Diagrammatic Reasoning in Cryptography and Cryptanalysis

Peter M. Hines

York Center for Complex Systems Analysis

University of York

P.S.S.L. – 2017

The very basics

Cryptography

The art and science of ensuring information can only be understood by *certain people*.

Cryptanalysis

The art and science of ensuring you are *one of those people*.

"It is clear that the cryptographers are winning the information war . . .

. . . experience tells us that every unbreakable cipher eventually succumbs to cryptanalysis."

– The Code Book, Simon Singh

Protocols as diagrams

Aims and Objectives:

1. Express entire protocols as categorical diagrams.
2. Use a single diagram to model
   - Underlying Algebra
     Commuting diagrams
   - Knowledge of Participants
     Partial order enrichment
   - Information flow
     2-categorical structure
3. Use these to ~~attack~~ study protocols.

Public key cryptography

The aim ...

For participants to come to *share a secret*, using only *public communication*.

"Secure communication over insecure channels" – R. Merkel

Is it possible for **Alice** and **Bob** to share a secret without ever having to meet?

Public Key Distribution

Alice and Bob can come to share a secret, even when all their communications are being monitored.

Diffie – Hellman (– Merkel) key exchange (1976)

- Relies on the *difficulty* of computing discrete logarithms.
- Very heavily used online.
- Highly vunerable to *quantum computers*.

Security through obscurity?

Previously discovered by Ellis, Cocks, Williamson of GCHQ.

A motivating thought-experiment

Prior to D.-H (or E-C-W), it was believed that such secret-sharing should be possible.

The 'untrusted courier' scenario

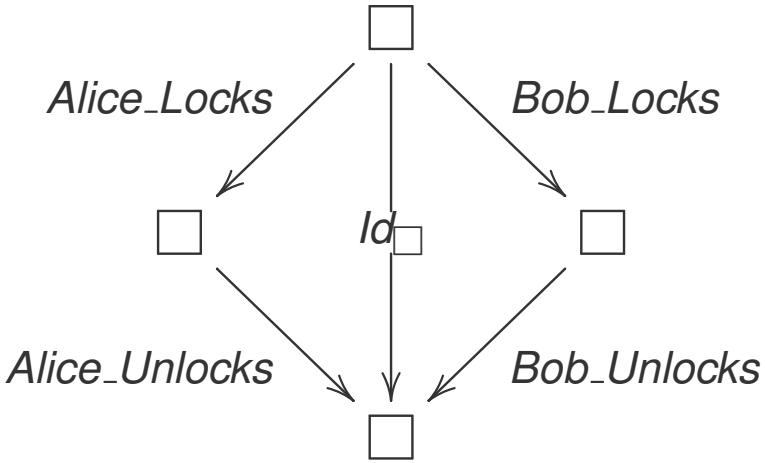
Alice wishes to send Bob some physical object.

- Alice padlocks it into a box & sends the locked box to Bob.
- Bob is unable to open it; he secures the box with his own padlock & returns it to Alice.
- Alice is unable to open it; she removes her padlock & sends it back to Bob.
- Bob receives a box that is secured with his padlock only.

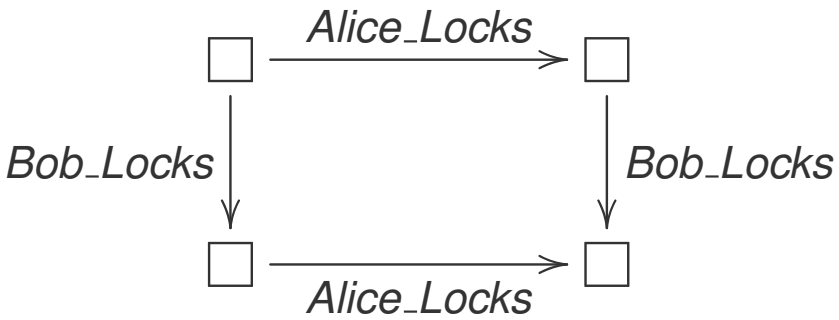
Commutativity & the untrusted courier

Algebraic requirements ...

- Locking operations have left inverses.



- Locking operations commute with each other.



Epistemic requirements ...

- Only **Alice** can perform:
  - *Alice_locks* : $\square \to \square$
  - *Alice_Unlocks* : $\square \to \square$
- Only **Bob** can perform:
  - *Bob_Locks* : $\square \to \square$
  - *Bob_Unlocks* : $\square \to \square$

A family of key exchange protocols

For obvious (quantum) reasons, we seek secret-sharing protocols that are not based on prime fields / factorization / discrete logarithms / etc.

Recent work (January 2017) suggests that *graph isomorphism* is also not a good place to start:

> *"Graph isomorphism in quasi-polynomial time" – László Babai, Univ. Chicago*

We will look at some proposed *algebraic* protocols instead.

An algebraic approach to secret sharing

Commuting Action Key Exchange (CAKE)

- A general family of key exchange (secret sharing) protocols.
- Introduced in 2004 by V. Shpilrain & G. Zapata
- Includes many interesting protocols as special cases

  (Ko-Lee key exchange, Braid group protocols, Shpilrain – Ushakov protocol, &c..).

We will look at the semigroup (monoid) version:

Example 3, Section 3 of *Combinatorial Group Theory and Public Key Cryptography* S.-Z. (2004).

The CAKE – sharing protocol

Alice and Bob will come to share a secret element of a monoid $\mathcal{M}$.

1. Alice and Bob both have large **key pools** $A, B \subseteq \mathcal{M}$ that satisfy
$$ab = ba \;\; \forall \; a \in A, \;\; b \in B.$$
2. A fixed public **root element** $\gamma \in \mathcal{M}$ is chosen.
3. Alice chooses her **private key**, $(\alpha_1, \alpha_2) \in A \times A$, and publicly broadcasts $\alpha_1 \gamma \alpha_2 \in \mathcal{M}$
4. Bob chooses his **private key**,$(\beta_1, \beta_2) \in B \times B$, and publicly broadcasts $\beta_1 \gamma \beta_2 \in \mathcal{M}$.
5. Alice computes $\alpha_1 \beta_1 \gamma \beta_2 \alpha_2$ and Bob computes $\beta_1 \alpha_1 \gamma \alpha_2 \beta_2$.

By the point-wise commutativity of $A, B \subseteq \mathcal{M}$, these are equal, giving Alice and Bob's **shared secret** σ as

$$\sigma \;\; = \;\; \alpha_1 \beta_1 \gamma \beta_2 \alpha_2 \;\; = \;\; \beta_1 \alpha_1 \gamma \alpha_2 \beta_2$$

An important point ...

This is a **general prescription** for building protocols.

It says *nothing* about security ... this depends on the properties of the ~~monoid~~ category $\mathcal{M}$.

Desirable properties for $\mathcal{M}$ were described by Shpilrain et al in 2004.

In a clearer form!

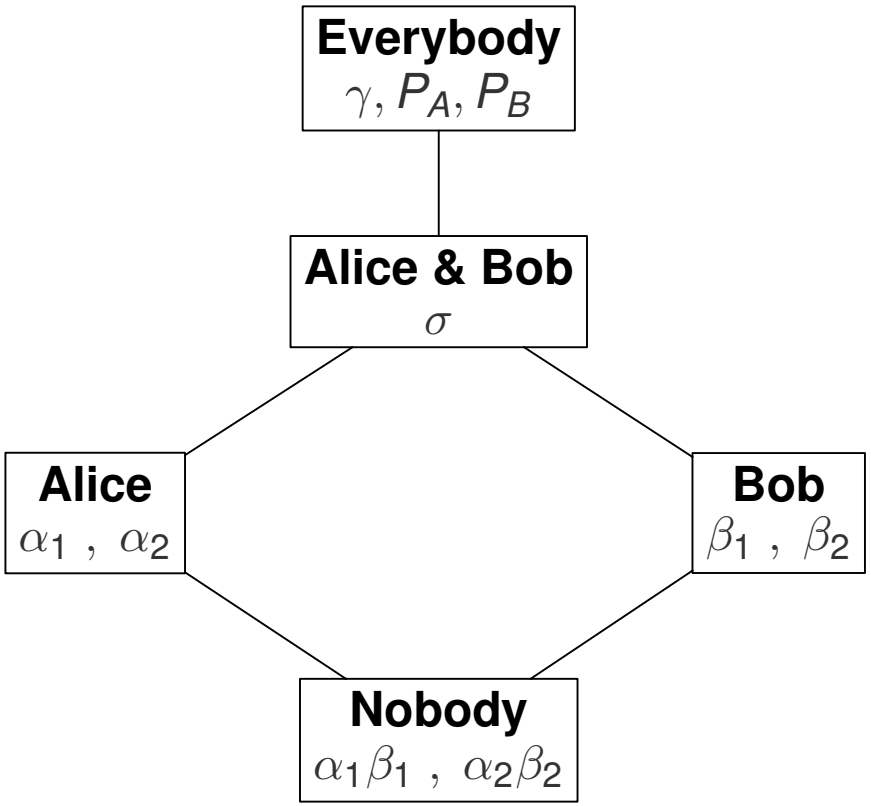
The algebraic data:

