Solving the *Tensor Isomorphism Problem* for special orbits with low rank points

Valerie Gilchrist, Laurane Marco, Christophe Petit, Gang Tang

Published at Crypto 2024

Commitment schemes

Commitment schemes

Suppose we are playing a game, and want to choose who will go first

Commitment schemes

Suppose we are playing a game, and want to choose who will go first

Coin flip!

Commitment schemes

Suppose we are playing a game, and want to choose who will go first

Coin flip!

Does it work over the phone?

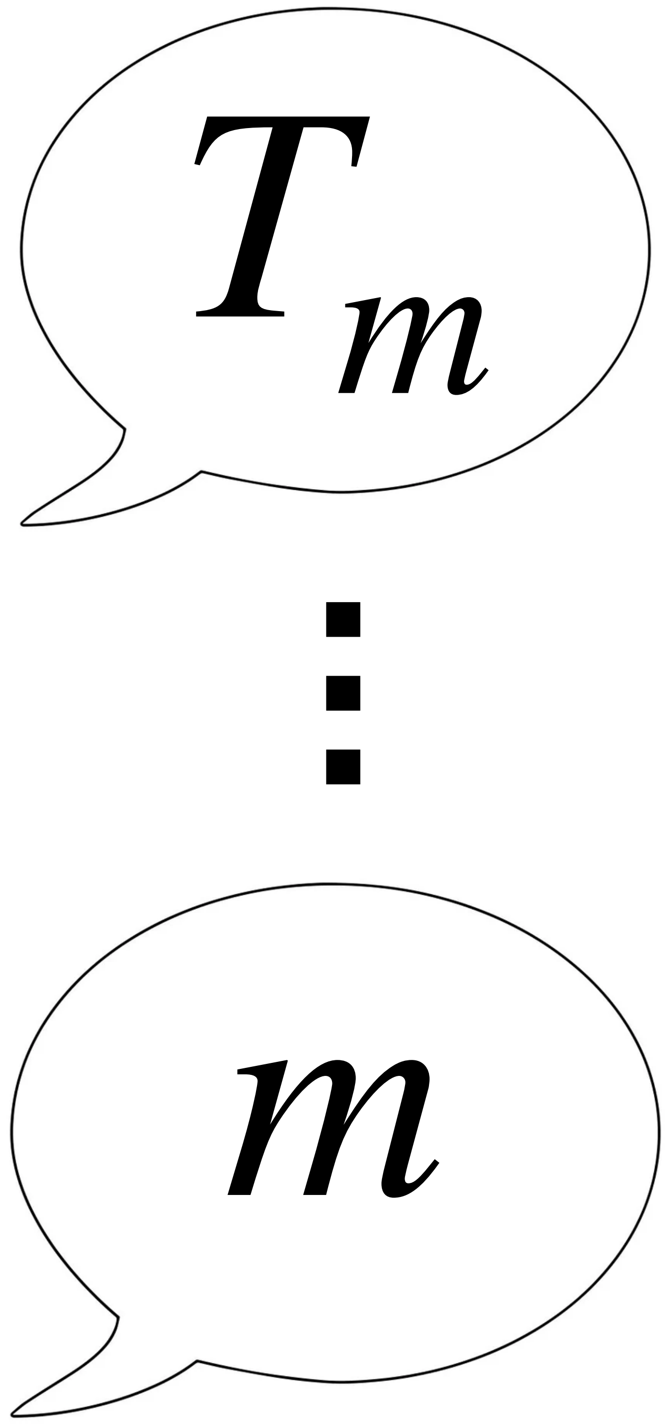
Commitment schemes

Commitment schemes

In a *commitment scheme* a sender wants to commit to some value m.

Commitment schemes

In a *commitment scheme* a sender wants to commit to some value m.



the sender publishes a commitment

*d*epending on m

Later the sender releases m,

and a verifier can check that T_m

was created using m

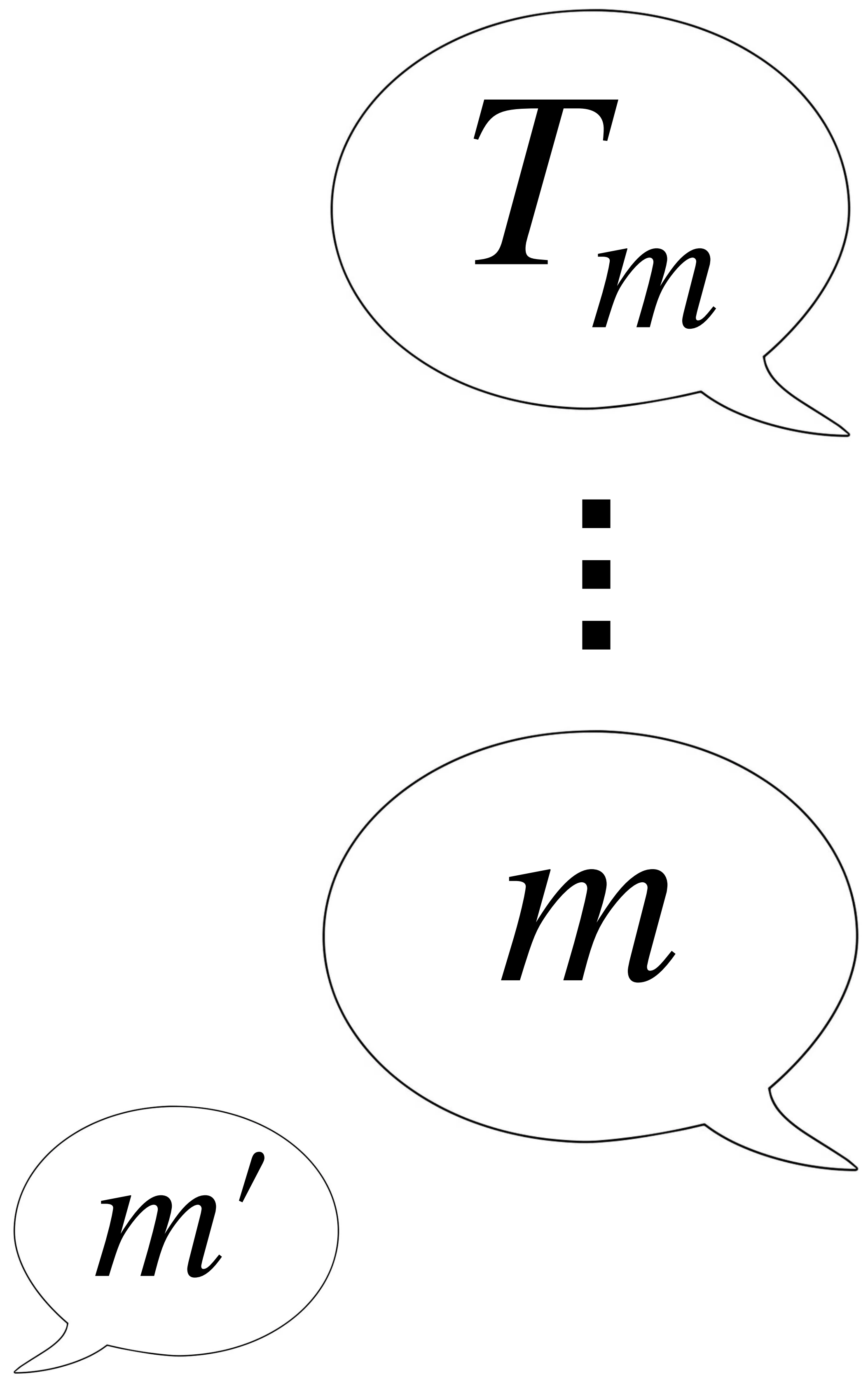
Commitment schemes

The scheme should be ***hiding:***

- the commitment T should leak no information about m

The scheme should be ***binding:***

- no other value $m' \neq m$ should be able to open T



Tensors

Tensors

Tensor product is an operation on vectors

Tensors

Tensor product is an operation on vectors

We would like to compute the tensor $\mathbf{u} \otimes \mathbf{v} \otimes \mathbf{w}$ where $\mathbf{u} = [u_1, u_2], \mathbf{v} = [v_1, v_2], \mathbf{w} = [w_1, w_2]$.

Tensors

Tensor product is an operation on vectors

We would like to compute the tensor $\mathbf{u} \otimes \mathbf{v} \otimes \mathbf{w}$ where $\mathbf{u} = [u_1, u_2], \mathbf{v} = [v_1, v_2], \mathbf{w} = [w_1, w_2]$.

We proceed by first expanding the matrix $\mathbf{v} \cdot \mathbf{w}^T$:

$$\mathbf{v} \cdot \mathbf{w}^T = \begin{bmatrix} v_1 w_1 & v_1 w_2 \\ v_2 w_1 & v_2 w_2 \end{bmatrix}$$

Tensors

Tensor product is an operation on vectors

We would like to compute the tensor $\mathbf{u} \otimes \mathbf{v} \otimes \mathbf{w}$ where $\mathbf{u} = [u_1, u_2], \mathbf{v} = [v_1, v_2], \mathbf{w} = [w_1, w_2]$.

We proceed by first expanding the matrix $\mathbf{v} \cdot \mathbf{w}^T$:

$$\mathbf{v} \cdot \mathbf{w}^T = \begin{bmatrix} v_1 w_1 & v_1 w_2 \\ v_2 w_1 & v_2 w_2 \end{bmatrix}$$

Now we multiply this matrix by each entry of $\mathbf{u}$, storing them in a list as we go:

$$u_1 \begin{bmatrix} v_1 w_1 & v_1 w_2 \\ v_2 w_1 & v_2 w_2 \end{bmatrix}, u_2 \begin{bmatrix} v_1 w_1 & v_1 w_2 \\ v_2 w_1 & v_2 w_2 \end{bmatrix}$$

Tensors

Tensors

Let t be a tensor. We can always write it as

$$t := \sum_{i,j,k} m_{i,j,k}\, e_i \otimes e_j \otimes e_k$$

Tensors

Let t be a tensor. We can always write it as

$$t := \sum_{i,j,k} m_{i,j,k}\, e_i \otimes e_j \otimes e_k$$

e.g.

$$t = \begin{bmatrix} u_1 v_1 w_1 & u_1 v_1 w_2 \\ u_1 v_2 w_1 & u_1 v_2 w_2 \end{bmatrix}, \begin{bmatrix} u_2 v_1 w_1 & u_2 v_1 w_2 \\ u_2 v_2 w_1 & u_2 v_2 w_2 \end{bmatrix}$$

Tensors

Let t be a tensor. We can always write it as

$$t := \sum_{i,j,k} m_{i,j,k}\, e_i \otimes e_j \otimes e_k$$

e.g.

$$t = \begin{bmatrix} u_1v_1w_1 & u_1v_1w_2 \\ u_1v_2w_1 & u_1v_2w_2 \end{bmatrix}, \begin{bmatrix} u_2v_1w_1 & u_2v_1w_2 \\ u_2v_2w_1 & u_2v_2w_2 \end{bmatrix}$$

➡

$$t = u_1v_1w_1 \cdot e_1 \otimes e_1 \otimes e_1 + u_1v_1w_2 \cdot e_1 \otimes e_1 \otimes e_2 \cdots$$

Tensors

Let t be a tensor. We can always write it as

$$t := \sum_{i,j,k} m_{i,j,k}\, e_i \otimes e_j \otimes e_k$$

e.g.

$$t = \begin{bmatrix} u_1v_1w_1 & u_1v_1w_2 \\ u_1v_2w_1 & u_1v_2w_2 \end{bmatrix}, \begin{bmatrix} u_2v_1w_1 & u_2v_1w_2 \\ u_2v_2w_1 & u_2v_2w_2 \end{bmatrix}$$

$$\begin{aligned} t &= u_1v_1w_1 \cdot e_1 \otimes e_1 \otimes e_1 \\ &\quad + u_1v_1w_2 \cdot e_1 \otimes e_1 \otimes e_2 \cdots \\ &= \sum_{i,j,k} u_iv_jw_k \cdot\ e_i \otimes e_j \otimes e_k \end{aligned}$$

Tensors

Tensors

$$t := \sum_{i,j,k} m_{i,j,k}\, e_i \otimes e_j \otimes e_k$$

Tensors

$$t := \sum_{i,j,k} m_{i,j,k}\, e_i \otimes e_j \otimes e_k$$

Let A, B, C be invertible matrices

We can compute the following isomorphism applied to t :

$$(A, B, C) \star t := \sum_{i,j,k} m_{i,j,k}\, Ae_i \otimes Be_j \otimes Ce_k$$

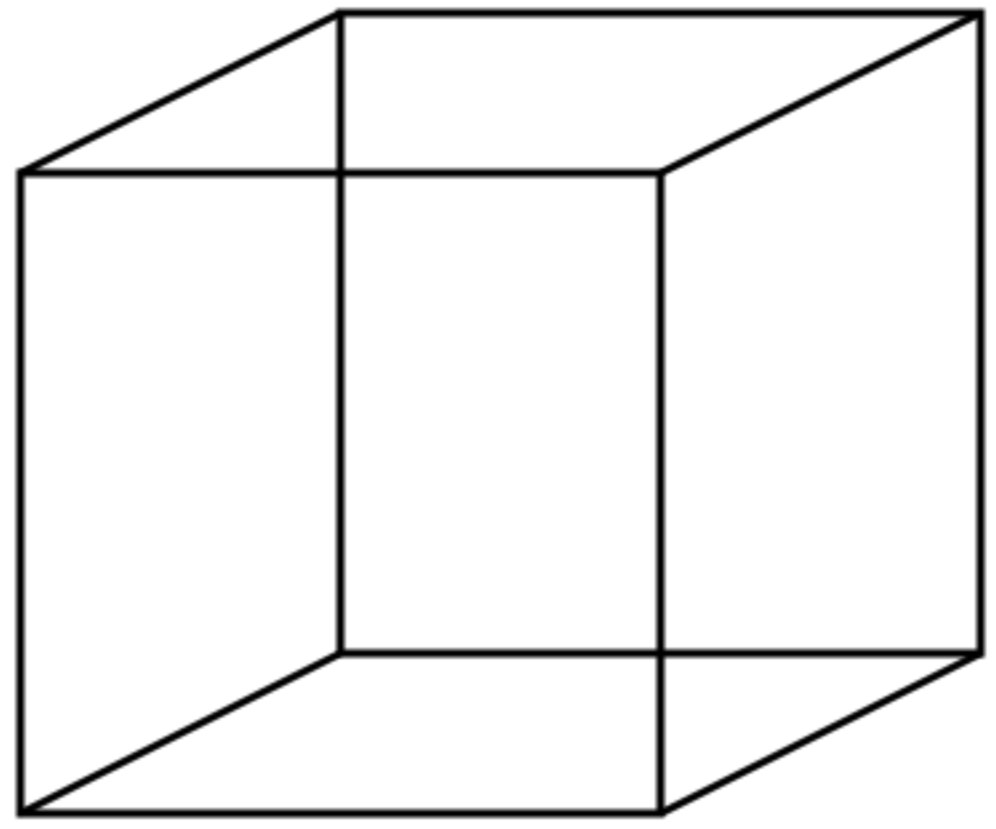
Tensors

$$t := \sum_{i,j,k} m_{i,j,k}\, e_i \otimes e_j \otimes e_k$$

Let A, B, C be invertible matrices

We can compute the following isomorphism applied to t :

$$(A, B, C) \star t := \sum_{i,j,k} m_{i,j,k}\, Ae_i \otimes Be_j \otimes Ce_k$$



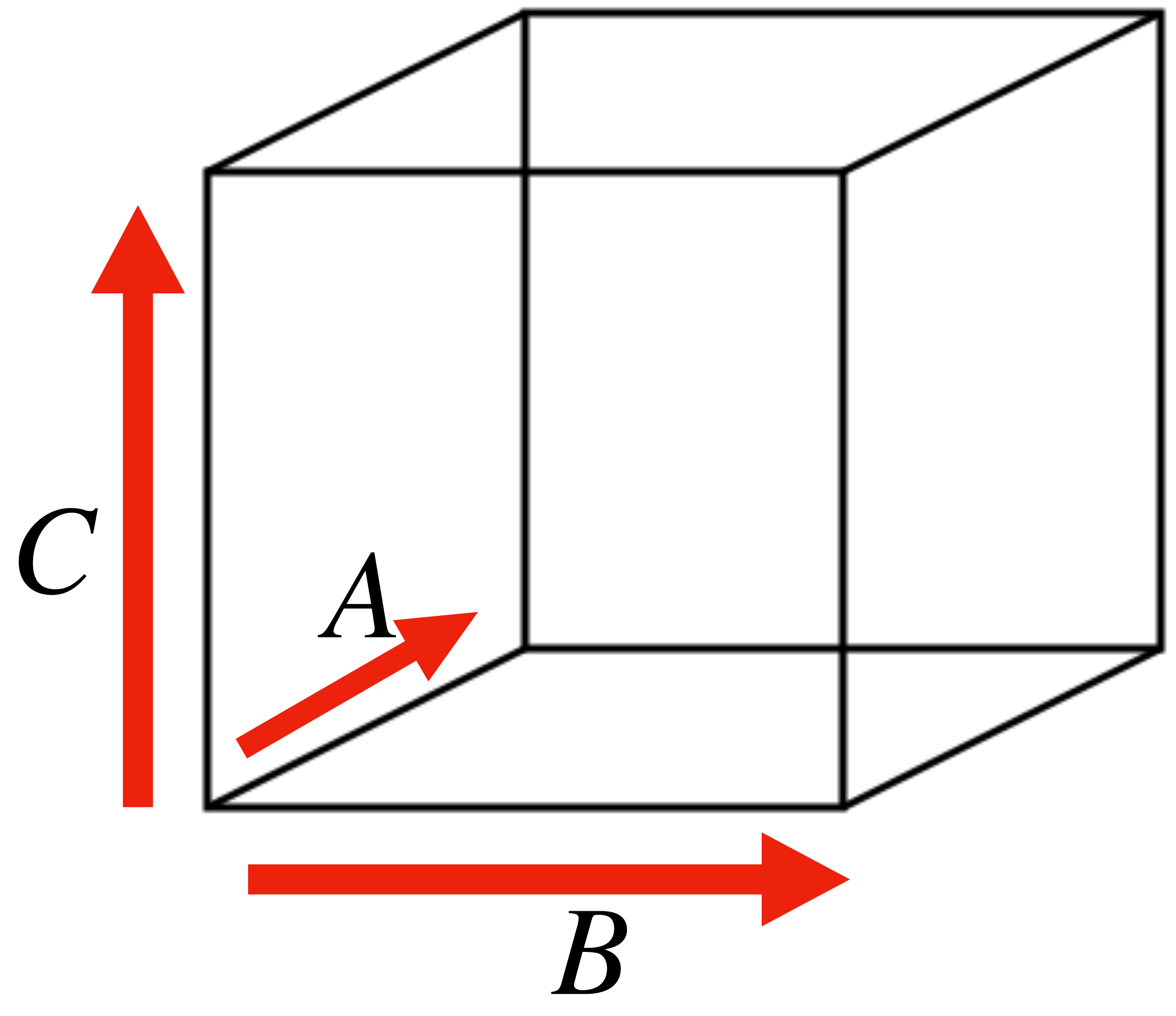
Tensors

$$t := \sum_{i,j,k} m_{i,j,k}\, e_i \otimes e_j \otimes e_k$$

Let A, B, C be invertible matrices

We can compute the following isomorphism applied to t :

$$(A, B, C) \star t := \sum_{i,j,k} m_{i,j,k}\, Ae_i \otimes Be_j \otimes Ce_k$$



Tensors

An example over $\mathbb{F}_7$:

Tensors

An example over $\mathbb{F}_7$:

