Cozzo D, Smart NP. Sashimi: cutting up CSI-FiSh secret keys to produce an actively secure distributed signing protocol.

- **Option 1**: CSI-SharK (1) introduces PVP,
  - Requires an assumption that is *quantumly broken* (2).
- **Option 2**: Sashimi (3) approach in the KeyGen,
  - Working, but the number of iterations is $\binom{N}{T-1}$,
  - Assumption *not secure for the non-abelian case*,
  - **Option 2b**: Extractable ZKPoK with commitment at the start to the secret.
- **Option 3** *[OPEN]*: can we have DKG for the Vandermonde Sharing?

Thanks