Alice	**Public**	**Bob**
	Public root γ	
Selects private $\alpha_1, \alpha_2 \in A$		Selects private $\beta_1, \beta_2 \in B$
Sends $\alpha_1 \gamma \alpha_2$	$P_A \longrightarrow$	
	$P_B \longleftarrow$	Sends $\beta_1 \gamma \beta_2$
Computes: $\alpha_1 P_B \alpha_2$	*By commutativity, these are equal.*	Computes: $\beta_1 P_A \beta_2$

Knowns and unknowns in CAKE

The participants: { Alice, Bob, Eve }.

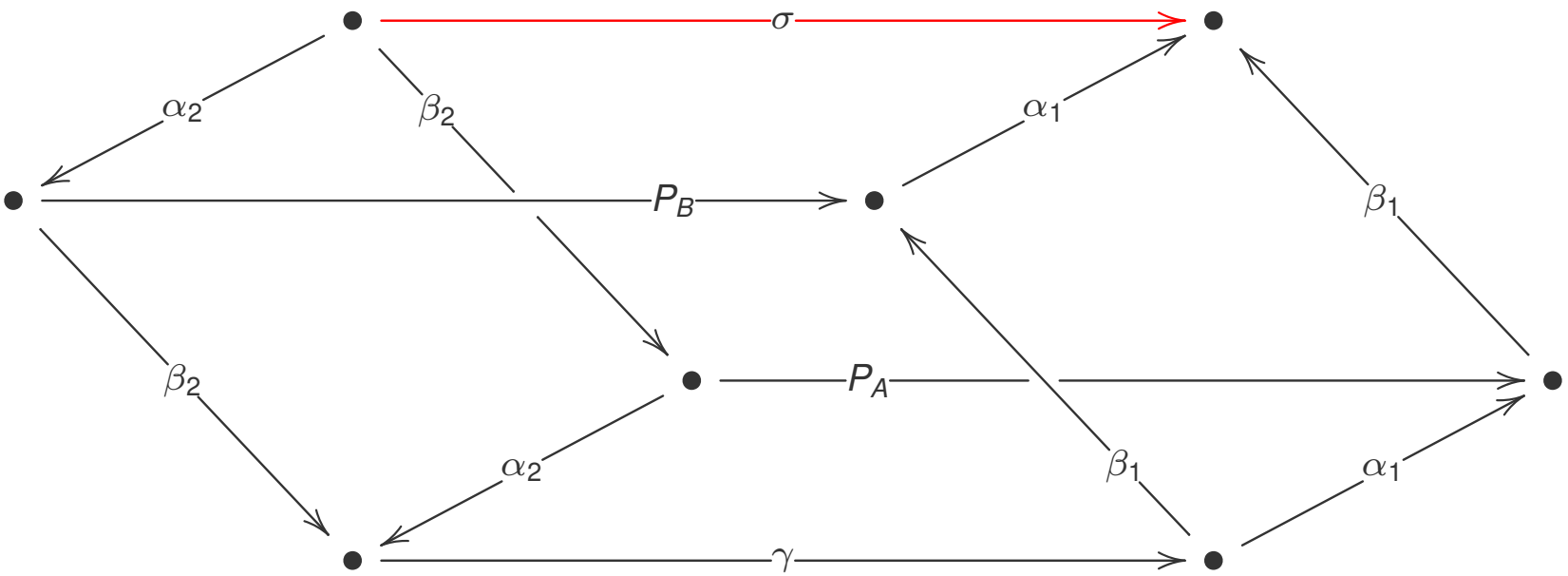
The epistemic data:



CAKE as a commuting diagram

The required arrows are:

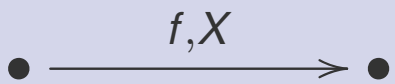
1. The root γ
2. Alice & Bob's private keys, (α_1, α_2) and (β_1, β_2)
3. Alice & Bob's public announcements, P_A and P_B
4. Their shared secret σ



Combining algebraic & epistemic data

Introducing epistemic data to diagrams

- Form the powerset-lattice of participants.
- Label each edge in the diagram by an element of this lattice:

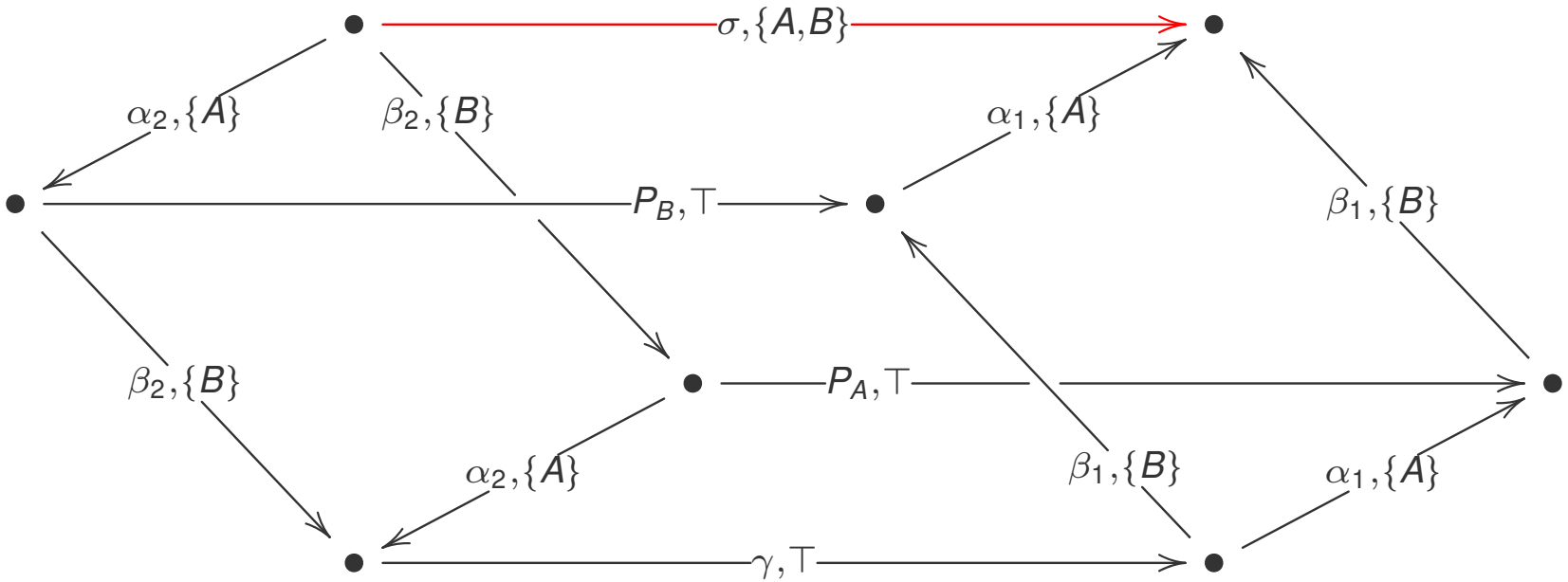


$X \in 2^{\{Alice, Bob, Eve\}}$ consists of participants who

- know the value of f, or (more accurately)
- are able to perform the operation f.

CAKE, in summary

The **Algebraic-Epistemic diagram** for semigroup-CAKE:



Commuting diagrams??