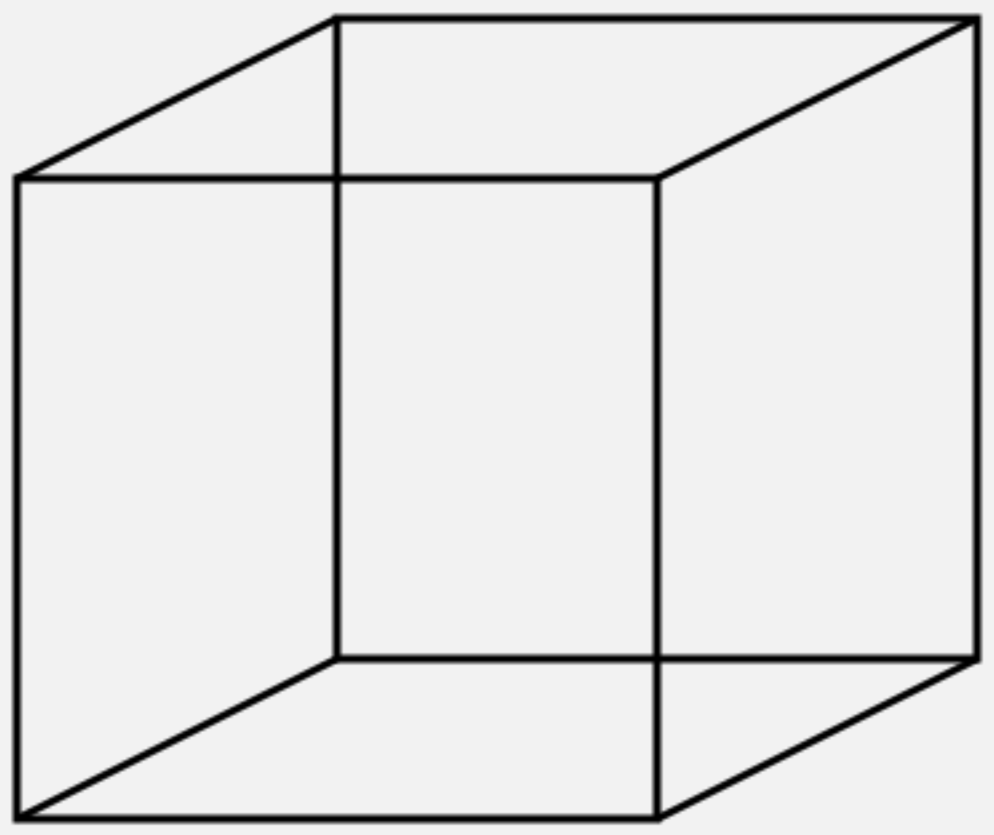
$$t = \sum_{i,j,k} e_i \otimes e_j \otimes e_k = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}, \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}, \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}, \qquad A = \begin{bmatrix} 6 & 3 & 6 \\ 5 & 6 & 3 \\ 4 & 4 & 4 \end{bmatrix}$$

Tensors

An example over $\mathbb{F}_7$:

$$t = \sum_{i,j,k} e_i \otimes e_j \otimes e_k = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}, \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}, \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}, \qquad A = \begin{bmatrix} 6 & 3 & 6 \\ 5 & 6 & 3 \\ 4 & 4 & 4 \end{bmatrix}$$

$$(A, I, I) \star t := \sum_{i,j,k} Ae_i \otimes e_j \otimes e_k = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}, \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 5 & 5 & 5 \\ 5 & 5 & 5 \\ 5 & 5 & 5 \end{bmatrix}$$



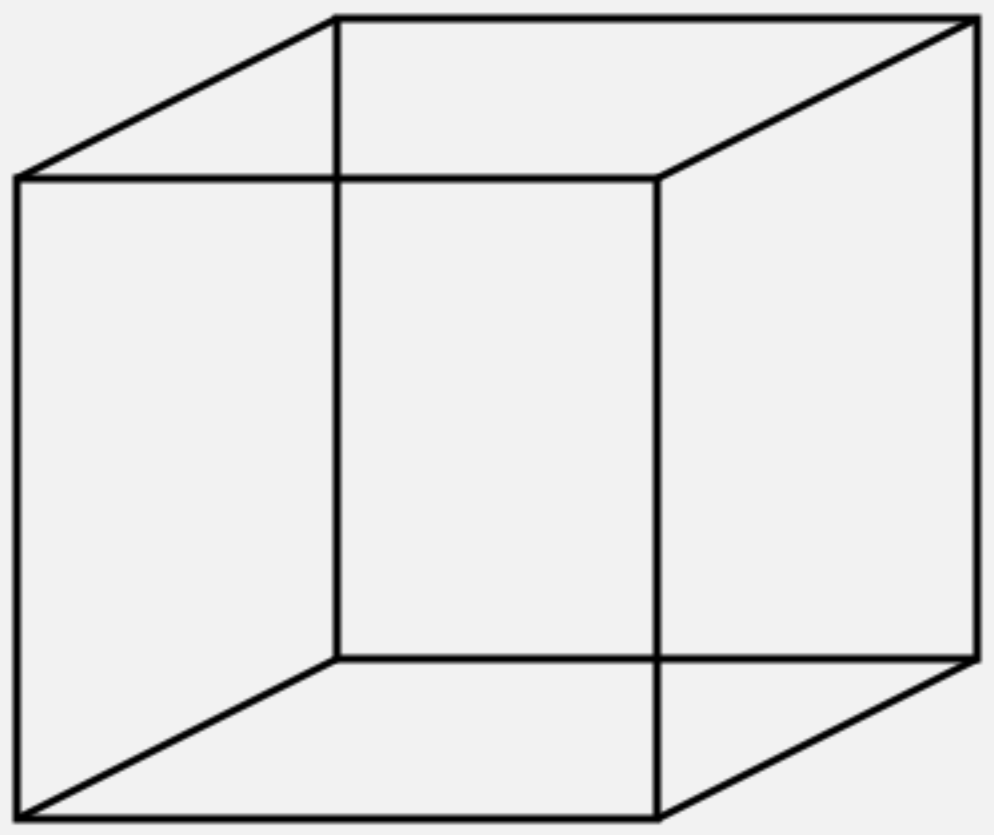
Tensors

An example over $\mathbb{F}_7$:

$$t = \sum_{i,j,k} e_i \otimes e_j \otimes e_k = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}, \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}, \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}, \qquad A = \begin{bmatrix} 6 & 3 & 6 \\ 5 & 6 & 3 \\ 4 & 4 & 4 \end{bmatrix}$$

$$(A, I, I) \star t := \sum_{i,j,k} Ae_i \otimes e_j \otimes e_k = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}, \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 5 & 5 & 5 \\ 5 & 5 & 5 \\ 5 & 5 & 5 \end{bmatrix}$$

$$= \sum_{j,k} Ae_1 \otimes e_j \otimes e_k + \sum_{j,k} Ae_2 \otimes e_j \otimes e_k + \sum_{j,k} Ae_3 \otimes e_j \otimes e_k$$



Tensors

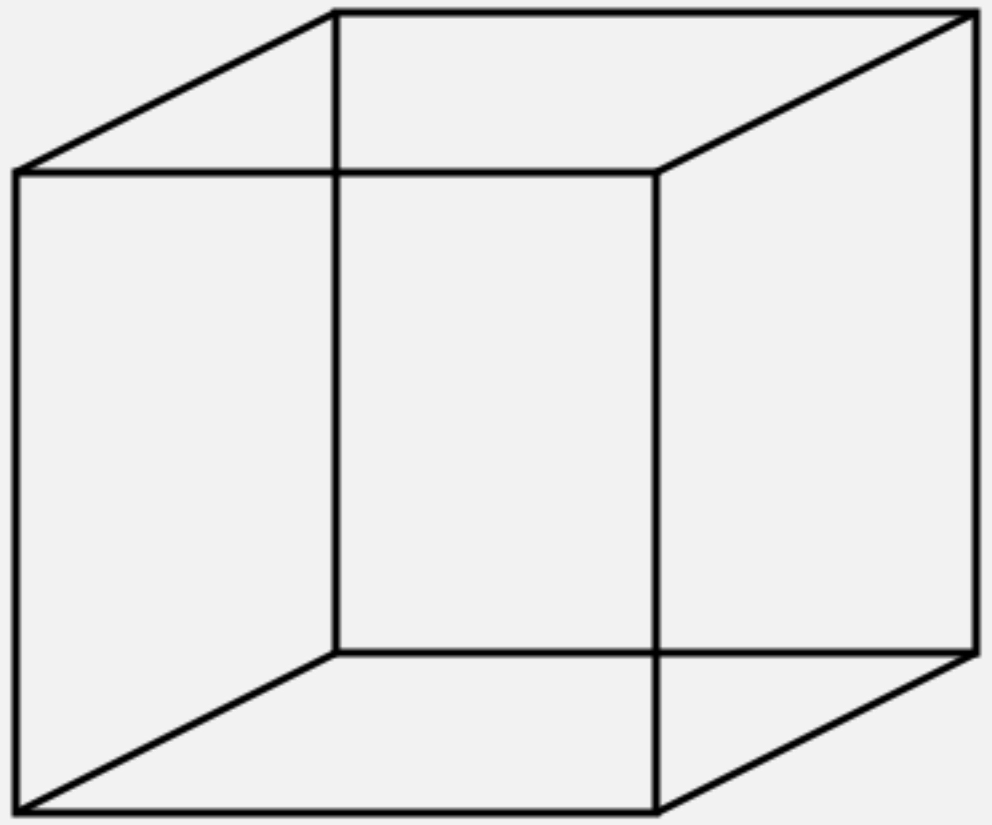
An example over $\mathbb{F}_7$:

$$t = \sum_{i,j,k} e_i \otimes e_j \otimes e_k = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}, \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}, \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}, \qquad A = \begin{bmatrix} 6 & 3 & 6 \\ 5 & 6 & 3 \\ 4 & 4 & 4 \end{bmatrix}$$

$$(A, I, I) \star t := \sum_{i,j,k} Ae_i \otimes e_j \otimes e_k = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}, \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 5 & 5 & 5 \\ 5 & 5 & 5 \\ 5 & 5 & 5 \end{bmatrix}$$

$$= \sum_{j,k} Ae_1 \otimes e_j \otimes e_k + \sum_{j,k} Ae_2 \otimes e_j \otimes e_k + \sum_{j,k} Ae_3 \otimes e_j \otimes e_k$$

$$\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$



Tensors

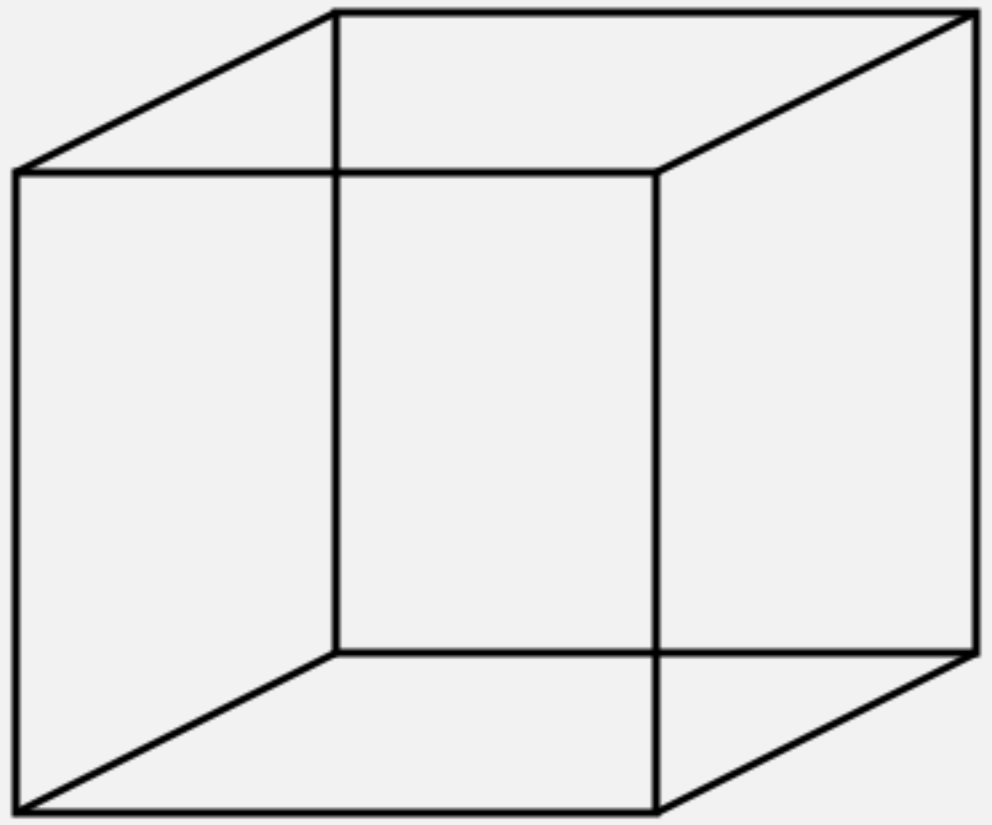
An example over $\mathbb{F}_7$:

$$t = \sum_{i,j,k} e_i \otimes e_j \otimes e_k = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}, \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}, \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}, \qquad A = \begin{bmatrix} 6 & 3 & 6 \\ 5 & 6 & 3 \\ 4 & 4 & 4 \end{bmatrix}$$

$$(A, I, I) \star t := \sum_{i,j,k} Ae_i \otimes e_j \otimes e_k = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}, \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 5 & 5 & 5 \\ 5 & 5 & 5 \\ 5 & 5 & 5 \end{bmatrix}$$

$$= \sum_{j,k} Ae_1 \otimes e_j \otimes e_k + \sum_{j,k} Ae_2 \otimes e_j \otimes e_k + \sum_{j,k} Ae_3 \otimes e_j \otimes e_k$$

$$[6,5,4] \otimes \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

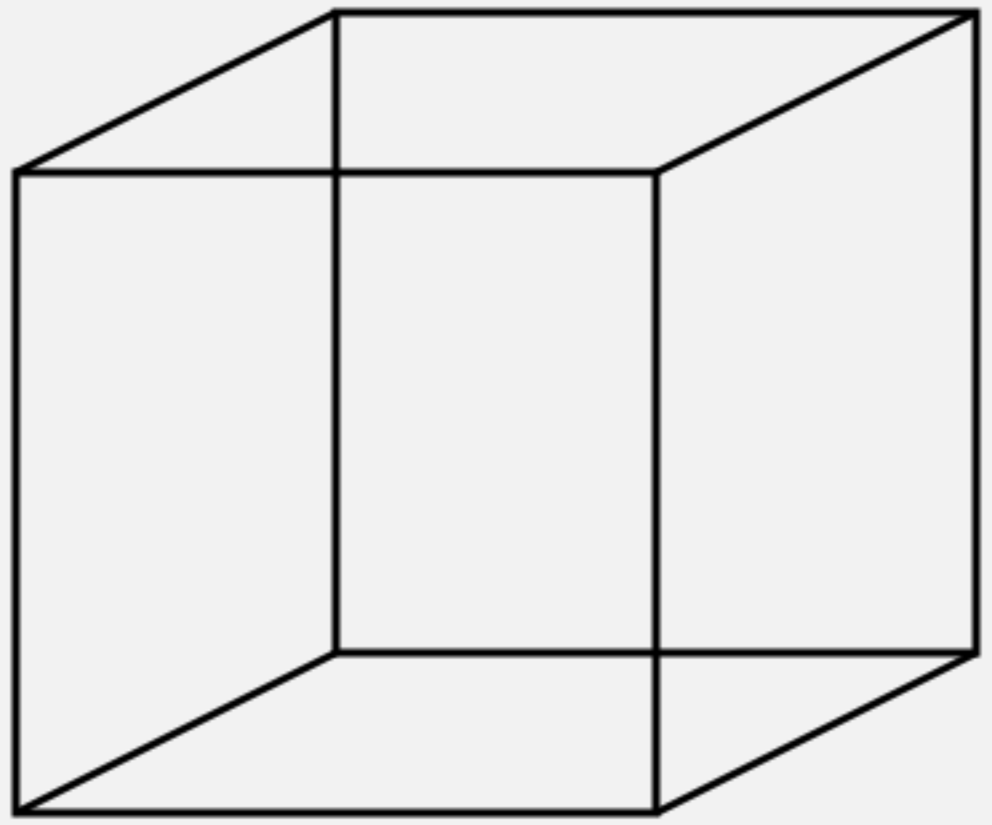


Tensors

An example over $\mathbb{F}_7$:

$$t = \sum_{i,j,k} e_i \otimes e_j \otimes e_k = \begin{bmatrix}1&1&1\\1&1&1\\1&1&1\end{bmatrix}, \begin{bmatrix}1&1&1\\1&1&1\\1&1&1\end{bmatrix}, \begin{bmatrix}1&1&1\\1&1&1\\1&1&1\end{bmatrix}, \qquad A = \begin{bmatrix}6&3&6\\5&6&3\\4&4&4\end{bmatrix}$$

$$(A, I, I) \star t := \sum_{i,j,k} Ae_i \otimes e_j \otimes e_k = \begin{bmatrix}1&1&1\\1&1&1\\1&1&1\end{bmatrix}, \begin{bmatrix}0&0&0\\0&0&0\\0&0&0\end{bmatrix}, \begin{bmatrix}5&5&5\\5&5&5\\5&5&5\end{bmatrix}$$



$$= \sum_{j,k} Ae_1 \otimes e_j \otimes e_k + \sum_{j,k} Ae_2 \otimes e_j \otimes e_k + \sum_{j,k} Ae_3 \otimes e_j \otimes e_k$$

$$[6,5,4] \otimes \begin{bmatrix}1&1&1\\1&1&1\\1&1&1\end{bmatrix}$$

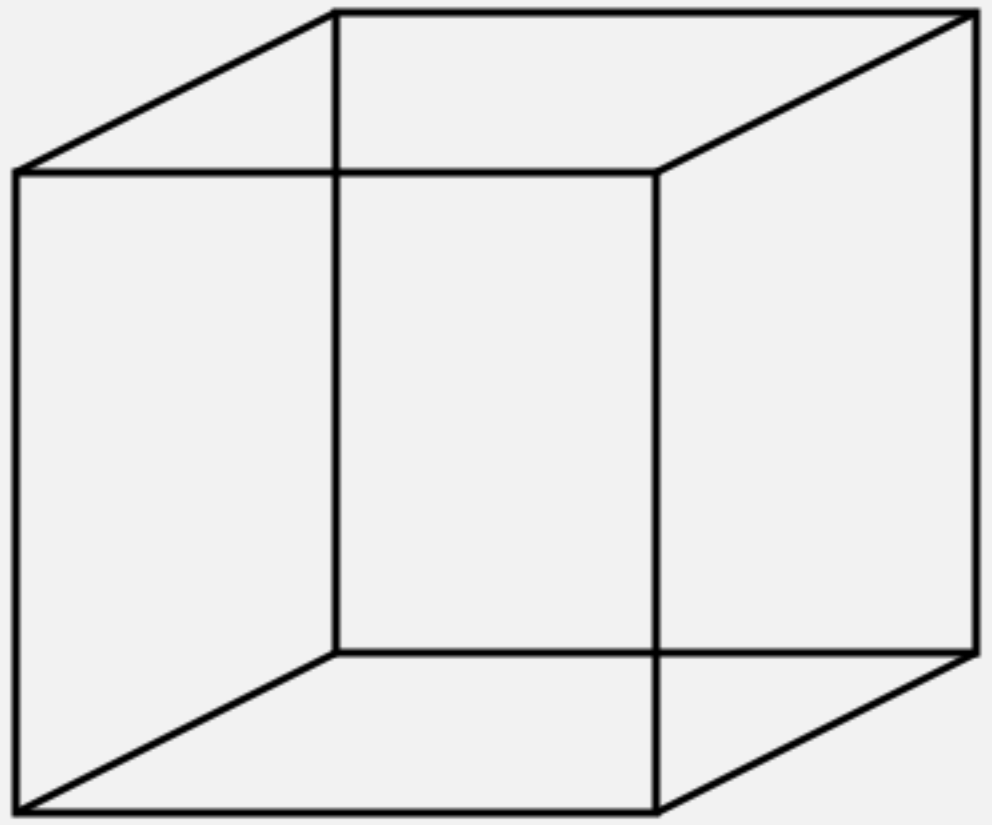
$$= \begin{bmatrix}6&6&6\\6&6&6\\6&6&6\end{bmatrix}, \begin{bmatrix}5&5&5\\5&5&5\\5&5&5\end{bmatrix}, \begin{bmatrix}4&4&4\\4&4&4\\4&4&4\end{bmatrix}$$