Treating $2^{\{A,B,E\}}$ as a $\wedge$-monoid:

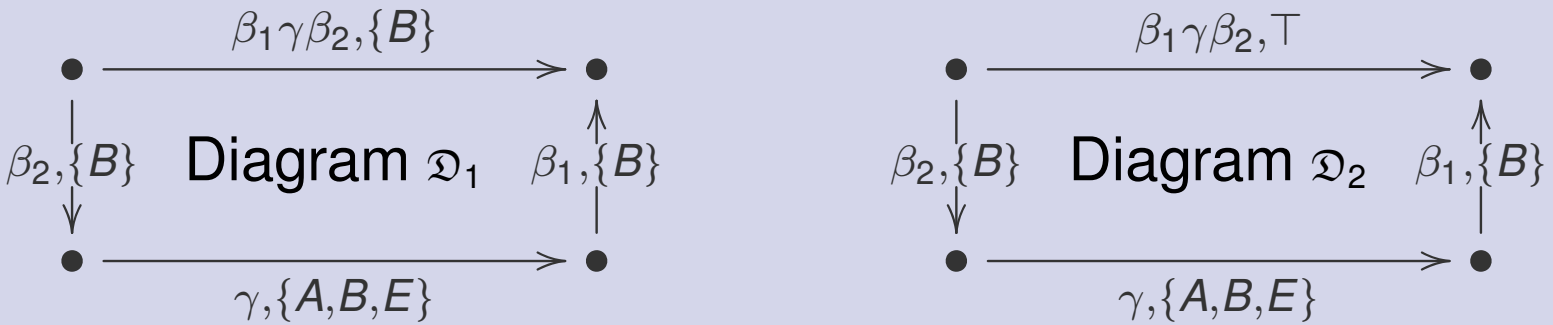
Question: Is this diagram for CAKE a commuting diagram over the product category $\mathcal{M} \times 2^{\{A,B,E\}}$?

Answer: No!

Turning a bug into a feature: *The reasons why / points at which it fails to commute are highly significant.*

Failure of commutativity & public announcements

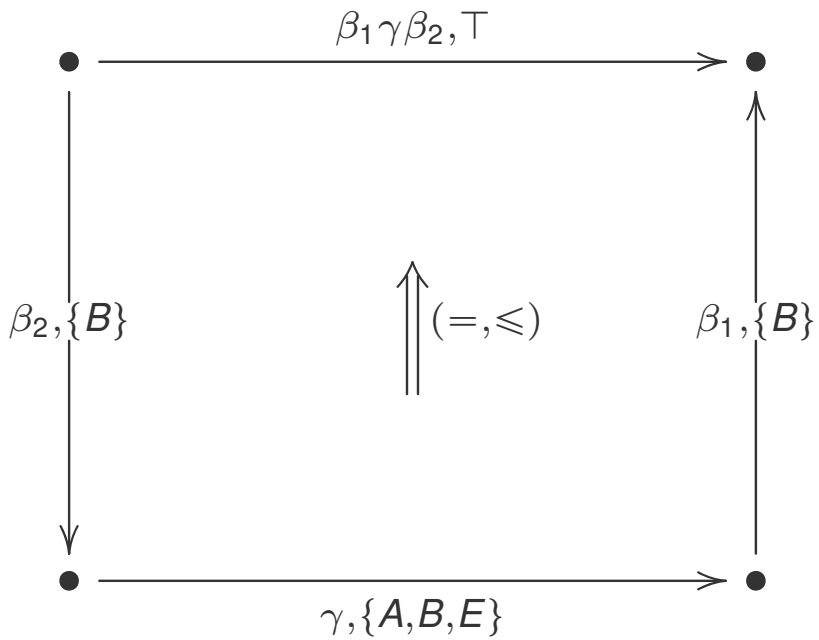
Diagram 1 commutes, Diagram 2 is a slice of CAKE.



1. In **diagram 1**, Bob computes $\beta_2\gamma\beta_1$, and *keeps quiet*.
2. In **diagram 2**, Bob computes $\beta_2\gamma\beta_1$, and *tells the whole world the result*.

Public announcements as 2-categorical data

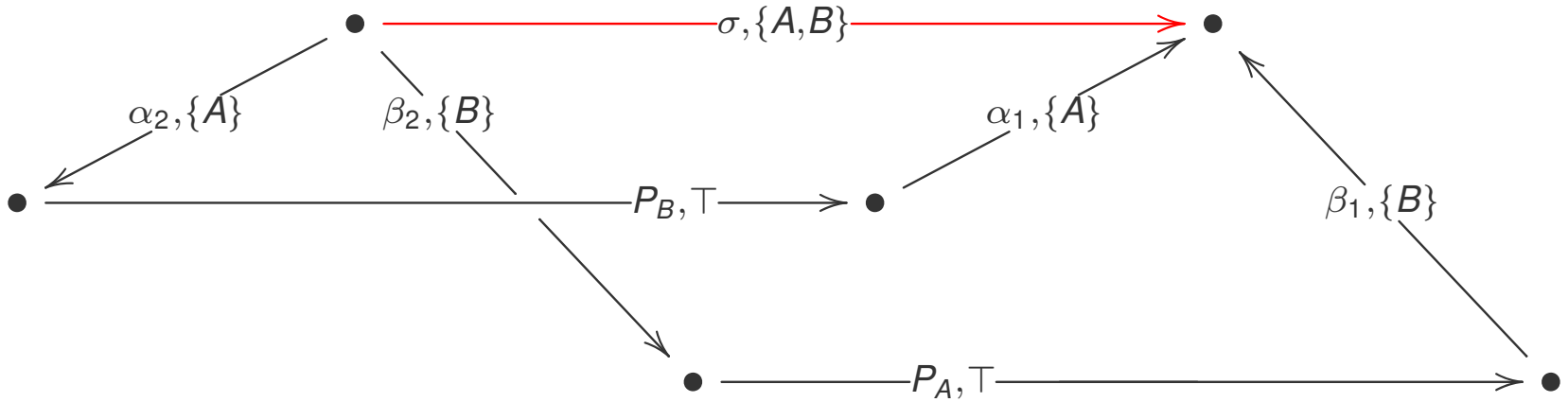
Announcements are (non-identity) 2-cells:



but not all such 2-cells are announcements!

Non-trivial two-cells without public announcements

Another slice of CAKE :



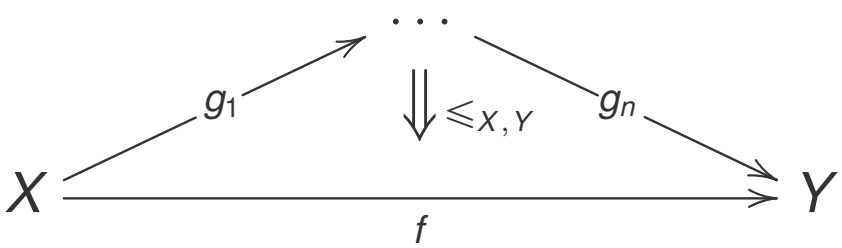
We have non-identity 2-cells, but no announcements.

Here, non-trivial 2-cells correspond to *Alice and Bob's distinct routes to calculating the shared secret*.

A simple definition ...

A diagram $\mathfrak{D}$ over a **Poset** enriched category satisfies the **edge-path condition (EPC)** when:

- Given **an edge and a path** between the nodes X and Y, we have the following 2-cell:



Interpreting the edge-path condition

Motivation: We claim this as a generic 'correctness criterion' for protocols:

In existing protocols ...

We always find this to be the case.

If it fails, then either:

1. We have failed to account for the results of some announcement,
2. We have missed some alternative route to calculating a secret value,
3. There is the possibility of *deadlock*.

The Edge-Path condition & protocols

The E-P condition is defined for arbitrary **Poset**-enriched categories.

We use diagrams over a product category $\mathcal{C} \times L$.