Tensors

An example over $\mathbb{F}_7$:

$$t = \sum_{i,j,k} e_i \otimes e_j \otimes e_k = \begin{bmatrix}1&1&1\\1&1&1\\1&1&1\end{bmatrix}, \begin{bmatrix}1&1&1\\1&1&1\\1&1&1\end{bmatrix}, \begin{bmatrix}1&1&1\\1&1&1\\1&1&1\end{bmatrix}, \qquad A = \begin{bmatrix}6&3&6\\5&6&3\\4&4&4\end{bmatrix}$$

$$(A, I, I) \star t := \sum_{i,j,k} Ae_i \otimes e_j \otimes e_k = \begin{bmatrix}1&1&1\\1&1&1\\1&1&1\end{bmatrix}, \begin{bmatrix}0&0&0\\0&0&0\\0&0&0\end{bmatrix}, \begin{bmatrix}5&5&5\\5&5&5\\5&5&5\end{bmatrix}$$



$$= \sum_{j,k} \underline{Ae_1} \otimes \underline{e_j \otimes e_k} + \sum_{j,k} Ae_2 \otimes e_j \otimes e_k + \sum_{j,k} Ae_3 \otimes e_j \otimes e_k$$

$$[6,5,4] \otimes \begin{bmatrix}1&1&1\\1&1&1\\1&1&1\end{bmatrix}$$

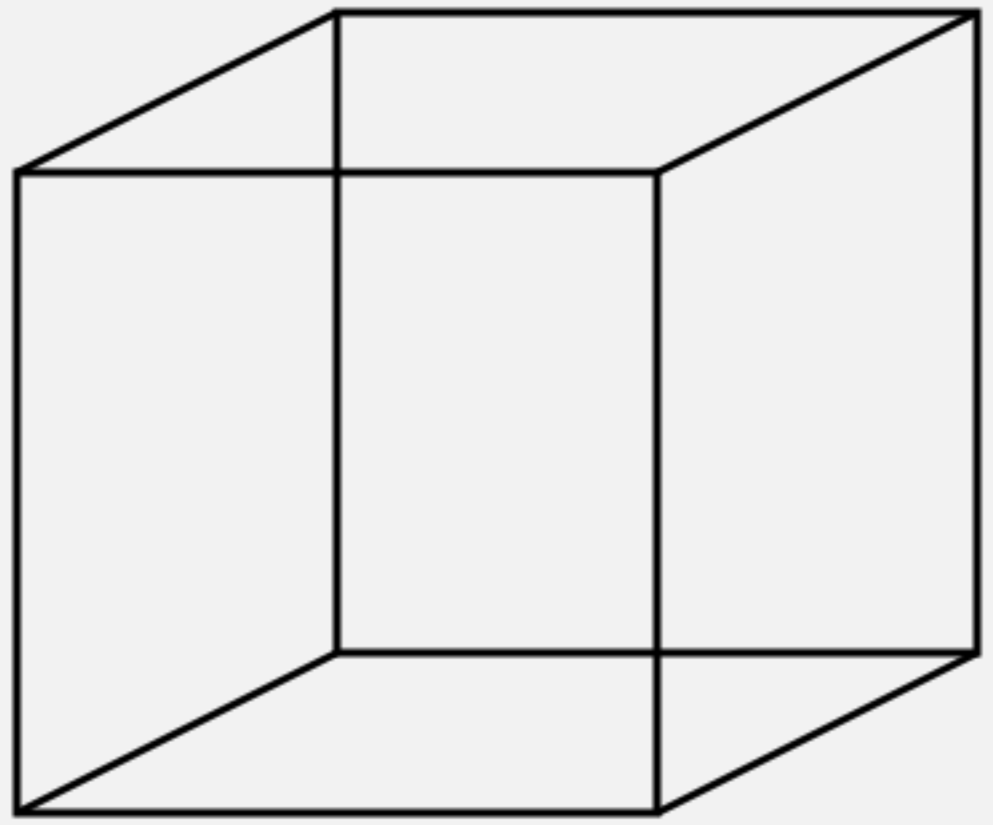
$$= \begin{bmatrix}6&6&6\\6&6&6\\6&6&6\end{bmatrix}, \begin{bmatrix}5&5&5\\5&5&5\\5&5&5\end{bmatrix}, \begin{bmatrix}4&4&4\\4&4&4\\4&4&4\end{bmatrix} + \begin{bmatrix}3&3&3\\3&3&3\\3&3&3\end{bmatrix}, \begin{bmatrix}6&6&6\\6&6&6\\6&6&6\end{bmatrix}, \begin{bmatrix}4&4&4\\4&4&4\\4&4&4\end{bmatrix} + \begin{bmatrix}6&6&6\\6&6&6\\6&6&6\end{bmatrix}, \begin{bmatrix}3&3&3\\3&3&3\\3&3&3\end{bmatrix}, \begin{bmatrix}4&4&4\\4&4&4\\4&4&4\end{bmatrix}$$

Tensors

An example over $\mathbb{F}_7$:

$$t = \sum_{i,j,k} e_i \otimes e_j \otimes e_k = \begin{bmatrix}1&1&1\\1&1&1\\1&1&1\end{bmatrix}, \begin{bmatrix}1&1&1\\1&1&1\\1&1&1\end{bmatrix}, \begin{bmatrix}1&1&1\\1&1&1\\1&1&1\end{bmatrix}, \qquad A = \begin{bmatrix}6&3&6\\5&6&3\\4&4&4\end{bmatrix}$$

$$(A, I, I) \star t := \sum_{i,j,k} Ae_i \otimes e_j \otimes e_k = \begin{bmatrix}1&1&1\\1&1&1\\1&1&1\end{bmatrix}, \begin{bmatrix}0&0&0\\0&0&0\\0&0&0\end{bmatrix}, \begin{bmatrix}5&5&5\\5&5&5\\5&5&5\end{bmatrix},$$



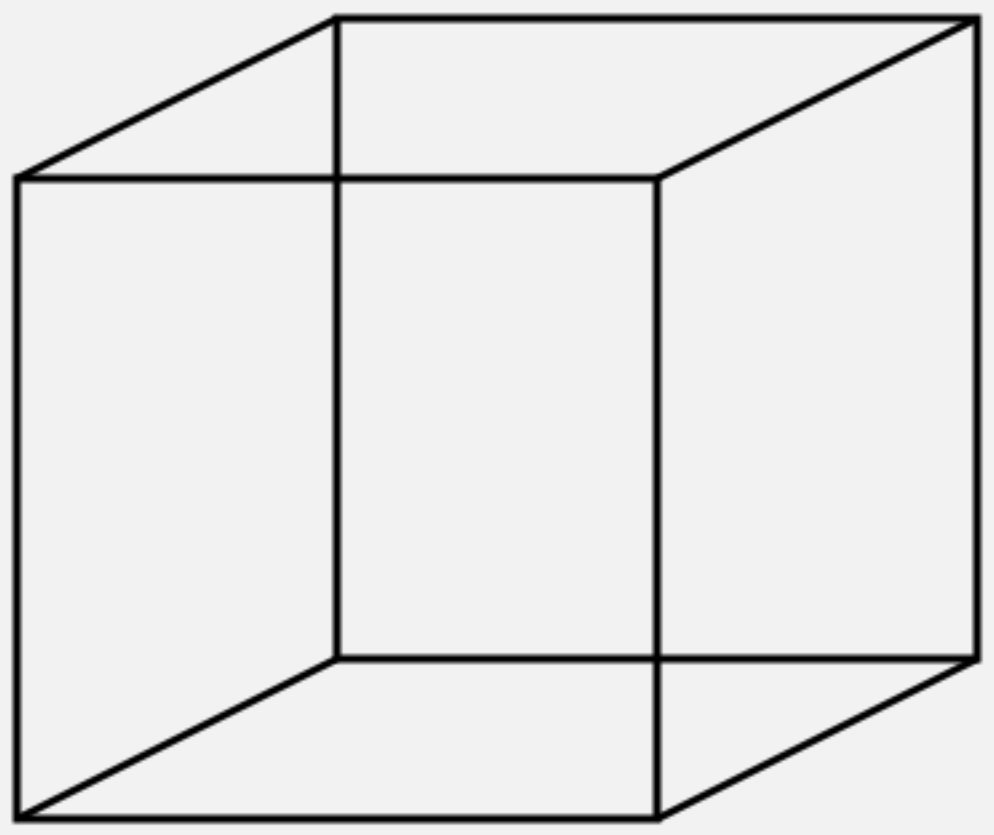
Tensors

An example over $\mathbb{F}_7$:

$$t = \sum_{i,j,k} e_i \otimes e_j \otimes e_k = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}, \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}, \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}, \qquad A = \begin{bmatrix} 6 & 3 & 6 \\ 5 & 6 & 3 \\ 4 & 4 & 4 \end{bmatrix}$$

$$(A, I, I) \star t := \sum_{i,j,k} Ae_i \otimes e_j \otimes e_k = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}, \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 5 & 5 & 5 \\ 5 & 5 & 5 \\ 5 & 5 & 5 \end{bmatrix},$$

$$(I, A, I) \star t := \sum_{i,j,k} e_i \otimes Ae_j \otimes e_k = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ 5 & 5 & 5 \end{bmatrix}, \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ 5 & 5 & 5 \end{bmatrix}, \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ 5 & 5 & 5 \end{bmatrix},$$

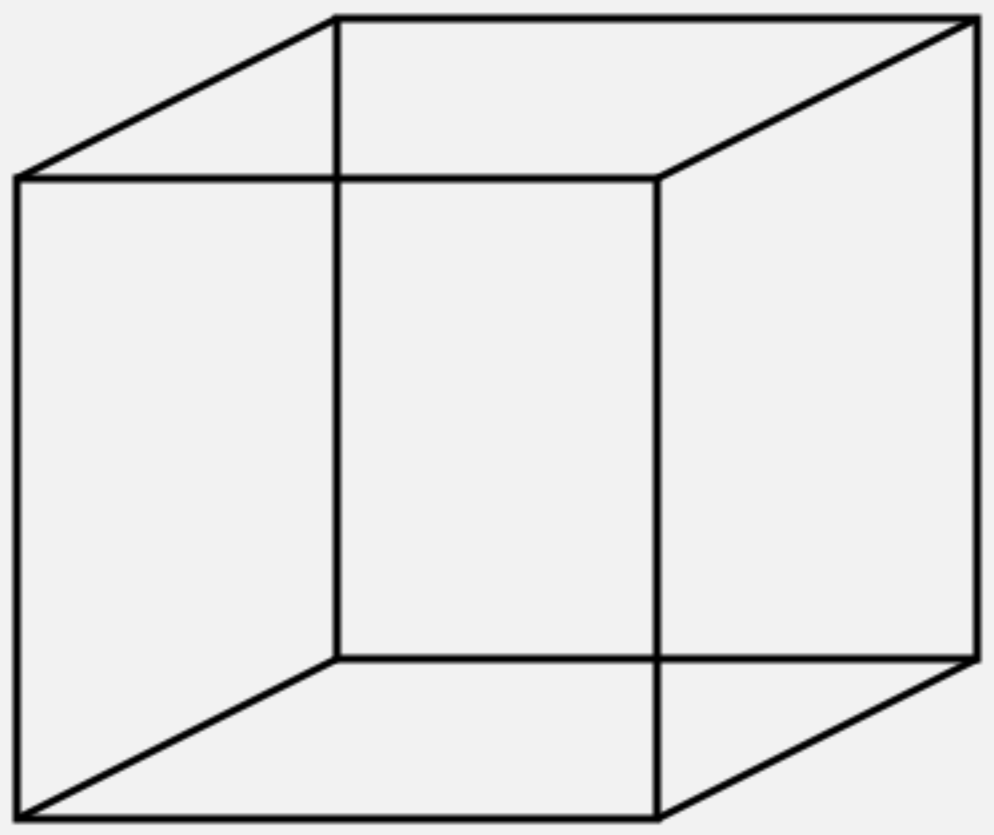


Tensors

An example over $\mathbb{F}_7$:

$$t = \sum_{i,j,k} e_i \otimes e_j \otimes e_k = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}, \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}, \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}, \qquad A = \begin{bmatrix} 6 & 3 & 6 \\ 5 & 6 & 3 \\ 4 & 4 & 4 \end{bmatrix}$$

$$(A, I, I) \star t := \sum_{i,j,k} Ae_i \otimes e_j \otimes e_k = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}, \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 5 & 5 & 5 \\ 5 & 5 & 5 \\ 5 & 5 & 5 \end{bmatrix},$$



$$(I, A, I) \star t := \sum_{i,j,k} e_i \otimes Ae_j \otimes e_k = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ 5 & 5 & 5 \end{bmatrix}, \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ 5 & 5 & 5 \end{bmatrix}, \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ 5 & 5 & 5 \end{bmatrix},$$

$$(I, I, A) \star t := \sum_{i,j,k} e_i \otimes e_j \otimes Ae_k = \begin{bmatrix} 1 & 0 & 5 \\ 1 & 0 & 5 \\ 1 & 0 & 5 \end{bmatrix}, \begin{bmatrix} 1 & 0 & 5 \\ 1 & 0 & 5 \\ 1 & 0 & 5 \end{bmatrix}, \begin{bmatrix} 1 & 0 & 5 \\ 1 & 0 & 5 \\ 1 & 0 & 5 \end{bmatrix},$$

Tensors

Tensors

Decisional Tensor Isomorphism Problem (dTIP):

Given **random** v_0, v_1 **decide** whether there exists

(A, B, C) such that $(A, B, C) \star v_0 = v_1$

Tensors

Decisional Tensor Isomorphism Problem (dTIP):

Given **random** v_0, v_1 **decide** whether there exists

(A, B, C) such that $(A, B, C) \star v_0 = v_1$

Computational Tensor Isomorphism Problem (cTIP):

Given **random** v_0, v_1 **compute**

(A, B, C) such that $(A, B, C) \star v_0 = v_1$

TI-family

* this is for the case where all the dimensions are n

TI-family

A **code** is the linear subspace that is generated by a set of matrices

$$G := [G_1, \ldots G_n]$$

* this is for the case where all the dimensions are n

TI-family

A **code** is the linear subspace that is generated by a set of matrices

$$G := [G_1, \ldots G_n]$$

Codes are **equivalent** if they generate the same subspace.

Equivalent codes take the form

$$G' = [\lambda_{1,1}G_1 + \cdots \lambda_{1,n}G_n, \cdots \lambda_{n,1}G_1 + \cdots \lambda_{n,n}G_n]$$

* this is for the case where all the dimensions are n

TI-family

A **code** is the linear subspace that is generated by a set of matrices

$$G := [G_1, \ldots G_n]$$

Codes are **equivalent** if they generate the same subspace.

Equivalent codes take the form

$$G' = [\lambda_{1,1}G_1 + \cdots \lambda_{1,n}G_n, \cdots \lambda_{n,1}G_1 + \cdots \lambda_{n,n}G_n]$$

Computational Tensor Isomorphism Problem (cTIP):

Given **random** v_0, v_1 **compute**

(A, B, C) such that $(A, B, C) \star v_0 = v_1$

* this is for the case where all the dimensions are n

TI-family

A **code** is the linear subspace that is generated by a set of matrices

$$G := [G_1, \ldots G_n]$$

Codes are **equivalent** if they generate the same subspace.

Equivalent codes take the form

$$G' = [\lambda_{1,1}G_1 + \cdots \lambda_{1,n}G_n, \cdots \lambda_{n,1}G_1 + \cdots \lambda_{n,n}G_n]$$

Computational Tensor Isomorphism Problem (cTIP):

Given **random** v_0, v_1 **compute**

(A, B, C) such that $(A, B, C) \star v_0 = v_1$

Matrix Code Equivalence (MCE):

Given G, G' compute (if it exists)

(A, B, C) such that $(A, B, C) \star G = G'$

* this is for the case where all the dimensions are n

TI-family

TI-family

A **trilinear form** is a map

$$\varphi : \mathbb{F}_q^n \times \mathbb{F}_q^n \times \mathbb{F}_q^n \to \mathbb{F}_q,$$

TI-family

A **trilinear form** is a map

$$\varphi : \mathbb{F}_q^n \times \mathbb{F}_q^n \times \mathbb{F}_q^n \to \mathbb{F}_q, \qquad \varphi(e_i, e_j, e_k) := m_{i,j,k}$$

TI-family

A **trilinear form** is a map

$$\varphi : \mathbb{F}_q^n \times \mathbb{F}_q^n \times \mathbb{F}_q^n \to \mathbb{F}_q, \qquad \varphi(e_i, e_j, e_k) := m_{i,j,k}$$

We say two trilinear forms, φ, ψ, are **equivalent** if there exists some $A \in GL_n(\mathbb{F}_q)$ such that

$$\varphi(u, v, w) = \psi(Au, Av, Aw)$$

TI-family

A **trilinear form** is a map

$$\varphi : \mathbb{F}_q^n \times \mathbb{F}_q^n \times \mathbb{F}_q^n \to \mathbb{F}_q, \qquad \varphi(e_i, e_j, e_k) := m_{i,j,k}$$

We say two trilinear forms, φ, ψ, are **equivalent** if there exists some $A \in GL_n(\mathbb{F}_q)$ such that

$$\varphi(u, v, w) = \psi(Au, Av, Aw)$$

Computational Tensor Isomorphism Problem (cTIP):

Given **random** v_0, v_1 **compute**

(A, B, C) such that $(A, B, C) \star v_0 = v_1$

TI-family

A **trilinear form** is a map

$$\varphi : \mathbb{F}_q^n \times \mathbb{F}_q^n \times \mathbb{F}_q^n \to \mathbb{F}_q, \qquad \varphi(e_i, e_j, e_k) := m_{i,j,k}$$

We say two trilinear forms, φ, ψ, are **equivalent** if there exists some $A \in GL_n(\mathbb{F}_q)$ such that

$$\varphi(u, v, w) = \psi(Au, Av, Aw)$$

Computational Tensor Isomorphism Problem (cTIP):

Given **random** v_0, v_1 **compute**

(A, B, C) such that $(A, B, C) \star v_0 = v_1$

Trilinear Form Equivalence (TFE):

Given D, D' compute (if it exists)

(A, B, C) such that $(A, B, C) \star D = D'$

DFG paper

The DFG paper (*Asiacrypt 2023*) seeks to use this hard problem in a commitment scheme

Non-Interactive Commitment from Non-Transitive Group Actions

Giuseppe D'Alconzo[1] [0000−0001−7377−6617], Andrea Flamini[2] [0000−0002−3872−7251], and Andrea Gangemi[2] [0000−0001−9689−8473]

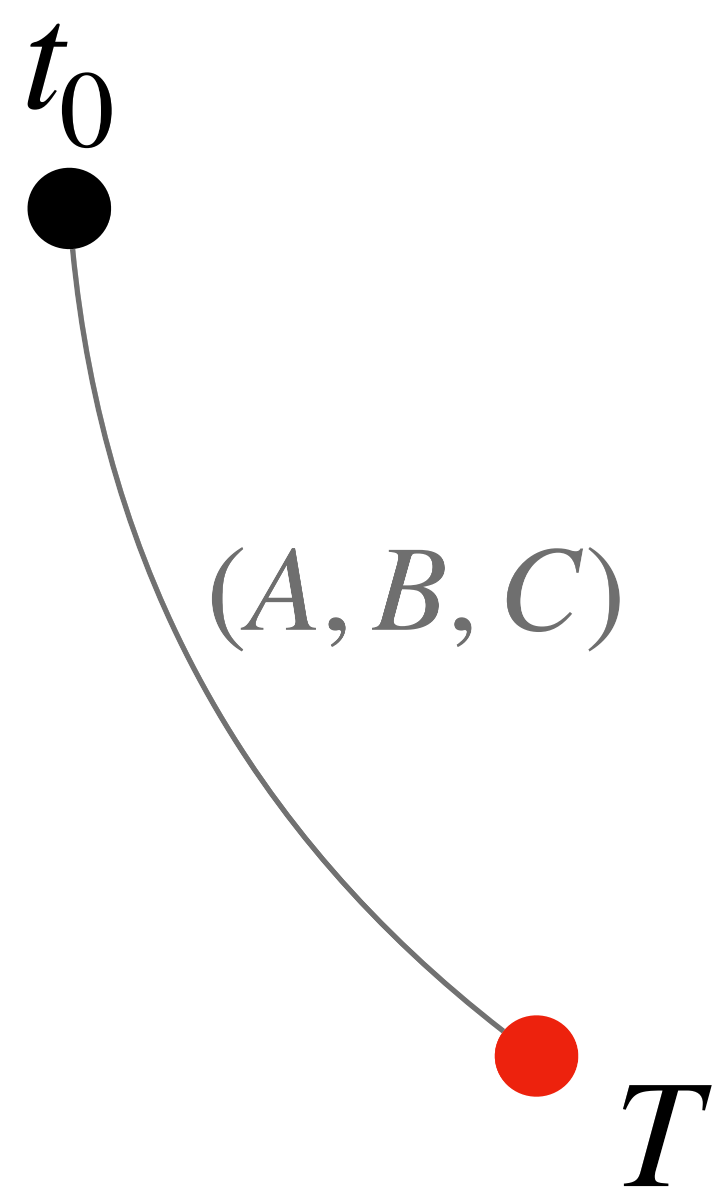
[1] Department of Mathematical Sciences, Politecnico di Torino, Corso Duca degli Abruzzi 24, 10129 Torino, Italy
[2] Department of Mathematics, University of Trento, Povo, 38123 Trento, Italy
giuseppe.dalconzo@polito.it, {andrea.flamini,andrea.gangemi}@unitn.it

Abstract. Group actions are becoming a viable option for post-quantum cryptography assumptions. Indeed, in recent years some works have shown how to construct primitives from assumptions based on isogenies of ellip-

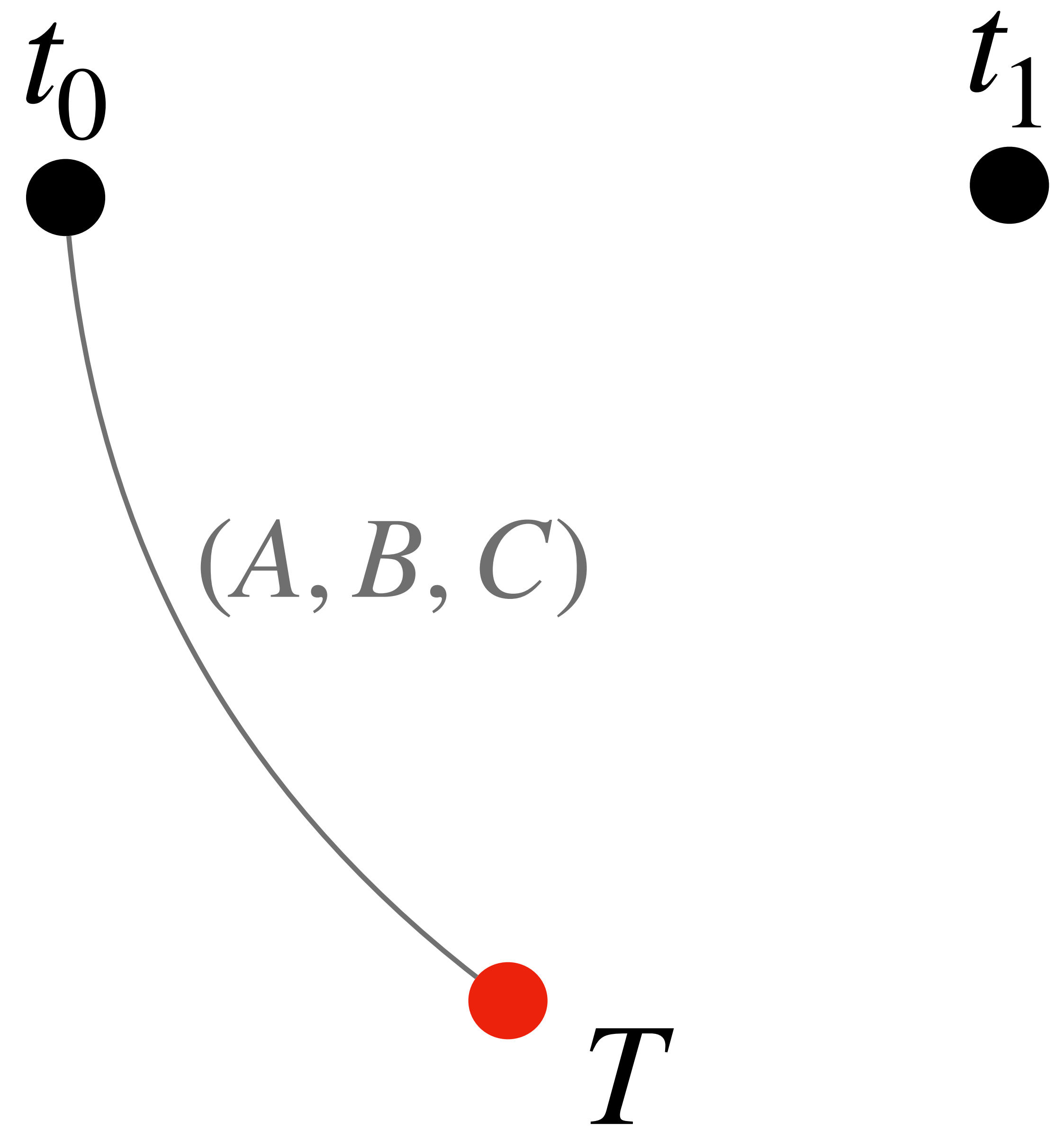
DFG paper

DFG paper

DFG paper

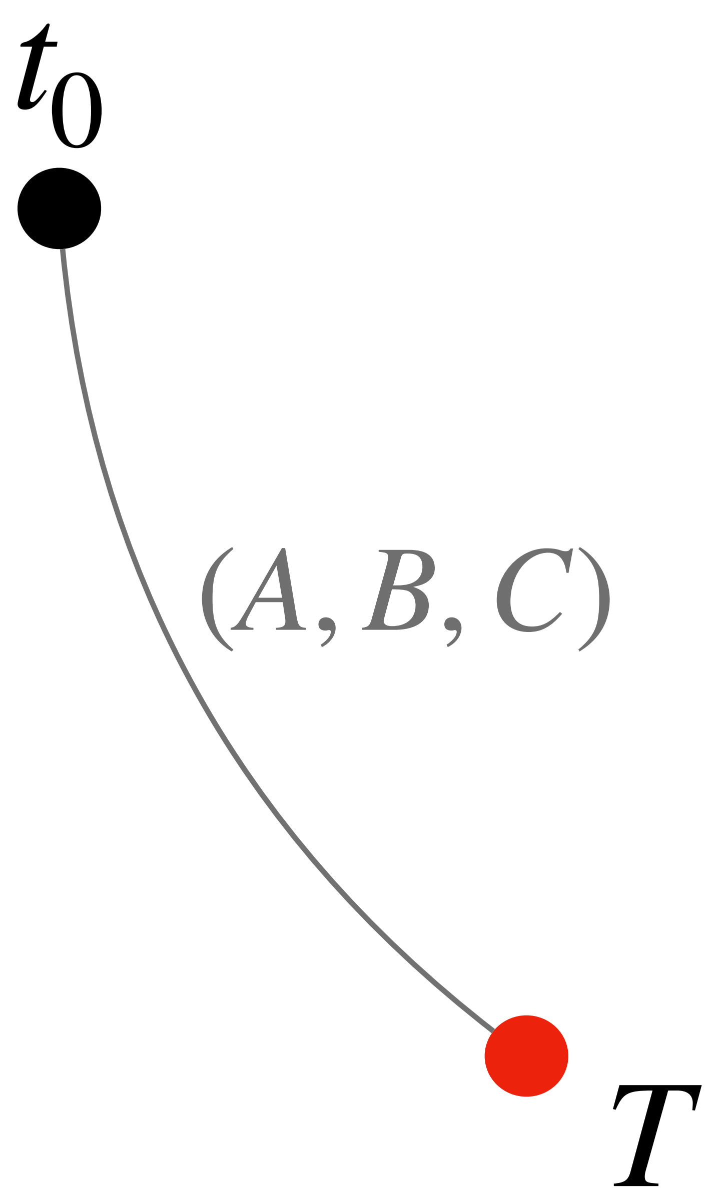


DFG paper



Is it **hiding**?

DFG paper



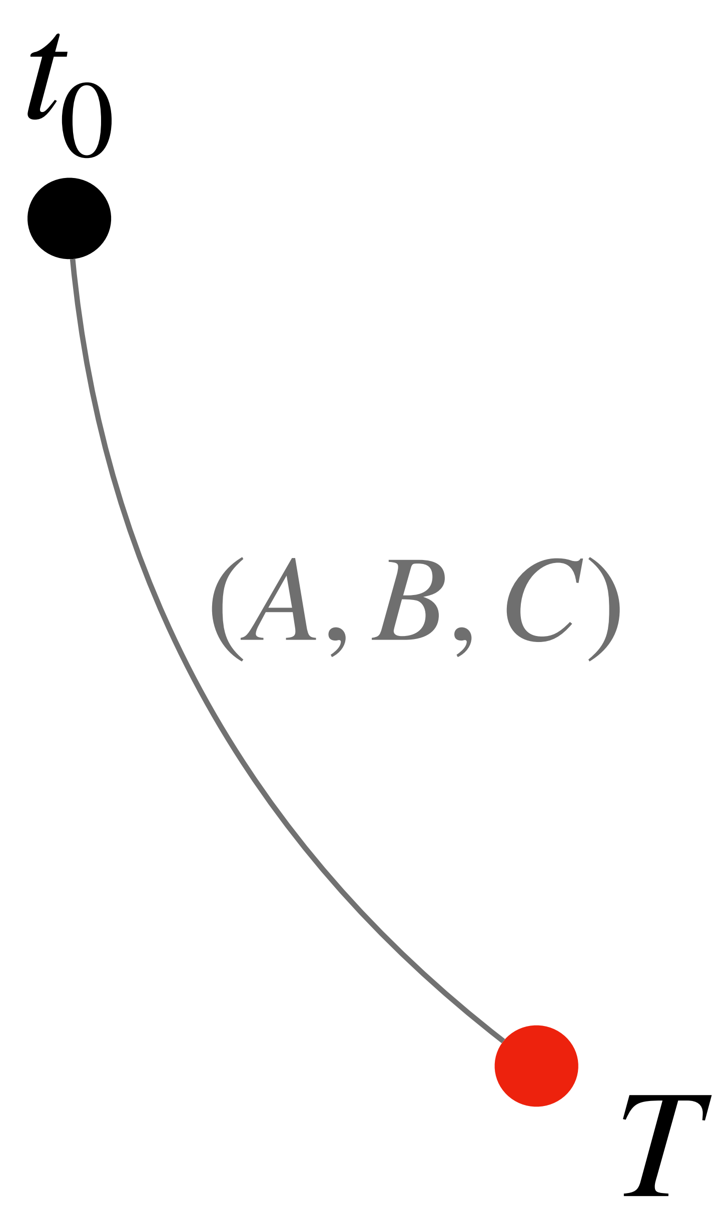
Is it **hiding**?

Decisional Tensor Isomorphism Problem (dTIP):

Given **random** v_0, v_1 **decide** whether there exists

(A, B, C) such that $(A, B, C) \star v_0 = v_1$

DFG paper



Is it **hiding**?

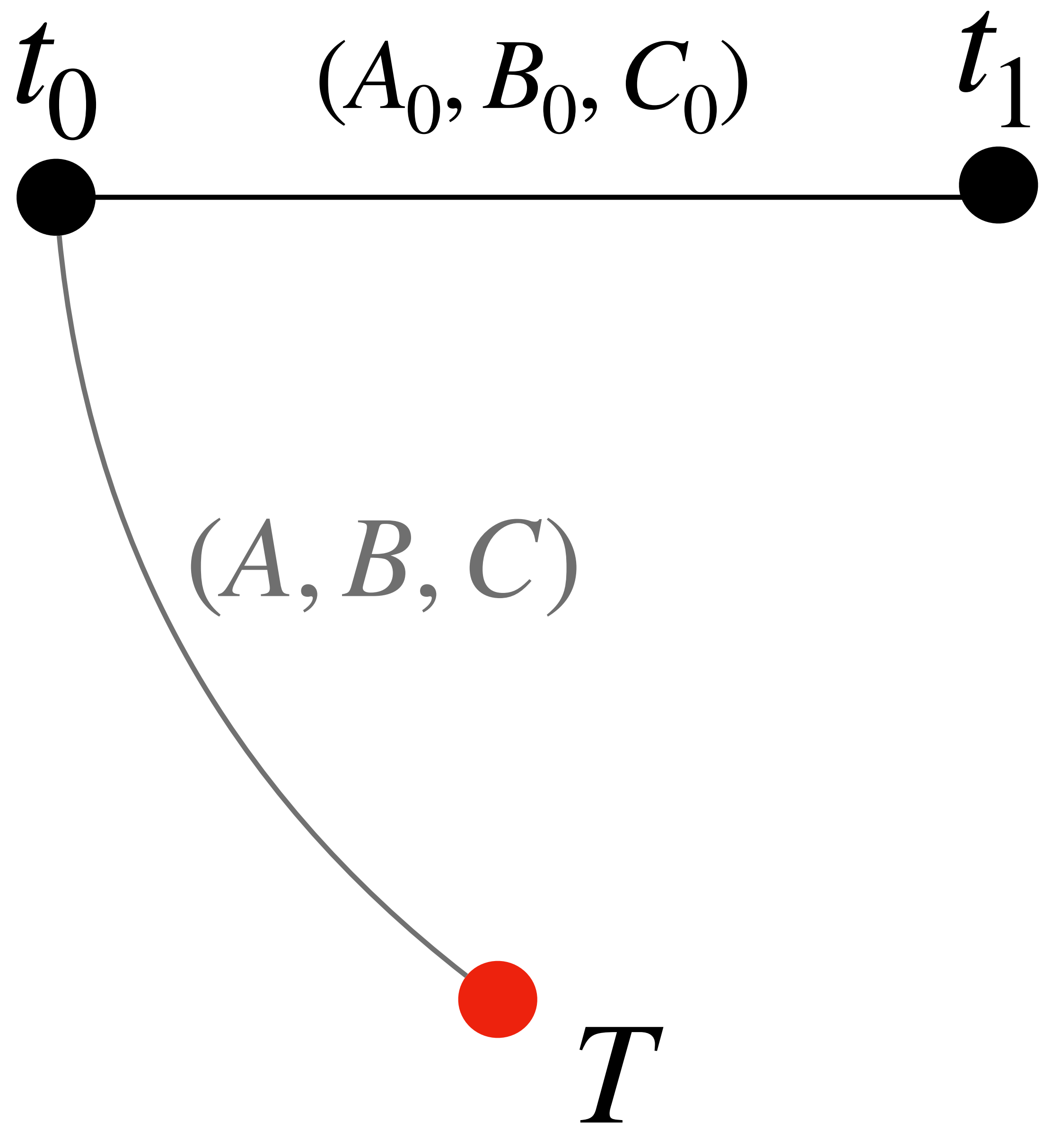
Decisional Tensor Isomorphism Problem (dTIP):

Given **random** v_0, v_1 **decide** whether there exists

(A, B, C) such that $(A, B, C) \star v_0 = v_1$

Is it **binding**?

DFG paper



Is it **hiding**?

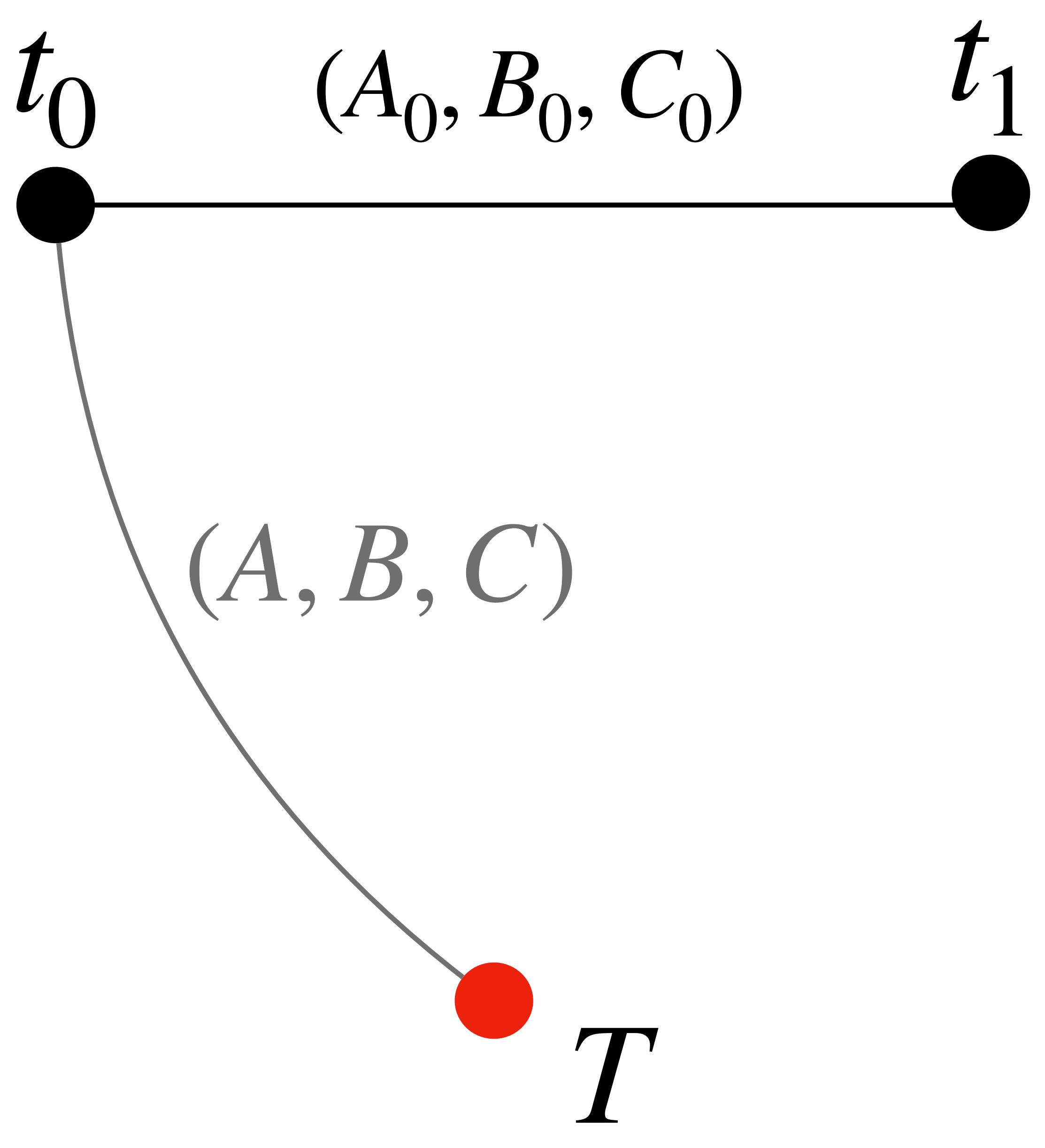
Decisional Tensor Isomorphism Problem (dTIP):

Given **random** v_0, v_1 **decide** whether there exists

(A, B, C) such that $(A, B, C) \star v_0 = v_1$

Is it **binding**?

DFG paper



Is it **hiding**?

Decisional Tensor Isomorphism Problem (dTIP):

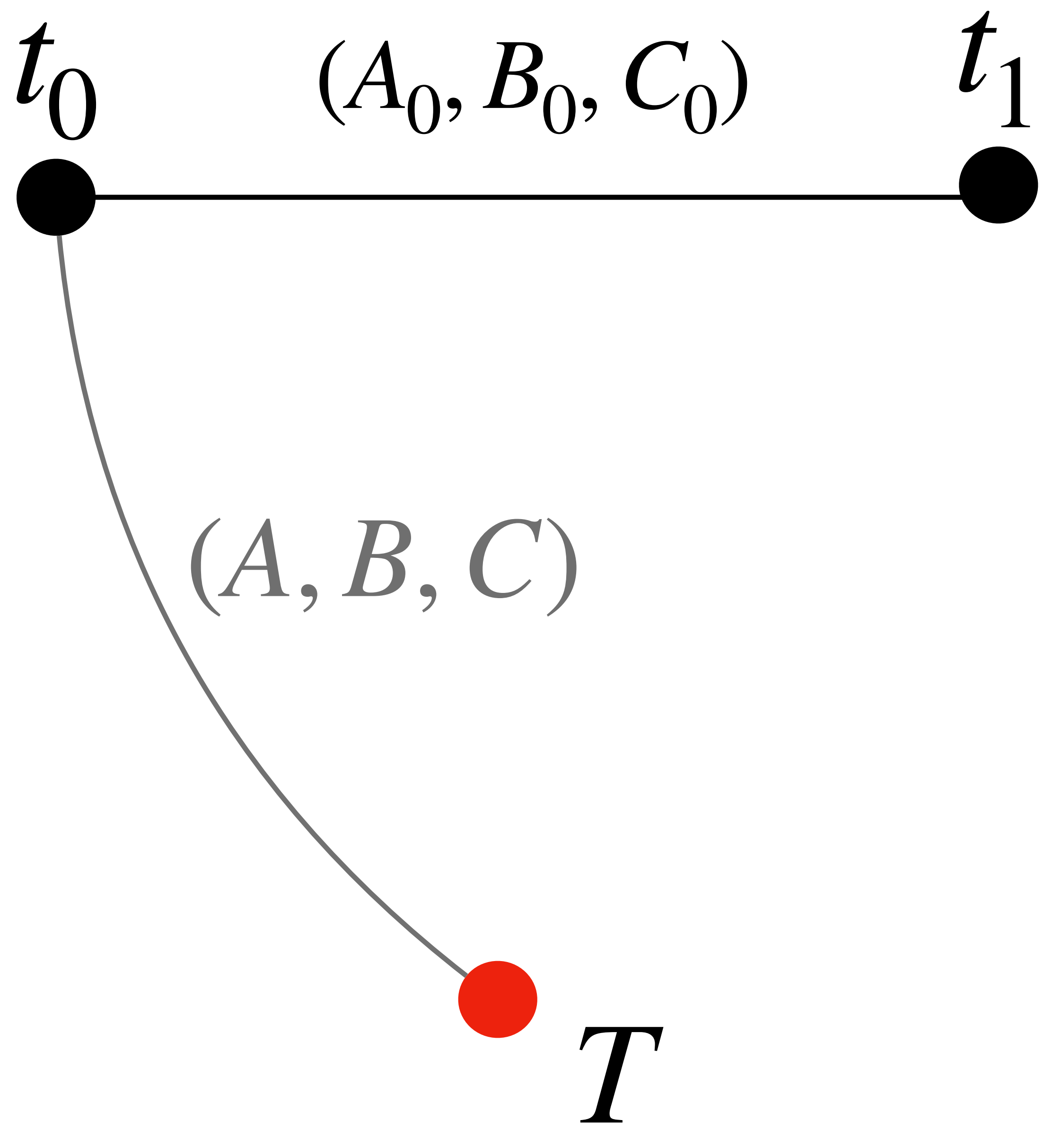
Given **random** v_0, v_1 **decide** whether there exists

(A, B, C) such that $(A, B, C) \star v_0 = v_1$

Is it **binding**?

$$T = (A, B, C) \star t_0$$

DFG paper



Is it **hiding**?