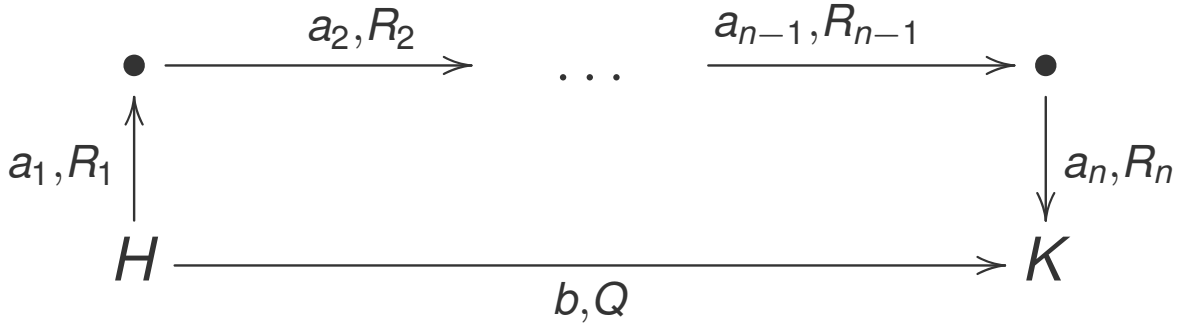
- $\mathcal{C}$ models the algebraic structure, and is enriched over the discrete partial order.
- $\mathcal{L}$ models the participants / epistemic data, and has 'more interesting' poset-enrichment.

For this talk, we simply need $\mathcal{L}$ to be a lattice
(usually the powerset-lattice of participants).

Even for current protocols, we need $\mathcal{C}$ to be a category, not just a monoid.

The edge-path condition: who knows what?

Consider a fragment of the A-E diagram for some protocol:



The edge-path condition states that

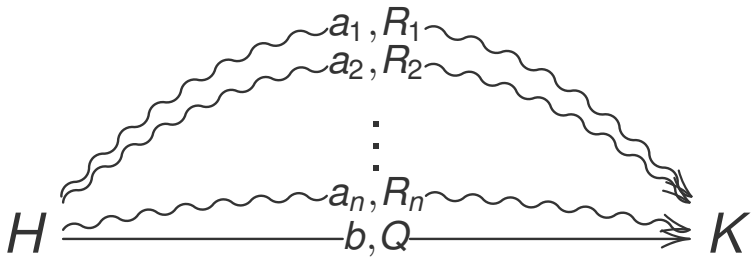
$$b = a_n \ldots a_1 \quad \text{and} \quad \bigwedge_{j=1}^{n} R_j \leqslant Q$$

In terms of powerset-lattices

Any participant $x \in \bigwedge_{j=1}^{n} R_j$ who knows (is able to perform) each operation $\{a_j\}_{j=1..n}$ certainly knows (is able to perform) the composite $r_n \ldots r_1$.

No participant left behind

Consider a fragment of an A-E diagram for some protocol with a **single edge** and **multiple paths** from node H to node K.



The edge-path condition states that

$$b = a_1 = \ldots = a_n \text{ and } R_j \leqslant Q \; \forall \; j = 1..n$$

In terms of powerset-lattices

The members of $R_1, R_2, \ldots, R_n$ are all able to calculate (perform) b, albeit in different ways. Therefore, the subset of participants who can perform b contains each R_j.

A worked example:

Two different approaches to

Diffie-Hellman key exchange

between three participants

The usual story ...

Three participants $\{Alice, Bob, Carol\}$ will come to share a secret.

Start with a (public) prime p and **root** $g \in \mathbb{Z}_p$.

- *Alice*, *Bob*, and *Carol* have private keys $a, b, c \in \mathbb{Z}_p$.
- They will construct the shared secret $g^{abc} = g^{bca} = g^{cab}$.
- All three of them are required, to construct this.
- The usual evesdropper *Eve* can see all communication.

Tripartite Diffie-Hellman, Round I

Based on the **public root** g, and their **private keys** a, b, c,

1. Alice computes g^a and announces the result to Bob.
2. Bob computes g^b and announces the result to Carol.
3. Carol computes g^c and announces the result to Alice.

Tripartite Diffie-Hellman, Round II

Based on the messages they receive,

1. Alice computes $(g^c)^a = g^{ca}$ and announces the result to Bob.
2. Bob computes $(g^a)^b = g^{ab}$ and announces the result to Carol.
3. Carol computes $(g^b)^c = g^{bc}$ and announces the result to Alice.

Tripartite Diffie-Hellman, Round III

They are now able to compute the shared secret.

1. Alice computes $\left(g^{bc}\right)^{a} = g^{abc}$.
2. Bob computes $\left(g^{ca}\right)^{b} = g^{abc}$
3. Carol computes $\left(g^{ab}\right)^{c} = g^{abc}$.

The underlying category

The action takes place in a small subcategory of **Set**:

- **Objects:** $\mathbb{Z}_p$ and $\{*\}$
- **Arrows:**
  1. *modular exponentiation* $(\ )^x : \mathbb{Z}_p \to \mathbb{Z}_p$, for all $x = 0 \ldots p-1$
  2. *selecting an element* $[x] : \{\star\} \to \mathbb{Z}_p$, where $[x](\star) = x \in \mathbb{Z}_p$

A deliberate choice

We have not to included discrete logarithms as arrows of this category.

The underlying category

The action takes place in a small subcategory of **Set**:

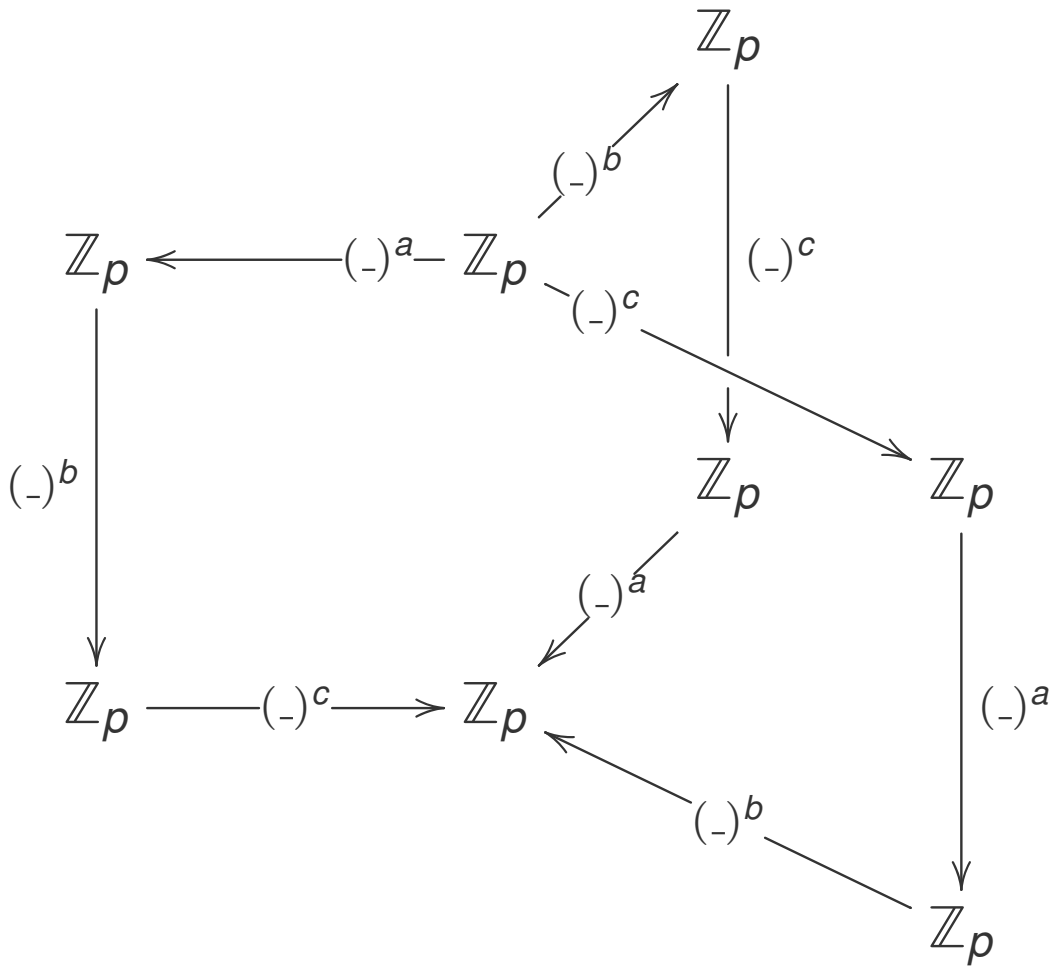
- **Objects:** $\mathbb{Z}_p$ and $\{*\}$
- **Arrows:**
  1. *modular exponentiation* $(\ )^x : \mathbb{Z}_p \to \mathbb{Z}_p$, for all $x = 0 \ldots p-1$
  2. *selecting an element* $[x] : \{\star\} \to \mathbb{Z}_p$, where $[x](\star) = x \in \mathbb{Z}_p$

A deliberate choice

We have not to included discrete logarithms as arrows of this category.