Decisional Tensor Isomorphism Problem (dTIP):

Given **random** v_0, v_1 **decide** whether there exists

(A, B, C) such that $(A, B, C) \star v_0 = v_1$

Is it **binding**?

$$T = (A, B, C) \star t_0$$
$$= (A, B, C) \star ((A_0, B_0, C_0) \star t_1)$$

Tensors

Tensors

rank-1 tensor : $\mathbf{u} \otimes \mathbf{v} \otimes \mathbf{w}$

Tensors

rank-1 tensor : $\mathbf{u} \otimes \mathbf{v} \otimes \mathbf{w}$

rank-k tensor : $\sum_{i=1}^{k} \mathbf{u}_i \otimes \mathbf{v}_i \otimes \mathbf{w}_i$

Tensors

rank-1 tensor : $\mathbf{u} \otimes \mathbf{v} \otimes \mathbf{w}$

rank-k tensor : $\sum_{i=1}^{k} \mathbf{u}_i \otimes \mathbf{v}_i \otimes \mathbf{w}_i$

hard problem :

given a tensor $T := [T_1, \ldots T_n]$,

compute the rank of T

Tensors

rank-1 tensor : $\mathbf{u} \otimes \mathbf{v} \otimes \mathbf{w}$

rank-k tensor : $\sum_{i=1}^{k} \mathbf{u}_i \otimes \mathbf{v}_i \otimes \mathbf{w}_i$

hard problem :

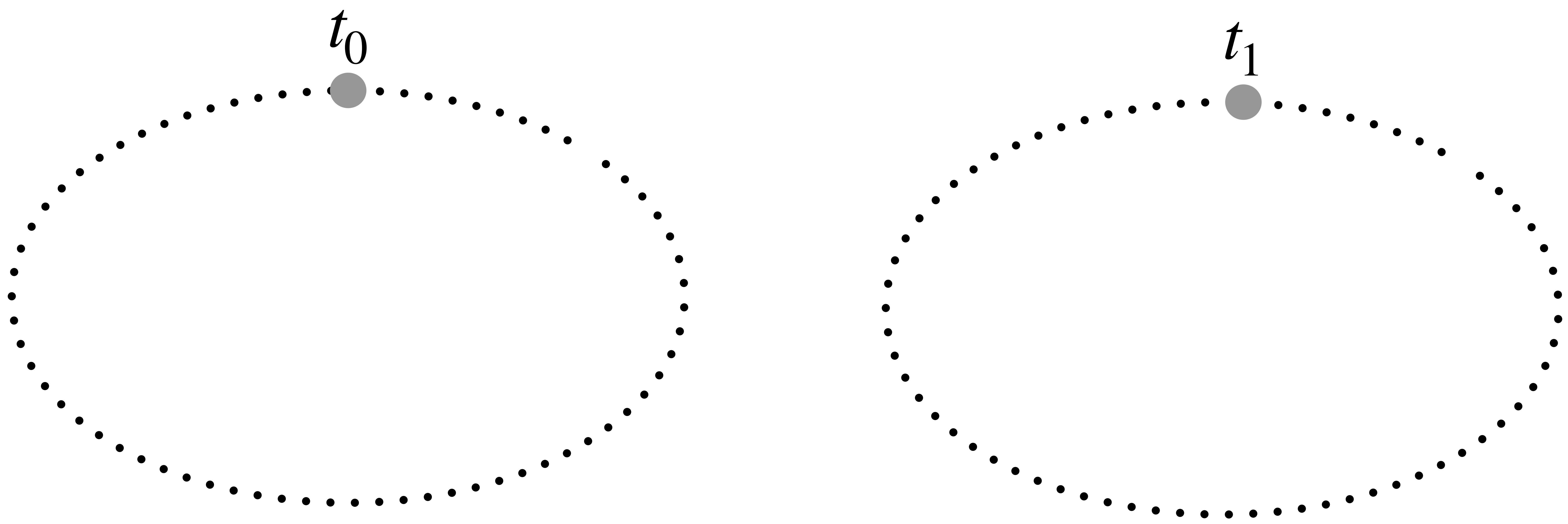
given a tensor $T := [T_1, \ldots T_n]$,

compute the rank of T

For **random** tensors, this problem is believed to be hard

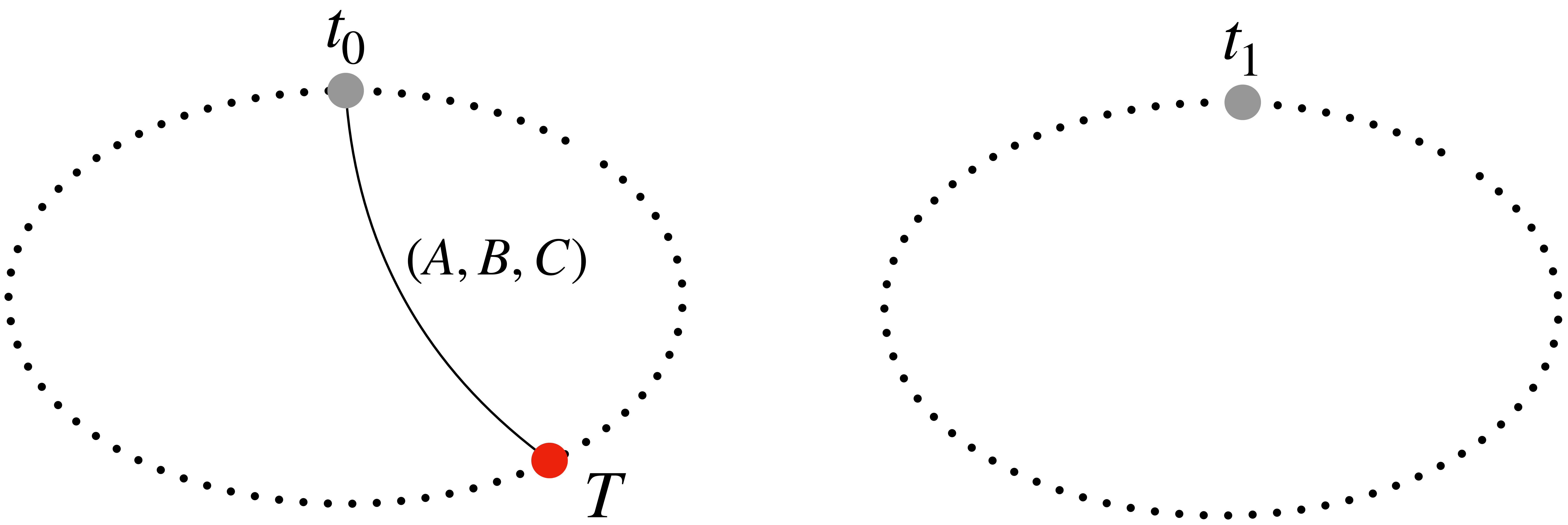
DFG paper

DFG paper



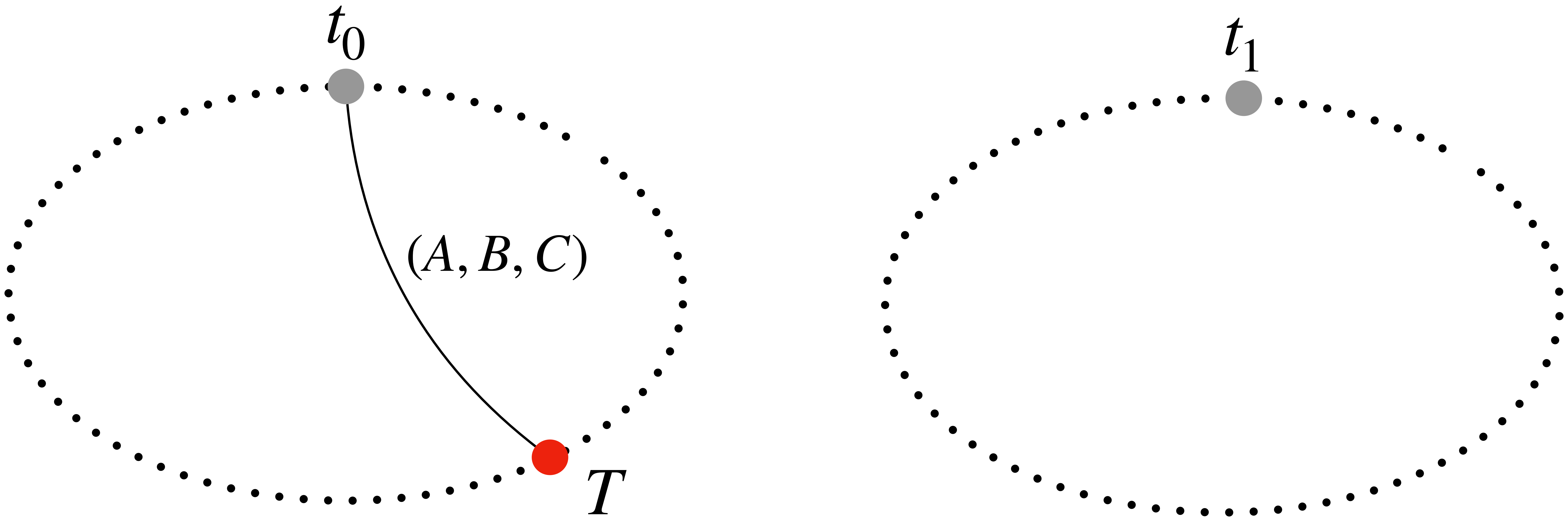
- we need t_0 and t_1 to be in different orbits

DFG paper



- we need t_0 and t_1 to be in different orbits

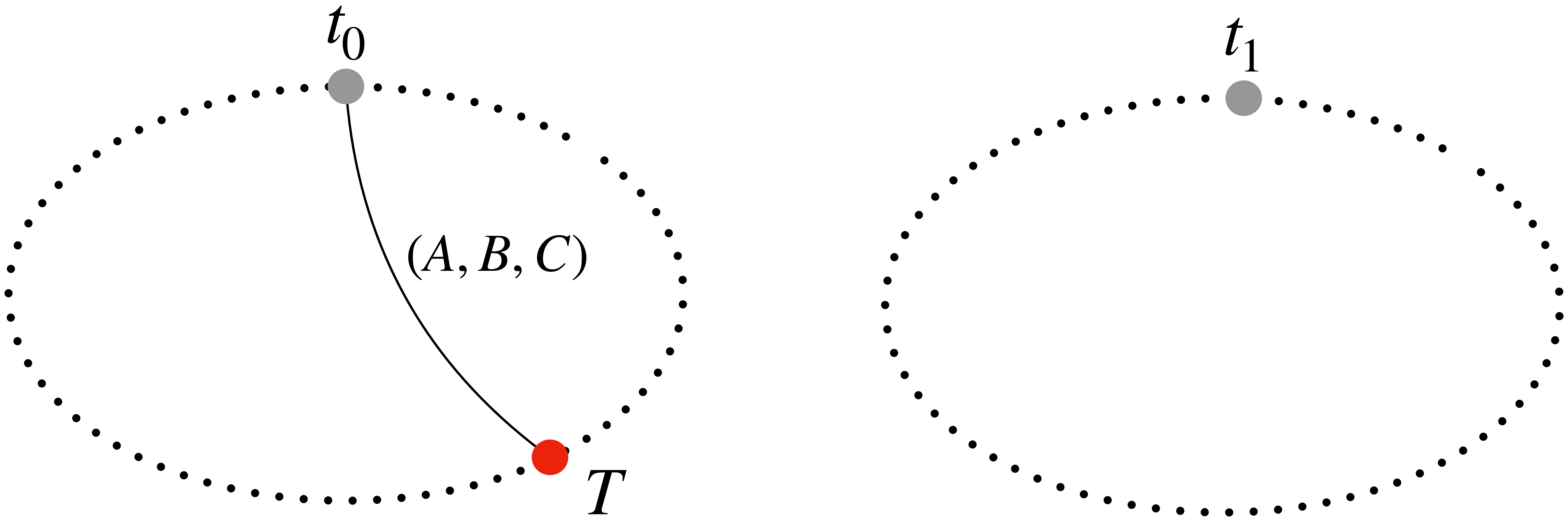
DFG paper



- we need t_0 and t_1 to be in different orbits

Lemma : $rank((A, B, C) \star t) = rank(t)$

DFG paper



- we need t_0 and t_1 to be in different orbits

Lemma : $rank((A, B, C) \star t) = rank(t)$

-different ranks ensures this

DFG paper

DFG paper

We need two tensors with different rank… but computing rank is hard…

DFG paper

We need two tensors with different rank… but computing rank is hard…

$$t_0 = \sum_{i=1}^{3} e_i \otimes e_i \otimes e_i = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$t_1 = \sum_{i=1}^{2} e_i \otimes e_i \otimes e_i = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

DFG paper

We need two tensors with different rank… but computing rank is hard…

$$t_0 = \sum_{i=1}^{3} e_i \otimes e_i \otimes e_i = \begin{bmatrix}1&0&0\\0&0&0\\0&0&0\end{bmatrix}, \begin{bmatrix}0&0&0\\0&1&0\\0&0&0\end{bmatrix}, \begin{bmatrix}0&0&0\\0&0&0\\0&0&1\end{bmatrix}$$

$$t_1 = \sum_{i=1}^{2} e_i \otimes e_i \otimes e_i = \begin{bmatrix}1&0&0\\0&0&0\\0&0&0\end{bmatrix}, \begin{bmatrix}0&0&0\\0&1&0\\0&0&0\end{bmatrix}, \begin{bmatrix}0&0&0\\0&0&0\\0&0&0\end{bmatrix}$$

An example, over $\mathbb{F}_7$:

$$\left(\begin{bmatrix}5&4&2\\4&2&0\\1&1&1\end{bmatrix}, \begin{bmatrix}1&0&0\\0&3&2\\0&2&4\end{bmatrix}, \begin{bmatrix}5&2&3\\2&1&1\\5&4&2\end{bmatrix} \right) \star t_1 = \begin{bmatrix}4&3&4\\3&5&6\\2&1&4\end{bmatrix}, \begin{bmatrix}6&1&6\\5&6&3\\1&4&2\end{bmatrix}, \begin{bmatrix}5&2&5\\6&3&5\\4&2&1\end{bmatrix}$$

DFG paper

DFG paper

t_0, t_1

$T = (A, B, C) \star t_b$

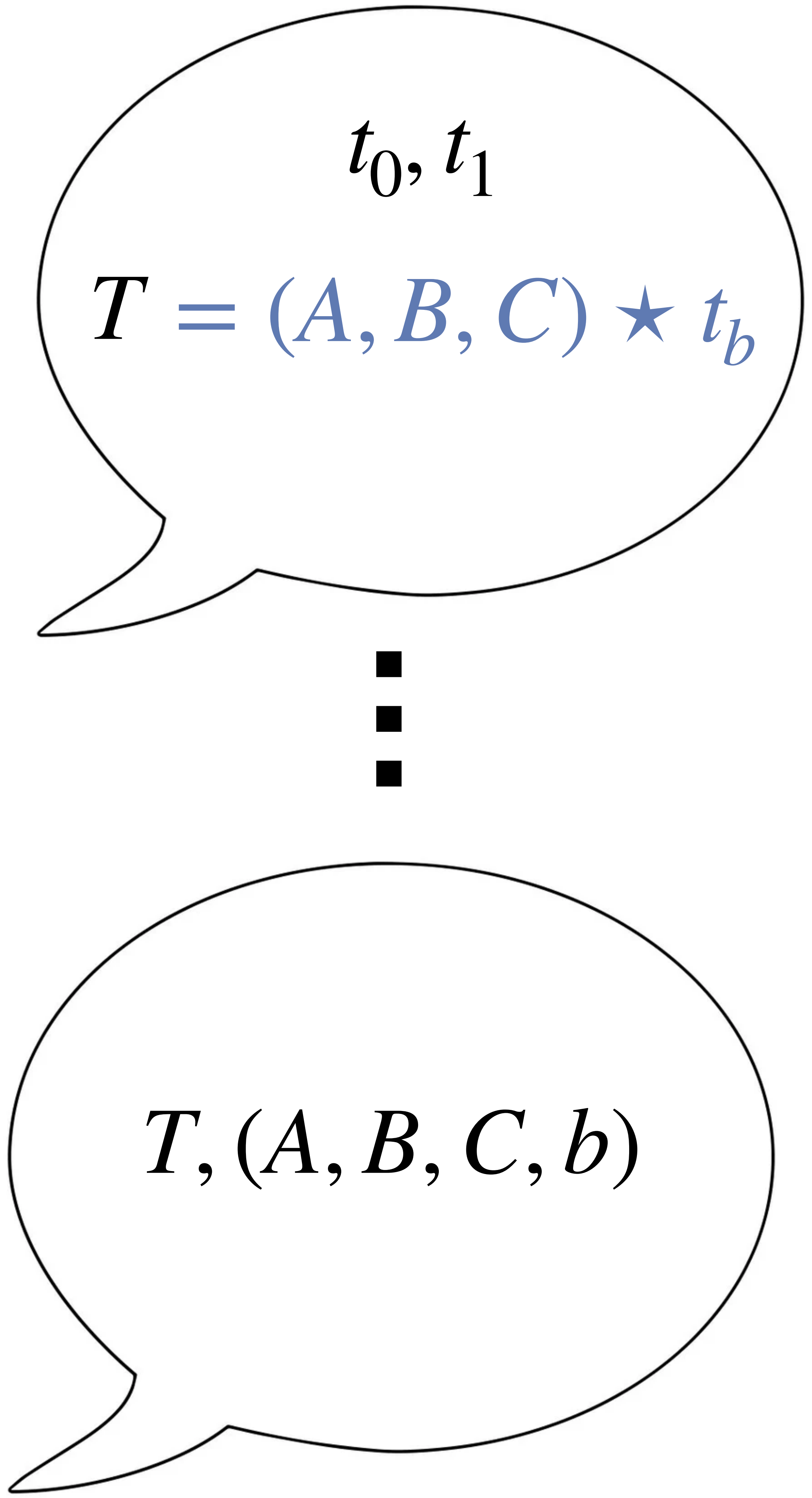
DFG paper

t_0, t_1

$T = (A, B, C) \star t_b$

⋮

DFG paper



DFG paper

t_0, t_1

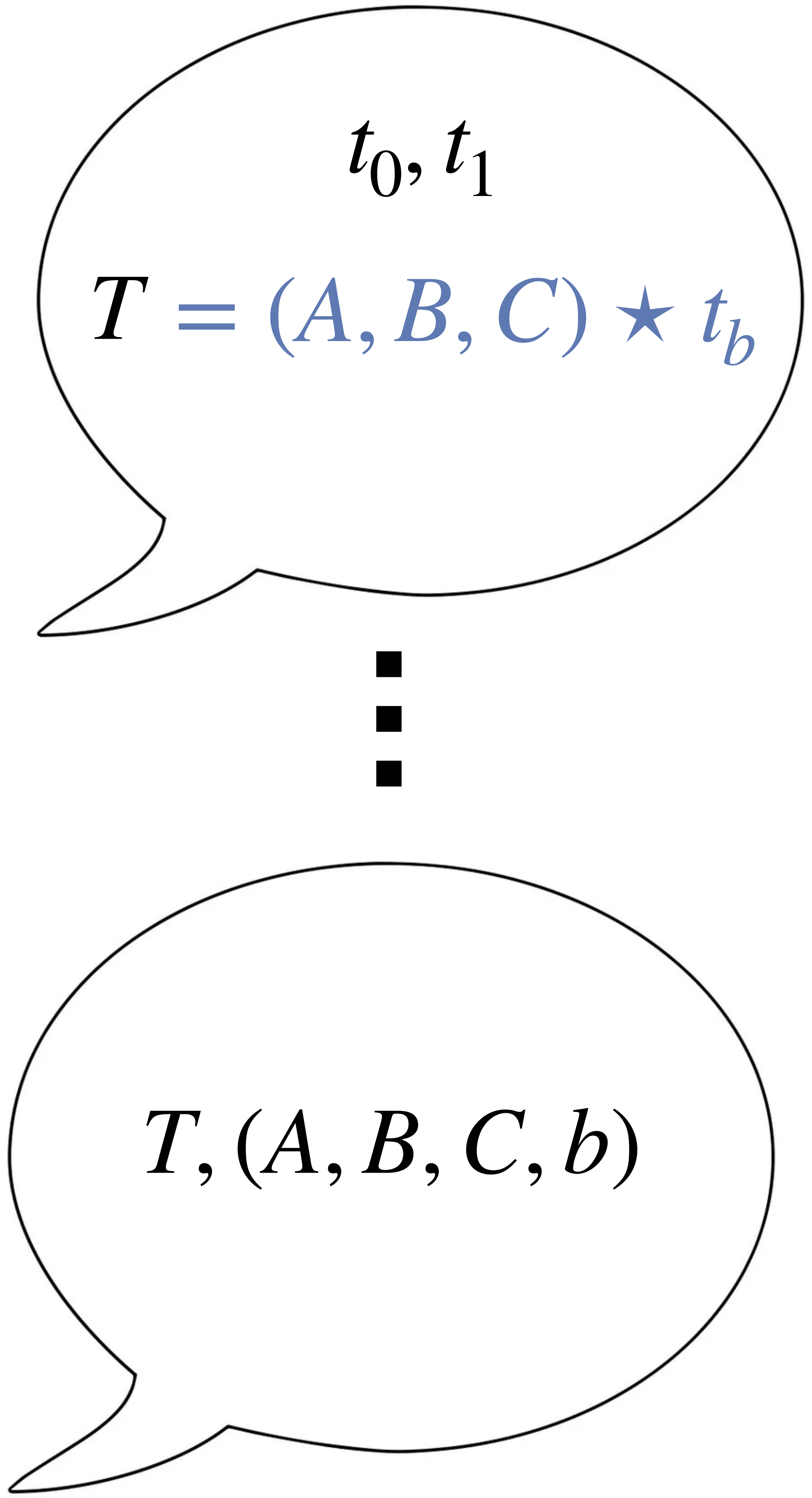
$T = (A, B, C) \star t_b$

⋮

$T, (A, B, C, b)$

binding → perfect

DFG paper



binding → perfect

hiding → related to the dTIP

DFG paper

t_0, t_1

$T = (A, B, C) \star t_b$

⋮

$T, (A, B, C, b)$

binding → perfect

hiding → related to the dTIP

Decisional Tensor Isomorphism Problem (dTIP):

Given **random** v_0, v_1 **decide** whether there exists

(A, B, C) such that $(A, B, C) \star v_0 = v_1$

Distinguishing attack

Distinguishing attack

the rank of a point :

Distinguishing attack

the rank of a point :

We say the rank of $\mathbf{u} = (u_1, \ldots u_n)$ in $T = [T_1, \ldots T_n]$ is exactly

$$rank(\mathbf{u})_T = rank\left(u_1 T_1 + \cdots u_n T_n\right)$$

Distinguishing attack

the rank of a point :

We say the rank of $\mathbf{u} = (u_1, \ldots u_n)$ in $T = [T_1, \ldots T_n]$ is exactly

$$rank(\mathbf{u})_T = rank\left(u_1 T_1 + \cdots u_n T_n\right)$$

We will be concerned with points of rank 0

Distinguishing attack

the rank of a point :

We say the rank of $\mathbf{u} = (u_1, \ldots u_n)$ in $T = [T_1, \ldots T_n]$ is exactly

$$rank(\mathbf{u})_T = rank\left(u_1T_1 + \cdots u_nT_n\right)$$

We will be concerned with points of rank 0

i.e. points $\mathbf{u}$ such that $rank\left(u_1T_1 + \cdots u_nT_n\right) = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$

Distinguishing attack

Distinguishing attack

Recall that

$$t_0 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Distinguishing attack

Recall that

$$t_0 = \begin{bmatrix}1 & 0 & 0\\0 & 0 & 0\\0 & 0 & 0\end{bmatrix}, \begin{bmatrix}0 & 0 & 0\\0 & 1 & 0\\0 & 0 & 0\end{bmatrix}, \begin{bmatrix}0 & 0 & 0\\0 & 0 & 0\\0 & 0 & 1\end{bmatrix}$$

So which **u** are such that

$$rank(\mathbf{u})_{t_0} = \begin{bmatrix}u_1 & 0 & 0\\0 & u_2 & 0\\0 & 0 & u_3\end{bmatrix} = \begin{bmatrix}0 & 0 & 0\\0 & 0 & 0\\0 & 0 & 0\end{bmatrix}?$$

Distinguishing attack

Recall that

$$t_0 = \begin{bmatrix}1 & 0 & 0\\0 & 0 & 0\\0 & 0 & 0\end{bmatrix}, \begin{bmatrix}0 & 0 & 0\\0 & 1 & 0\\0 & 0 & 0\end{bmatrix}, \begin{bmatrix}0 & 0 & 0\\0 & 0 & 0\\0 & 0 & 1\end{bmatrix}$$

So which **u** are such that

$$rank(\mathbf{u})_{t_0} = \begin{bmatrix}u_1 & 0 & 0\\0 & u_2 & 0\\0 & 0 & u_3\end{bmatrix} = \begin{bmatrix}0 & 0 & 0\\0 & 0 & 0\\0 & 0 & 0\end{bmatrix}?$$

t_0 only has the trivial rank 0 point : $\mathbf{u} = (0,0,0)$

Distinguishing attack

Distinguishing attack

$$t_1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$rank(\mathbf{u})_{t_1} = \begin{bmatrix} u_1 & 0 & 0 \\ 0 & u_2 & 0 \\ 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Distinguishing attack

$$t_1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$rank(\mathbf{u})_{t_1} = \begin{bmatrix} u_1 & 0 & 0 \\ 0 & u_2 & 0 \\ 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

t_1 has one (non-trivial) rank 0 point : scalar multiples of $\mathbf{u} = (0,0,1)$

Distinguishing attack

Distinguishing attack

So t_1 has one rank 0 point and t_0 has no rank 0 points

Distinguishing attack

So t_1 has one rank 0 point and t_0 has no rank 0 points

Lemma : t_b and $(A, B, C) \star t_b$ have the same number of rank 0 points

Distinguishing attack

So t_1 has one rank 0 point and t_0 has no rank 0 points

Lemma : t_b and $(A, B, C) \star t_b$ have the same number of rank 0 points

Thus, given some commitment, $T = (A, B, C) \star t_b$,

if T has **no rank 0 points**, then $b = 0$

if T has **one rank 0 point** (up to scalar multiplication),
then $b = 1$

Distinguishing attack

So t_1 has one rank 0 point and t_0 has no rank 0 points

Lemma : t_b and $(A, B, C) \star t_b$ have the same number of rank 0 points

Thus, given some commitment, $T = (A, B, C) \star t_b$,

if T has **no rank 0 points**, then $b = 0$

if T has **one rank 0 point** (up to scalar multiplication),
then $b = 1$

Computing the rank 0 points in T requires solving n^2 linear equations in n variables

Distinguishing attack

So t_1 has one rank 0 point and t_0 has no rank 0 points

Lemma : t_b and $(A, B, C) \star t_b$ have the same number of rank 0 points

Thus, given some commitment, $T = (A, B, C) \star t_b$,

if T has **no rank 0 points**, then $b = 0$

if T has **one rank 0 point** (up to scalar multiplication), then $b = 1$

→ $O(n^4)$ complexity

Computing the rank 0 points in T requires solving n^2 linear equations in n variables

Distinguishing attack

Parameters

n	q
14	4093
22	4093
30	2039

No parameters were given in DFG.

These parameters were taken from MEDS.

Distinguishing attack

Parameters

n	q
14	4093
22	4093
30	2039

No parameters were given in DFG.

These parameters were taken from MEDS.

Distinguishing attack:

$\rightarrow$ Runtime < 1 second

Distinguishing attack

Distinguishing attack

This attack broke **hiding** and a special case of dTIP

Decisional Tensor Isomorphism Problem (dTIP):

Given **random** v_0, v_1 **decide** whether there exists

(A, B, C) such that $(A, B, C) \star v_0 = v_1$

Distinguishing attack

This attack broke **hiding** and a special case of dTIP

Decisional Tensor Isomorphism Problem (dTIP):

Given **random** v_0, v_1 **decide** whether there exists

(A, B, C) such that $(A, B, C) \star v_0 = v_1$

What about cTIP?

Computational Tensor Isomorphism Problem (cTIP):

Given **random** v_0, v_1 **compute**

(A, B, C) such that $(A, B, C) \star v_0 = v_1$

Computational attack

Computational attack

Suppose we have determined $b = 0$, let's recover (A, B, C) from $T = (A, B, C) \star t_0$

Computational attack

Suppose we have determined $b = 0$, let's recover (A, B, C) from $T = (A, B, C) \star t_0$

A first attempt : Gröbner basis?

Computational attack

Suppose we have determined $b = 0$, let's recover (A, B, C) from $T = (A, B, C) \star t_0$

A first attempt : Gröbner basis?

We can use a *Gröbner basis* to solve systems of multivariate polynomials

→ uses *Buchberger's algorithm*

→ manipulates the polynomials to eventually apply Gaussian elimination

Computational attack

Computational attack

$$T = (A, B, C) \star t_0$$

Computational attack

$$T = (A, B, C) \star t_0$$

$$T = \left(\begin{bmatrix} a_{1,1} & a_{1,2} & a_{1,3} \\ a_{2,1} & a_{2,2} & a_{2,3} \\ a_{3,1} & a_{3,2} & a_{3,3} \end{bmatrix}, \begin{bmatrix} b_{1,1} & b_{1,2} & b_{1,3} \\ b_{2,1} & b_{2,2} & b_{2,3} \\ b_{3,1} & b_{3,2} & b_{3,3} \end{bmatrix}, \begin{bmatrix} c_{1,1} & c_{1,2} & c_{1,3} \\ c_{2,1} & c_{2,2} & c_{2,3} \\ c_{3,1} & c_{3,2} & c_{3,3} \end{bmatrix} \right) \star t_0$$

Computational attack

$$T = (A, B, C) \star t_0$$

$$T = \left(\begin{bmatrix} a_{1,1} & a_{1,2} & a_{1,3} \\ a_{2,1} & a_{2,2} & a_{2,3} \\ a_{3,1} & a_{3,2} & a_{3,3} \end{bmatrix}, \begin{bmatrix} b_{1,1} & b_{1,2} & b_{1,3} \\ b_{2,1} & b_{2,2} & b_{2,3} \\ b_{3,1} & b_{3,2} & b_{3,3} \end{bmatrix}, \begin{bmatrix} c_{1,1} & c_{1,2} & c_{1,3} \\ c_{2,1} & c_{2,2} & c_{2,3} \\ c_{3,1} & c_{3,2} & c_{3,3} \end{bmatrix} \right) \star t_0$$

$\rightarrow 3n^2$ variables

Computational attack

$$T = (A, B, C) \star t_0$$

$$T = \left(\begin{bmatrix} a_{1,1} & a_{1,2} & a_{1,3} \\ a_{2,1} & a_{2,2} & a_{2,3} \\ a_{3,1} & a_{3,2} & a_{3,3} \end{bmatrix}, \begin{bmatrix} b_{1,1} & b_{1,2} & b_{1,3} \\ b_{2,1} & b_{2,2} & b_{2,3} \\ b_{3,1} & b_{3,2} & b_{3,3} \end{bmatrix}, \begin{bmatrix} c_{1,1} & c_{1,2} & c_{1,3} \\ c_{2,1} & c_{2,2} & c_{2,3} \\ c_{3,1} & c_{3,2} & c_{3,3} \end{bmatrix} \right) \star t_0$$

$\rightarrow 3n^2$ variables

$\rightarrow 3n^2$ equations

Computational attack

$$T = (A, B, C) \star t_0$$

$$T = \left(\begin{bmatrix} a_{1,1} & a_{1,2} & a_{1,3} \\ a_{2,1} & a_{2,2} & a_{2,3} \\ a_{3,1} & a_{3,2} & a_{3,3} \end{bmatrix}, \begin{bmatrix} b_{1,1} & b_{1,2} & b_{1,3} \\ b_{2,1} & b_{2,2} & b_{2,3} \\ b_{3,1} & b_{3,2} & b_{3,3} \end{bmatrix}, \begin{bmatrix} c_{1,1} & c_{1,2} & c_{1,3} \\ c_{2,1} & c_{2,2} & c_{2,3} \\ c_{3,1} & c_{3,2} & c_{3,3} \end{bmatrix} \right) \star t_0$$

$\rightarrow 3n^2$ variables

$\rightarrow 3n^2$ equations

$\rightarrow$ **cubic** equations

Computational attack

$$T = (A, B, C) \star t_0$$

$$T = \left(\begin{bmatrix} a_{1,1} & a_{1,2} & a_{1,3} \\ a_{2,1} & a_{2,2} & a_{2,3} \\ a_{3,1} & a_{3,2} & a_{3,3} \end{bmatrix}, \begin{bmatrix} b_{1,1} & b_{1,2} & b_{1,3} \\ b_{2,1} & b_{2,2} & b_{2,3} \\ b_{3,1} & b_{3,2} & b_{3,3} \end{bmatrix}, \begin{bmatrix} c_{1,1} & c_{1,2} & c_{1,3} \\ c_{2,1} & c_{2,2} & c_{2,3} \\ c_{3,1} & c_{3,2} & c_{3,3} \end{bmatrix} \right) \star t_0$$

$\rightarrow 3n^2$ variables

$\rightarrow 3n^2$ equations

$\rightarrow$ **cubic** equations

Upon first try, our instance has too many solutions…

Computational attack

Computational attack

How can we reduce the number of solutions?

Computational attack

How can we reduce the number of solutions?

The *stabilizer group* of t_0 is matrix triples (M_1, M_2, M_3) such that

$$(M_1, M_2, M_3) \star t_0 = t_0$$

Computational attack

How can we reduce the number of solutions?

The *stabilizer group* of t_0 is matrix triples (M_1, M_2, M_3) such that

$$(M_1, M_2, M_3) \star t_0 = t_0$$

Some examples of stabilizer elements include :

Computational attack

How can we reduce the number of solutions?

The *stabilizer group* of t_0 is matrix triples (M_1, M_2, M_3) such that

$$(M_1, M_2, M_3) \star t_0 = t_0$$

Some examples of stabilizer elements include :

$$\begin{bmatrix} \lambda_a & 0 & 0 \\ 0 & \lambda_a & 0 \\ 0 & 0 & \lambda_a \end{bmatrix} \cdot \begin{bmatrix} \lambda_b & 0 & 0 \\ 0 & \lambda_b & 0 \\ 0 & 0 & \lambda_b \end{bmatrix} \cdot \begin{bmatrix} \lambda_c & 0 & 0 \\ 0 & \lambda_c & 0 \\ 0 & 0 & \lambda_c \end{bmatrix} = I$$

Computational attack

How can we reduce the number of solutions?

The *stabilizer group* of t_0 is matrix triples (M_1, M_2, M_3) such that

$$(M_1, M_2, M_3) \star t_0 = t_0$$

Some examples of stabilizer elements include :

$$\begin{bmatrix} \lambda_a & 0 & 0 \\ 0 & \lambda_a & 0 \\ 0 & 0 & \lambda_a \end{bmatrix} \cdot \begin{bmatrix} \lambda_b & 0 & 0 \\ 0 & \lambda_b & 0 \\ 0 & 0 & \lambda_b \end{bmatrix} \cdot \begin{bmatrix} \lambda_c & 0 & 0 \\ 0 & \lambda_c & 0 \\ 0 & 0 & \lambda_c \end{bmatrix} = I$$

any permutation matrices (P, P, P),

e.g. $\begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$

Computational attack

Computational attack

By identifying stabilizer elements, we were able to filter out possibilities for (A, B, C)

Computational attack

By identifying stabilizer elements, we were able to filter out possibilities for (A, B, C)

Note, t_0 has exactly n rank 1 points, and thus so does $T = (A, B, C) \star t_0$

Computational attack

By identifying stabilizer elements, we were able to filter out possibilities for (A, B, C)

Note, t_0 has exactly n rank 1 points, and thus so does $T = (A, B, C) \star t_0$

Lemma : the rows of A^{-1} are all rank 1 points

Computational attack

By identifying stabilizer elements, we were able to filter out possibilities for (A, B, C)

Note, t_0 has exactly n rank 1 points, and thus so does $T = (A, B, C) \star t_0$

Lemma : the rows of A^{-1} are all rank 1 points

To find candidates for B and C we can solve

the (linear) set of equations given by

$$(I, B, I) \star t_0 = (A^{-1}, I, C^{-1}) \star T$$

Computational attack

By identifying stabilizer elements, we were able to filter out possibilities for (A, B, C)

Note, t_0 has exactly n rank 1 points, and thus so does $T = (A, B, C) \star t_0$

Lemma : the rows of A^{-1} are all rank 1 points

To find candidates for B and C we can solve

the (linear) set of equations given by

$$(I, B, I) \star t_0 = (A^{-1}, I, C^{-1}) \star T$$

➡ $O(n^6)$ complexity

Computational attack

Runtime for the attack:

n	q	Time (s)
14	4093	9.3
22	4093	141.6
30	2039	858.9

MinRank

MinRank

There was only 1 rank-0 matrix: $\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$

MinRank

There was only 1 rank-0 matrix: $\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$

Already for rank 1, there are too many matrices to check…

MinRank

There was only 1 rank-0 matrix: $\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$

Already for rank 1, there are too many matrices to check…

MinRank:

Given an integer $r \in \mathbb{N}$ and k matrices $M_1, \ldots M_k$,
find integers $x_1, \ldots x_k$ (not all zero) such that

$$rank\left(x_1 M_1 + \ldots + x_k M_k\right) \leq r$$

MinRank

We used the following work (*AsiaCrypt 2020*):

Improvements of Algebraic Attacks for solving the Rank Decoding and MinRank problems

Magali Bardet[4,5], Maxime Bros[1], Daniel Cabarcas[6], Philippe Gaborit[1], Ray Perlner[2], Daniel Smith-Tone[2,3], Jean-Pierre Tillich[4], and Javier Verbel[6]

[1] Univ. Limoges, CNRS, XLIM, UMR 7252, F-87000 Limoges, France
maxime.bros@unilim.fr
[2] National Institute of Standards and Technology, USA
[3] University of Louisville, USA
[4] Inria, 2 rue Simone Iff, 75012 Paris, France
[5] LITIS, University of Rouen Normandie, France
[6] Universidad Nacional de Colombia Sede Medellín, Medellín, Colombia

Abstract. In this paper, we show how to significantly improve algebraic techniques for solving the MinRank problem, which is ubiquitous in multivariate and rank metric code based cryptography. In the case of

MinRank

We used the following work (*AsiaCrypt 2020*):

Improvements of Algebraic Attacks for solving the Rank Decoding and MinRank problems

Magali Bardet[4,5], Maxime Bros[1], Daniel Cabarcas[6], Philippe Gaborit[1], Ray Perlner[2], Daniel Smith-Tone[2,3], Jean-Pierre Tillich[4], and Javier Verbel[6]

[1] Univ. Limoges, CNRS, XLIM, UMR 7252, F-87000 Limoges, France
maxime.bros@unilim.fr
[2] National Institute of Standards and Technology, USA
[3] University of Louisville, USA
[4] Inria, 2 rue Simone Iff, 75012 Paris, France
[5] LITIS, University of Rouen Normandie, France
[6] Universidad Nacional de Colombia Sede Medellín, Medellín, Colombia

Abstract. In this paper, we show how to significantly improve algebraic techniques for solving the MinRank problem, which is ubiquitous in multivariate and rank metric code based cryptography. In the case of

→ we were able to solve with **direct linearization**

MinRank

We used the following work (*AsiaCrypt 2020*):

Improvements of Algebraic Attacks for solving the Rank Decoding and MinRank problems

Magali Bardet[4,5], Maxime Bros[1], Daniel Cabarcas[6], Philippe Gaborit[1], Ray Perlner[2], Daniel Smith-Tone[2,3], Jean-Pierre Tillich[4], and Javier Verbel[6]

[1] Univ. Limoges, CNRS, XLIM, UMR 7252, F-87000 Limoges, France
maxime.bros@unilim.fr
[2] National Institute of Standards and Technology, USA
[3] University of Louisville, USA
[4] Inria, 2 rue Simone Iff, 75012 Paris, France
[5] LITIS, University of Rouen Normandie, France
[6] Universidad Nacional de Colombia Sede Medellín, Medellín, Colombia

Abstract. In this paper, we show how to significantly improve algebraic techniques for solving the MinRank problem, which is ubiquitous in multivariate and rank metric code based cryptography. In the case of

→ we were able to solve with **direct linearization**

→ for $r > 1$ the complexity quickly increases

Repair

Repair

What do we need?

Repair

What do we need?

→ tensors in different orbits

Repair

What do we need?

→ tensors in different orbits

→ no low rank points

Repair

What do we need?

→ tensors in different orbits

→ no low rank points

…**random** tensors!

Repair

What do we need?