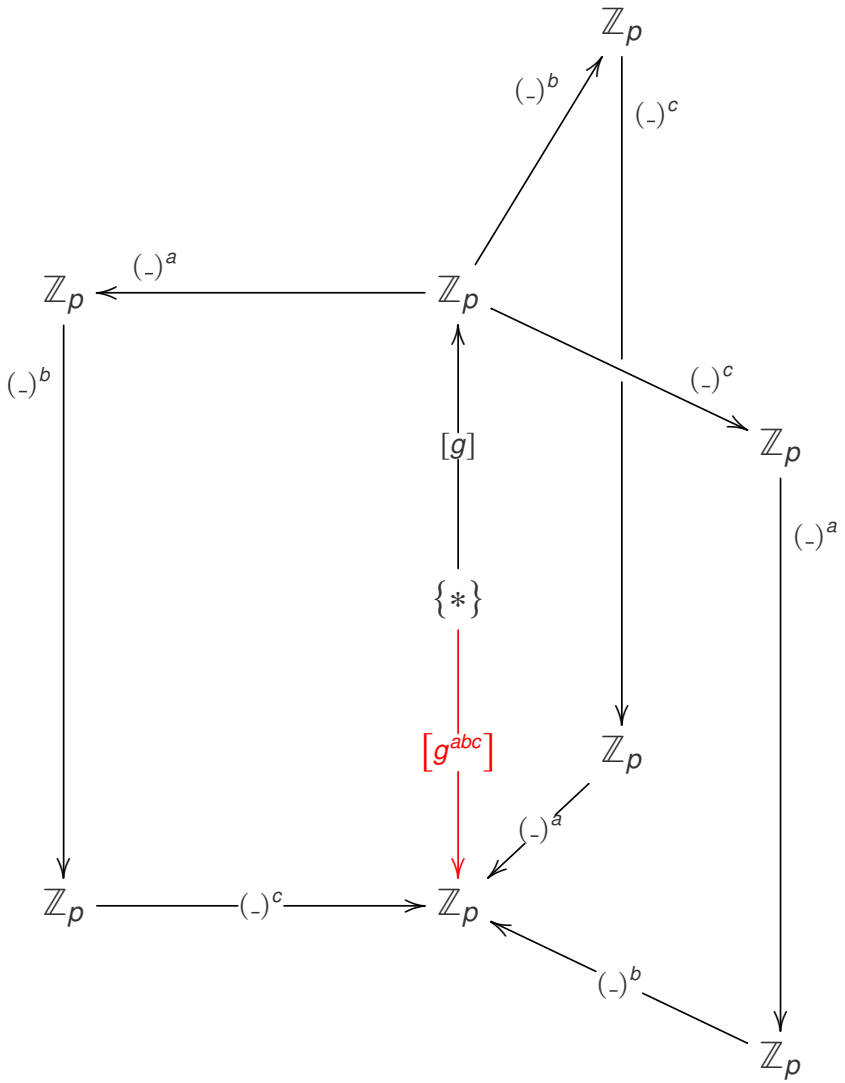
The core identity

The basic identity is $(((\_)^a)^b)^c = (((\_)^b)^c)^a = (((\_)^c)^a)^b$



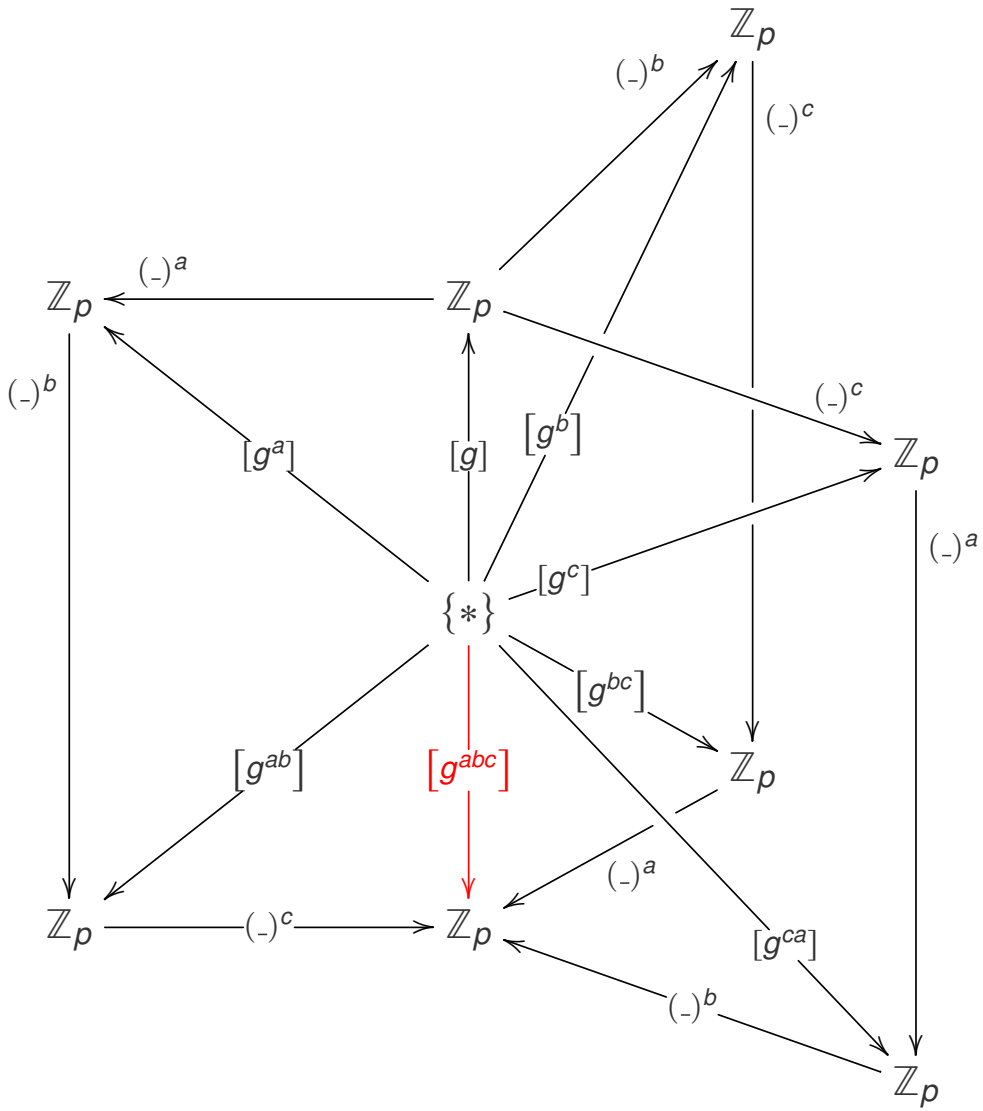
Adding in the root element

We require these equalities *applied to the root* $g \in \mathbb{Z}$.