→ tensors in different orbits

→ no low rank points

…**random** tensors!

v_0, v_1

$T = (A, B, C) \star v_b$

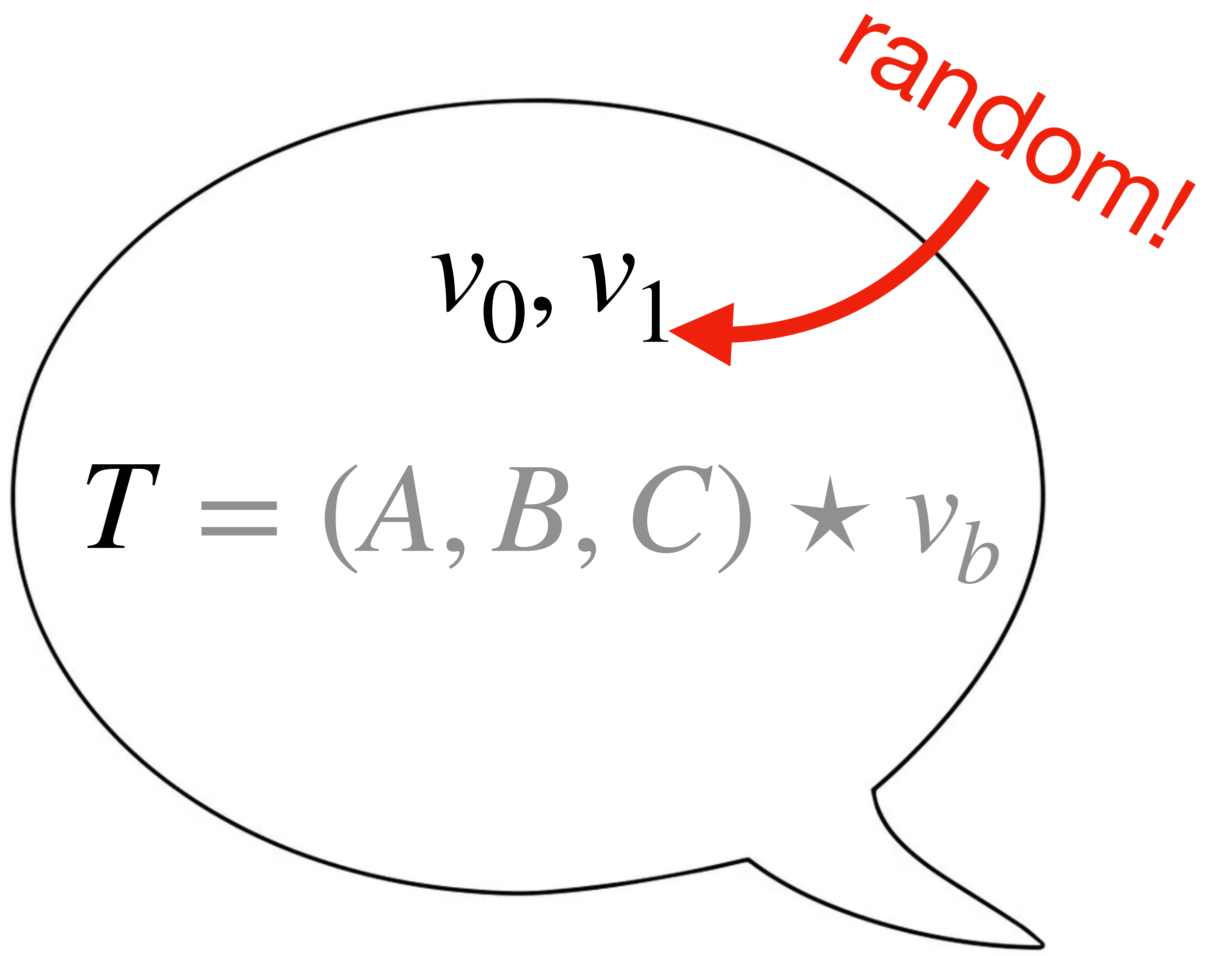
Repair

What do we need?

→ tensors in different orbits

→ no low rank points

…**random** tensors!



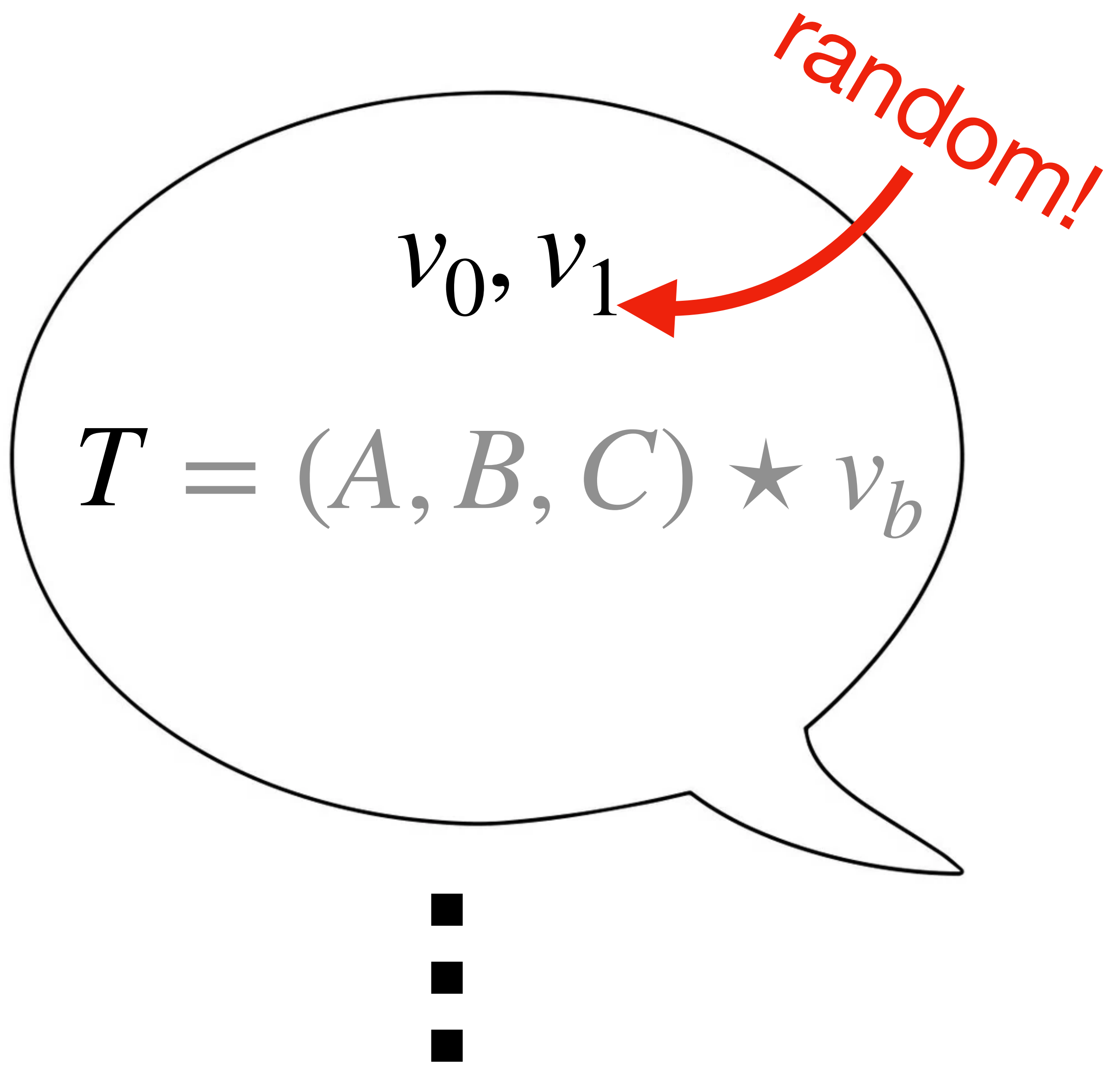
Repair

What do we need?

→ tensors in different orbits

→ no low rank points

…**random** tensors!



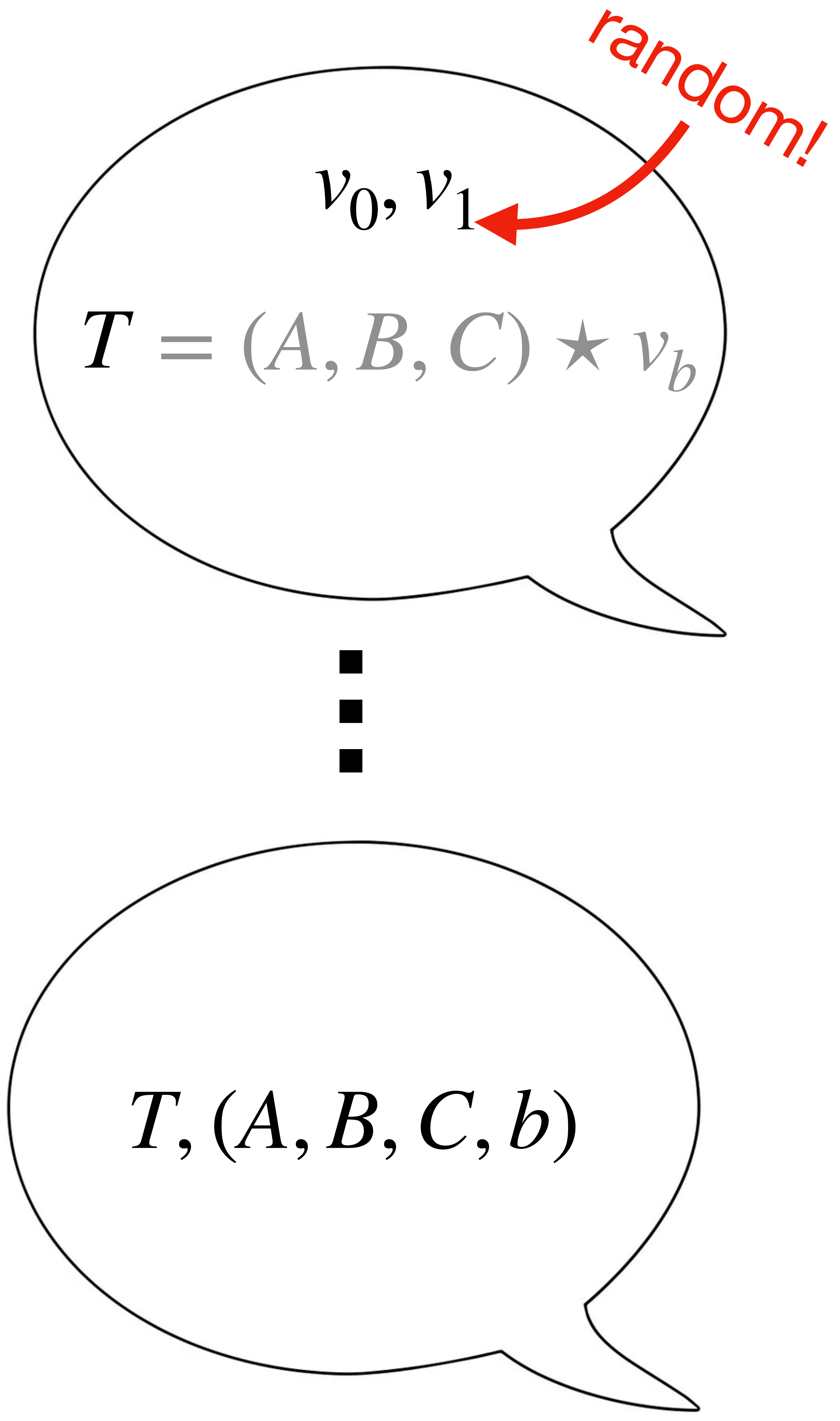
Repair

What do we need?

→ tensors in different orbits

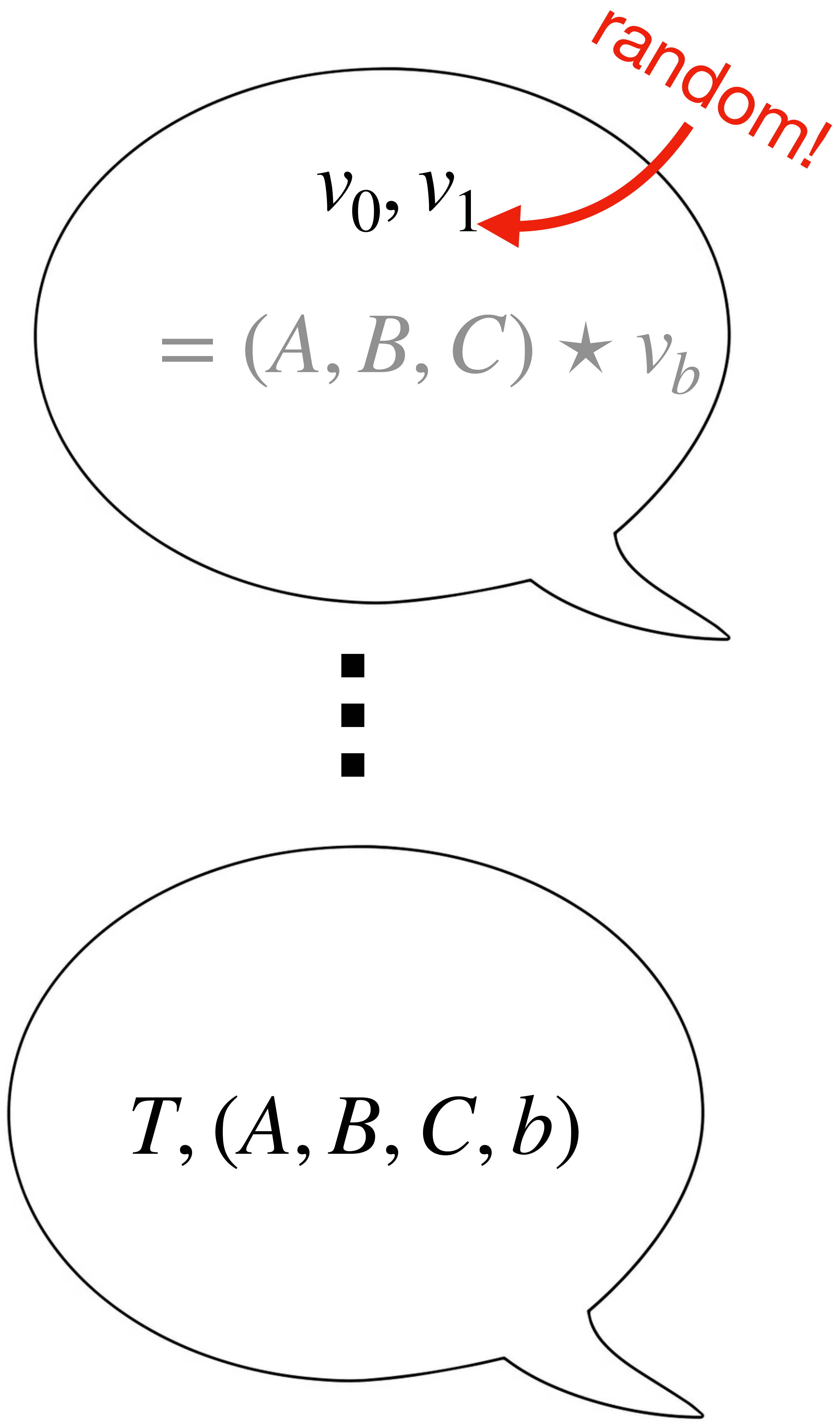
→ no low rank points

…**random** tensors!



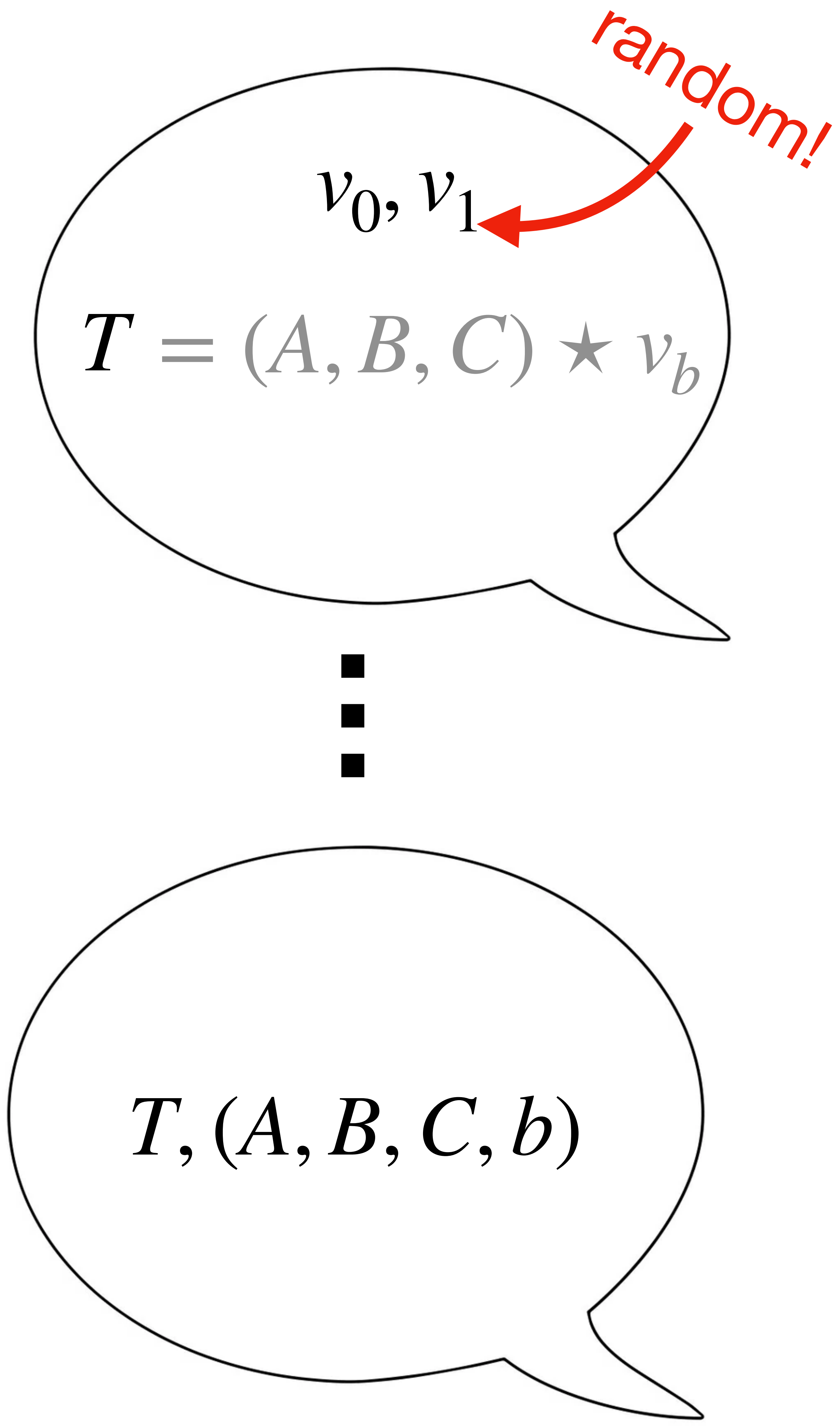
Repair

How can we ensure that v_0, v_1 are generated randomly?



Repair

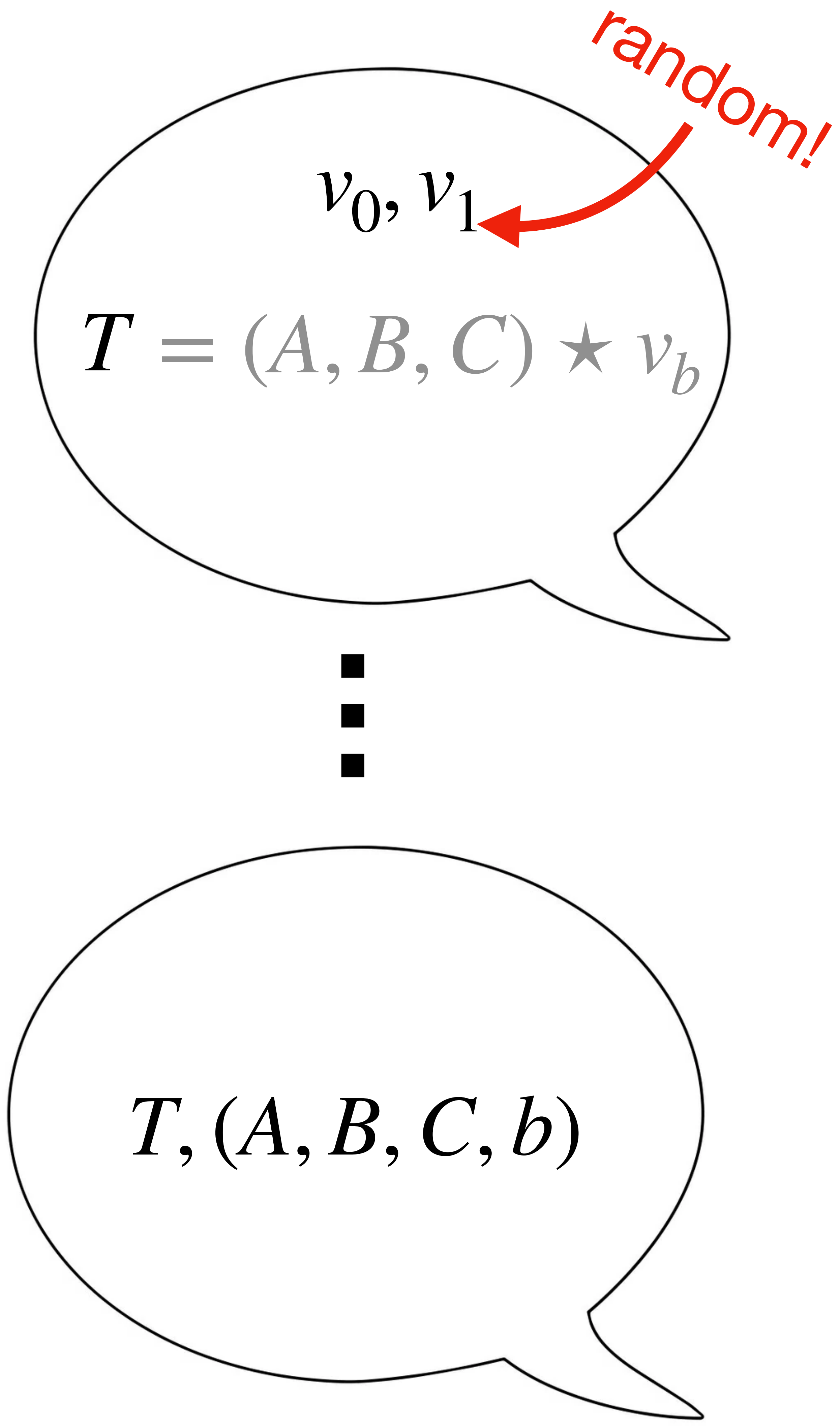
How can we ensure that v_0, v_1 are generated randomly?