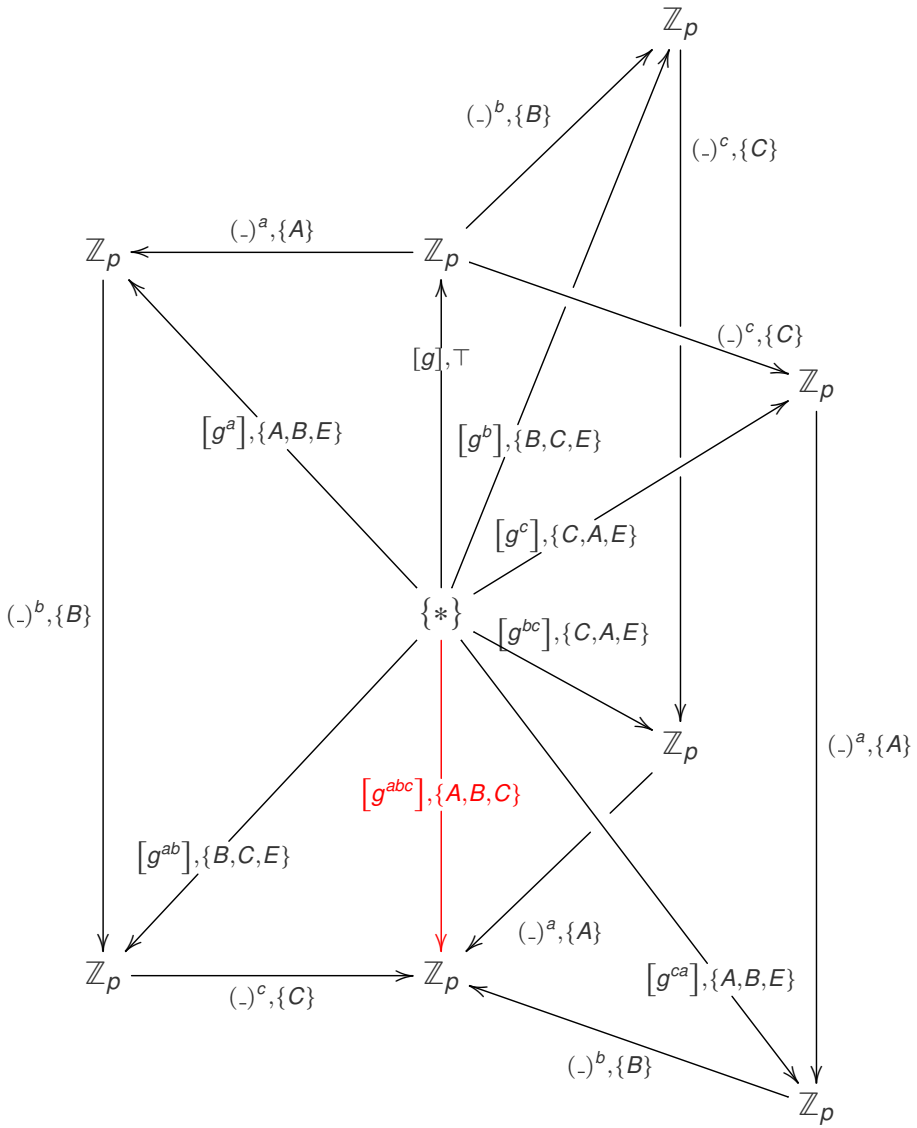
What announcements are made?

The elements $g^a, g^b, g^c, g^{ab}, g^{bc}, g^{ca}$ are all announced:



Who knows what?

Adding in the epistemic data:



Subdiagrams of Algebraic-Epistemic diagrams

An obvious step

Consider subdiagrams consisting of:

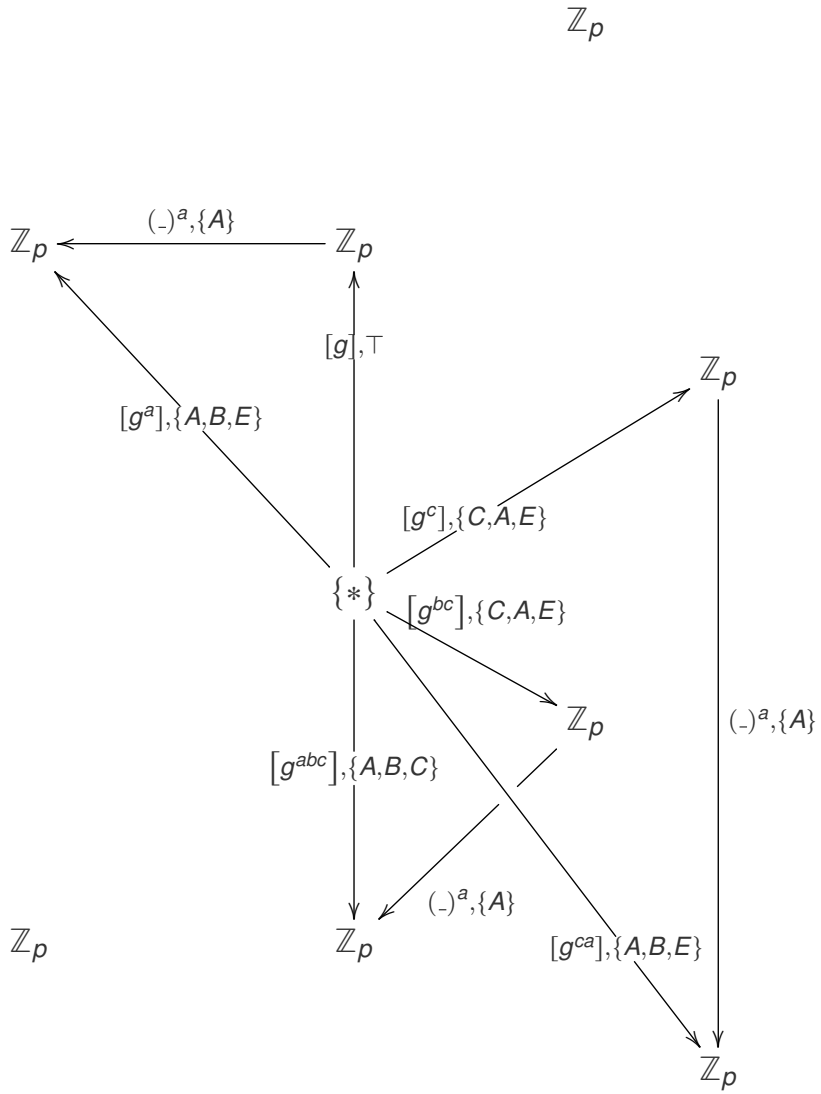
All edges whose lattice label is $\geqslant$ some element $X \in 2^P$.

These subdiagrams also satisfy the Edge-Path condition.

They correspond to *different participants views of the protocol*.

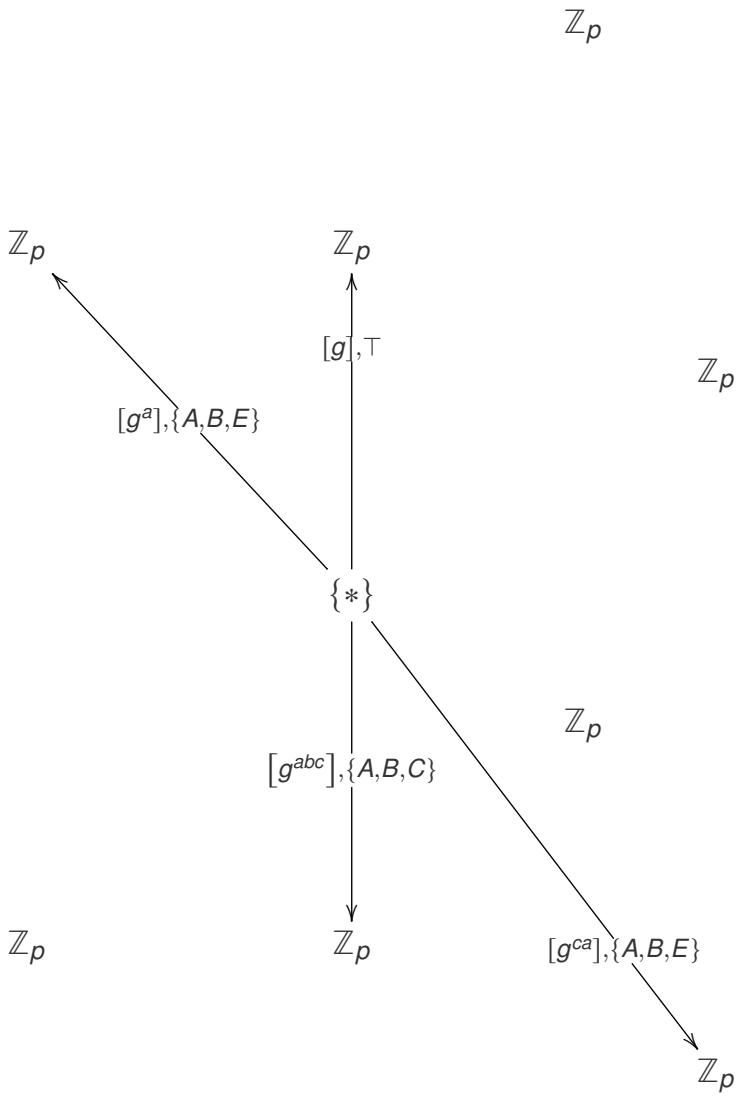
Alice's perspective on the protocol:

All edges with label $\geqslant \{A\}$



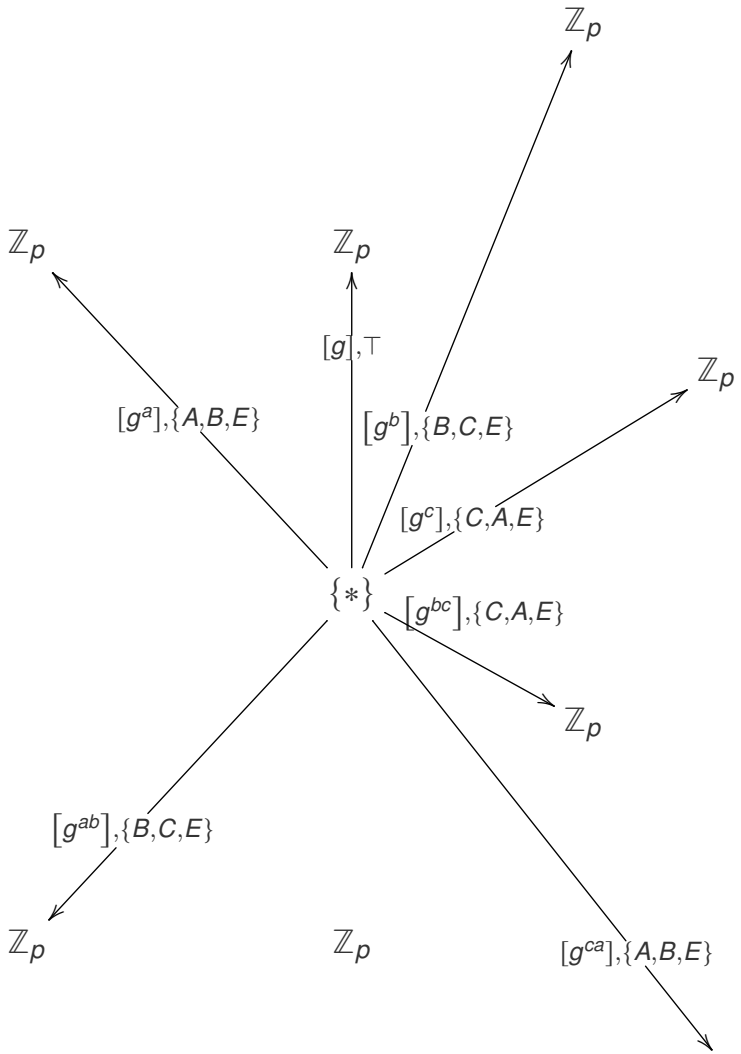
What Alice and Bob both see:

All edges with label $\geqslant \{A, B\}$



What the evesdropper knows!

All edges with label $\geqslant \{E\}$.



Does this help??

Simple diagram-chasing makes it easy to answer some questions:

Question Can we vary the order of computations / announcements?

Answer Yes, quite a bit!

Question Does it matter if any of the participants (apart from Eve) are evesdropping?

Answer No, not at all!

Question What does Eve need to know, to find the shared secret?

Answer *Any of the private keys will do*!

We can also ***compare approaches*** to the same problem.

Another approach ...

How else may *Alice*, *Bob*, and *Carol* communicate privately?

As before, assume:

- Prime p,
- Public Root $g \in \mathbb{Z}_p$
- Private keys $a, b, c \in \mathbb{Z}_p$

Every pair will compute a *distinct* shared secret.

$$\textit{Alice} - - \textit{Bob} \quad \textit{Bob} - - \textit{Carol} \quad \textit{Carol} - - \textit{Alice}$$

Pairwise three-party Diffie-Hellman

- *Alice*, *Bob*, and *Carol* compute

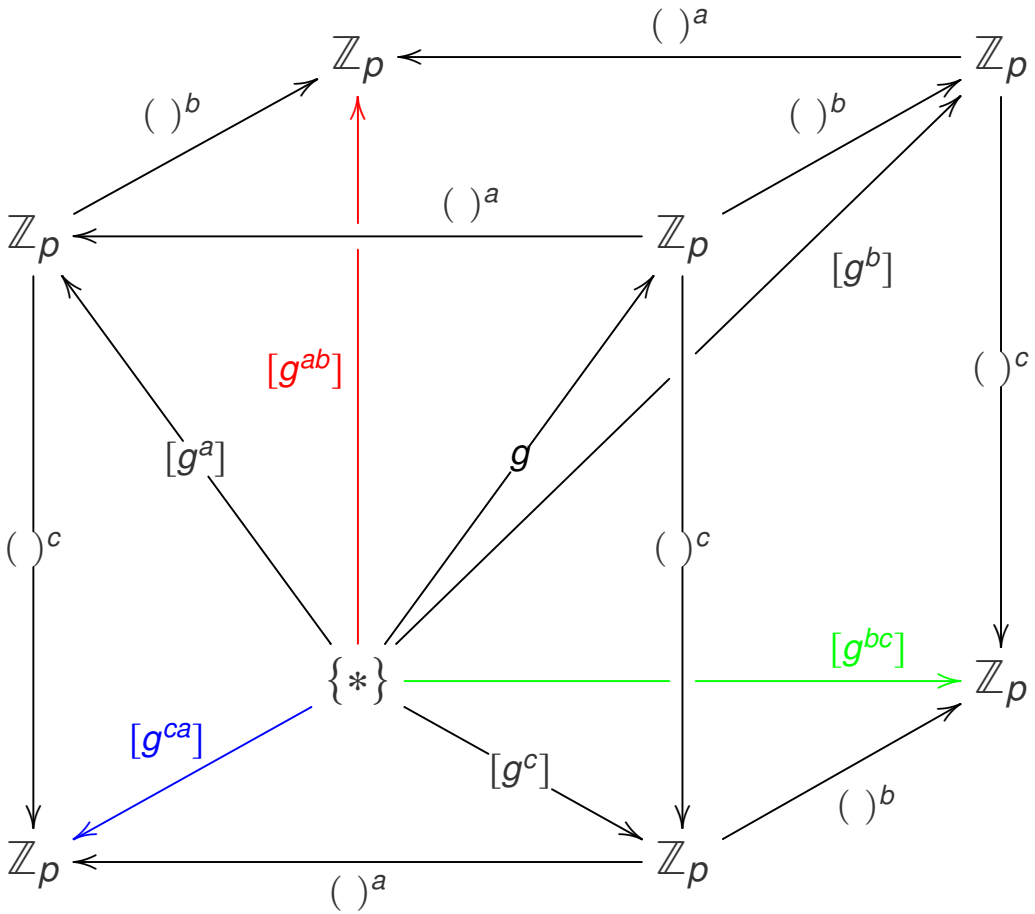
$$g^a \quad \text{and} \quad g^b \quad \text{and} \quad g^c$$

  respectively. They publicly announce their results.
- They each compute a *pair* of shared secrets:

Alice computes g^{ba} and g^{ca}
Bob computes g^{cb} and g^{ab}
Carol computes g^{ac} and g^{bc}

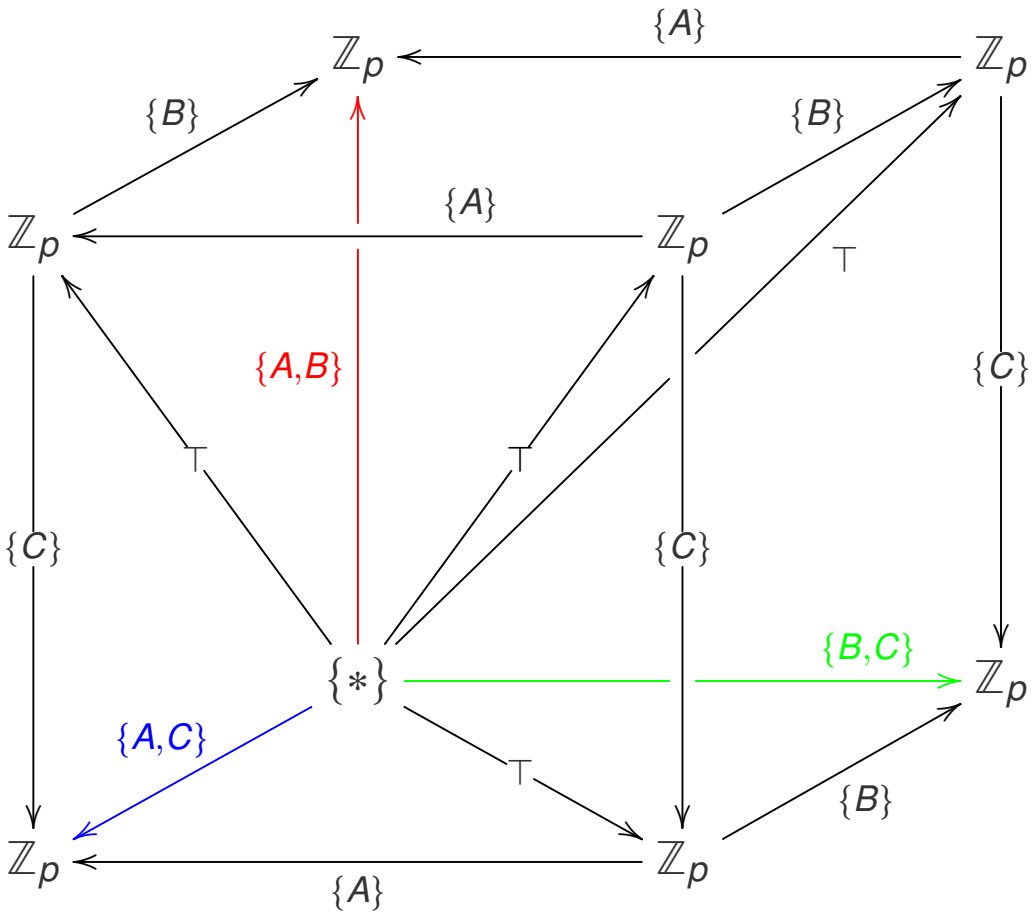
A-E diagram for 3-way secret sharing

The (commuting) algebraic labelling:



A-E diagram for 3-way secret sharing

The (EPC satisfying) lattice labelling:



Comparing this approach ...

Again, by simple diagram-chasing:

Question Can any additional information be announced?

Answer No, not without compromising the protocol!

Question What happens if Eve discovers (say) Bob's secret key?

Answer She can discover two out of the three shared secrets.

Question Is this the same as tripartite Diffie-Hellman?

Answer *No, definitely not!*

Can we go further??

Drawing diagrams gives a *visual representation* of algebraic relationships, epistemic knowledge, and information flow.

1. The category theory used hase been very basic.
2. The difficulty of solving algebraic problems has been treated as a '**black box**'.

Is it too much to hope these points are related ??

Back to CAKE

Recall the CAKE protocol

This is a *general recipe* for producing public key protocols.

The key ingredient for security is the *choice of monoid*.

What structures have been proposed, and why?

An interesting first choice ...

CAKE was first proposed in:

Combinatorial group theory and public key cryptography (2004)

General proposals for cryptosystems based on algebraic structures.

The first concrete protocol was given in:

Thompson's group $\mathcal{F}$ and Public Key Cryptography (2004)

"This group has several properties that make it **particularly fit for cryptographic purposes**."

This is an ex-protocol.

- **F. Matucci** (2006)

  The Shpilrain-Ushakov Protocol for Thompson's Group F is always breakable

- **Ruinskiy, Shamir, Tsaban** (2007)

  Length-Based Cryptanalysis: the case of Thompson's group

  Conjecture: *" no public key cryptosystem based on the difficulty of solving an equation in this group can be secure."*

- **Hines** (2013)

  Modular arithmetic identities from categorical coherence

  (Implicitly) *A large collection of representations of Thompson's $\mathcal{F}$ as modular arithmetic functions.*

Thompson's group $\mathcal{F}$ and associativity

- **R. McKenzie, R. Thompson** (1971): Close connection between Thompson's group $\mathcal{F}$, and associativity laws
- **K. Brown** (2004) A group homomorphism $\_ \star \_ : \mathcal{F} \times \mathcal{F} \to \mathcal{F}$ that is *associative up to isomorphism*.
- **M. V. Lawson** (2004) The canonical associativity isomorphisms for a class of single-object tensors is precisely $\mathcal{F}$.
- **P. Dehornoy** (2005) 'The only [non-trivial] relations in this presentation of $\mathcal{F}$ correspond to the well-known MacLane-Stasheff pentagon.'
- **M. Brinn** (2005) 'the resemblance of the usual coherence theorems with Thompson's group $\mathcal{F}$'.
- **M. Fiore, T. Leinster** (2010) Thompson's group $\mathcal{F}$ is the symmetry group of an idempotent U in the free strict monoidal category generated by U.

A relevant cohence theorem:

Coherence and Strictification for Self-Similarity
Journal of Homotopy & Related Structures (Hines 2016)

A semi-monoidal equivalence of monogenic categories	
Self-similarity $S \cong S \otimes S$ up to isomorphism (a.k.a. idempotency)	Strict self-similarity $S = S \star S$ (a.k.a. being a monoid)

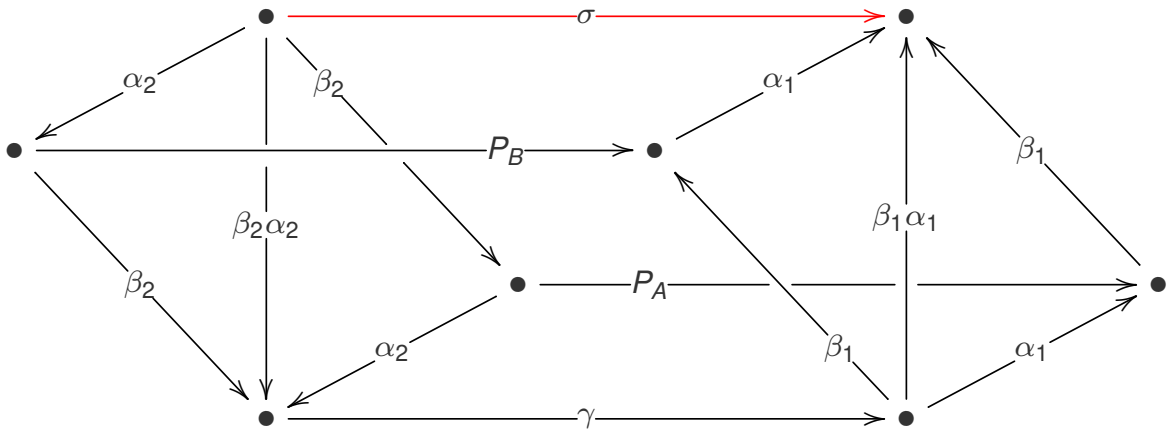
As a consequence:

1. No monoid can have a strictly associative tensor.
2. For any monoid with a (semi-) monoidal tensor, either:
   1. The group of associativity isomorphisms is precisely $\mathcal{F}$.
   2. The unique object is the unit object.

Cryptographic protocols as canonical diagrams

Based on these: Thompson's group $\mathcal{F}$ is the group of canonical associativity isomorphisms for a tensor.

Diagrams for the Shpilrain-Ushakov protocol
are **commuting canonical diagrams**
in the sense of MacLane's coherence theorem.



Simple exercises!!

Within a (semi-) monoidal category:

1. Given a canonical diagram, how easy is it to decide whether it commutes?
2. Given a commuting canonical diagram, how easy is it to fill in missing edges?
   – can this be done uniquely?
3. How can we find sets A, B of canonical isomorphisms that point-wise commute?

The mother of all coincidences?

A personal viewpoint ...

Shpilrain & Ushakov (2004) gave *motivation* for choosing Thompson's group $\mathcal{F}$.

These properties seem inseparable from the *categorical interpretation*.

Do we see similar elsewhere?

Some other places to look ...

- Proposed use of Thompson's group $\mathcal{V}$

  – the coherence isomorphisms for a symmetric tensor on a monoid.
  M. Fiore, M. Campos (2013)

- Proposed use of poylcyclic monoids / groups.

  – related to coherence isomorphisms for tensors on monoids with projections / injections.
  Hines, Lawson (1998,1999)

- Shor's quantum algorithm for factoring.

  – related to Laplaza's theory of coherence for distributivity
  Hines (2013)

- Other proposed algebraic structures (!)

  –T.B.C.