Repair

How can we ensure that v_0, v_1 are generated randomly?

$\rightarrow$ pseudo-random number generator

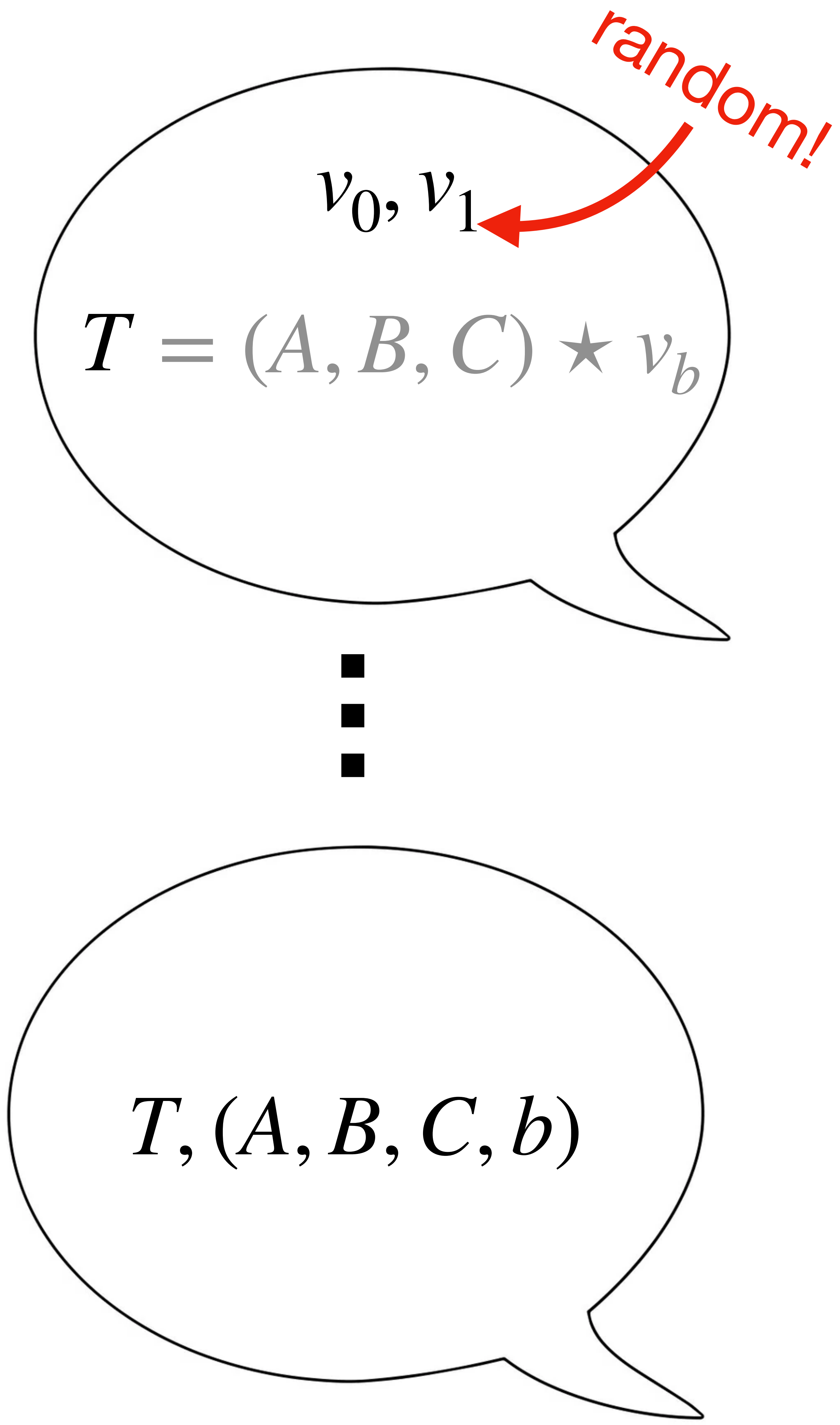


Repair

How can we ensure that v_0, v_1 are generated randomly?

→ pseudo-random number generator

→ cryptographic hash function



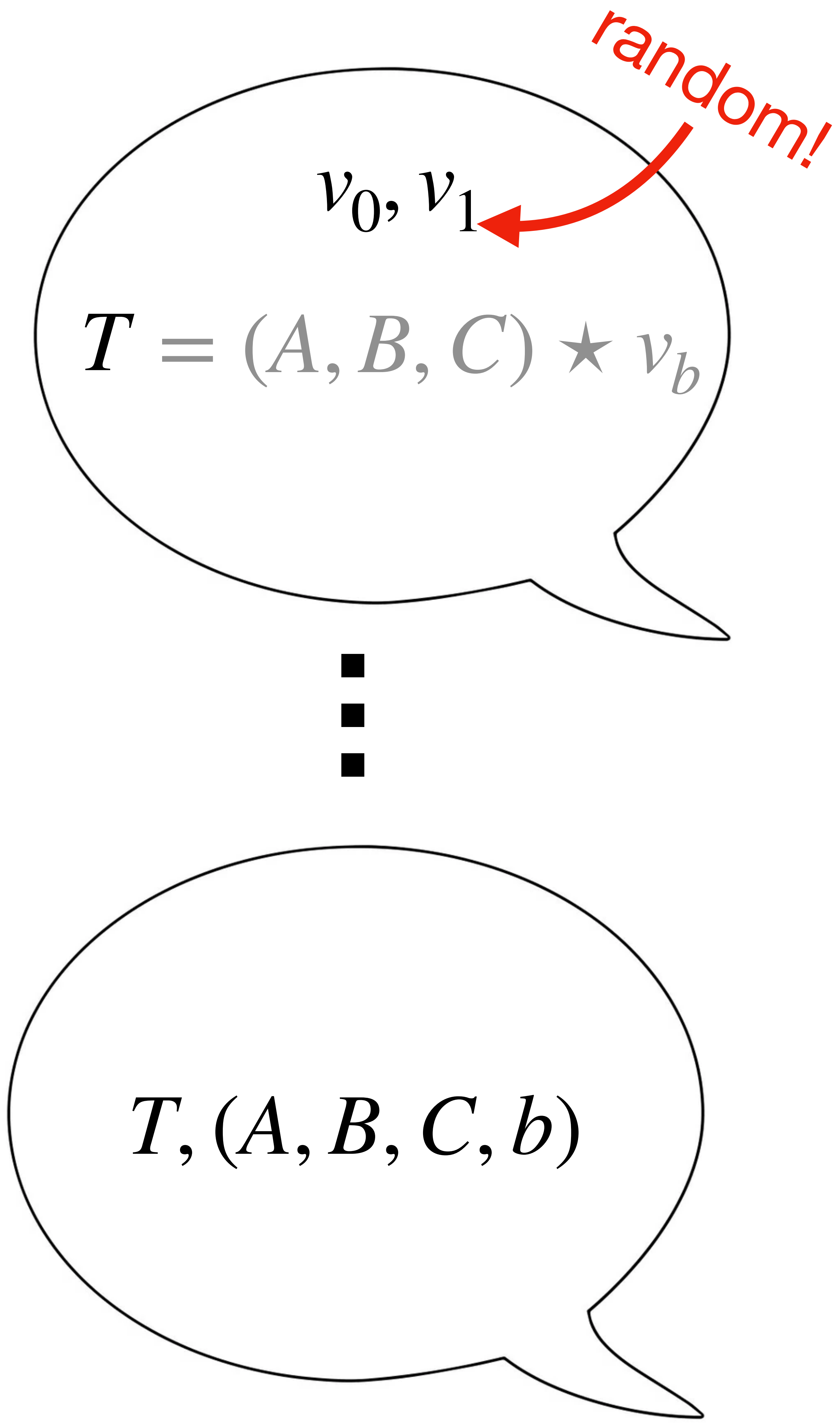
Repair

How can we ensure that v_0, v_1 are generated randomly?

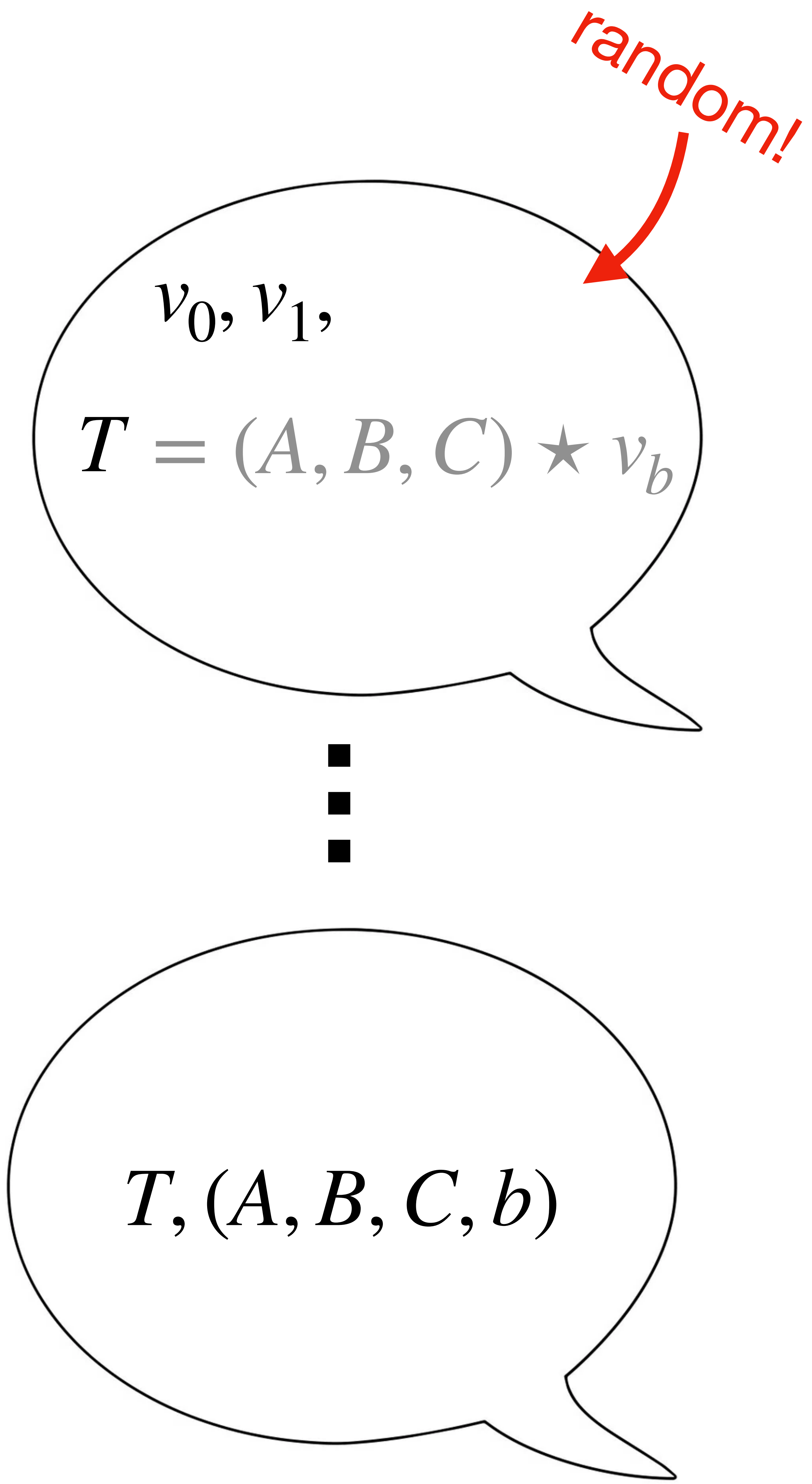
→ pseudo-random number generator

→ cryptographic hash function

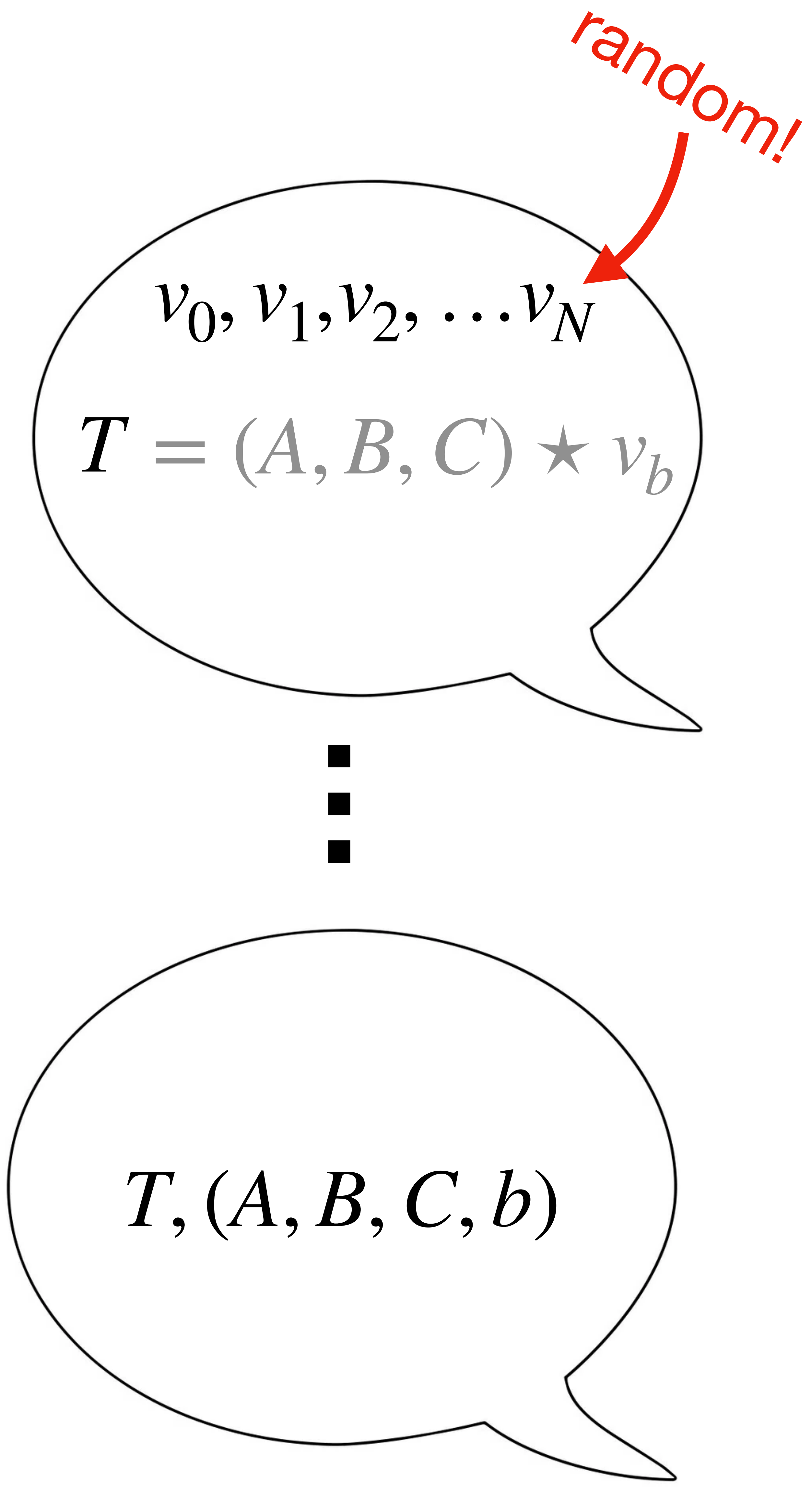
→ trusted third party



Repair

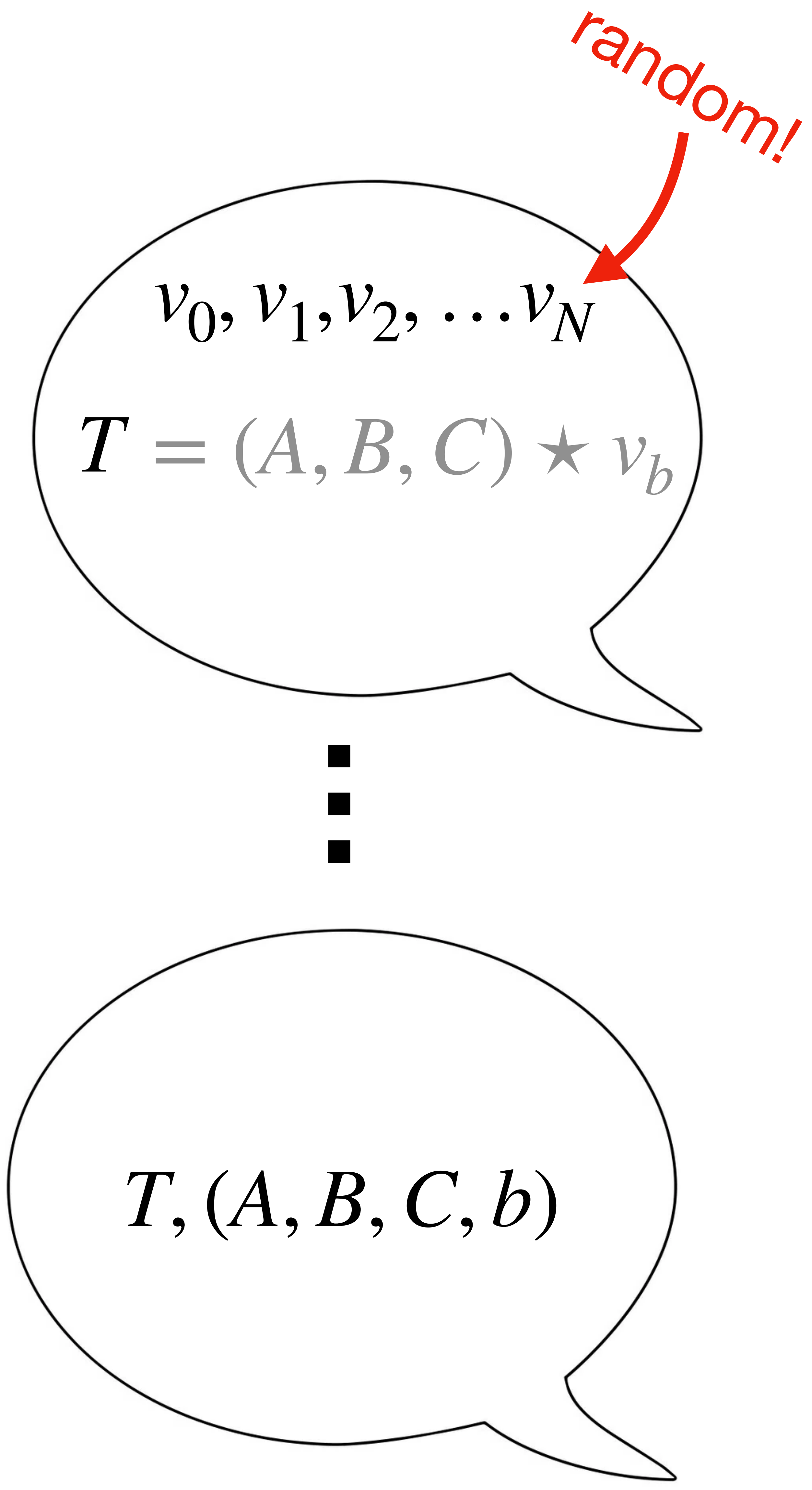


Repair



Repair

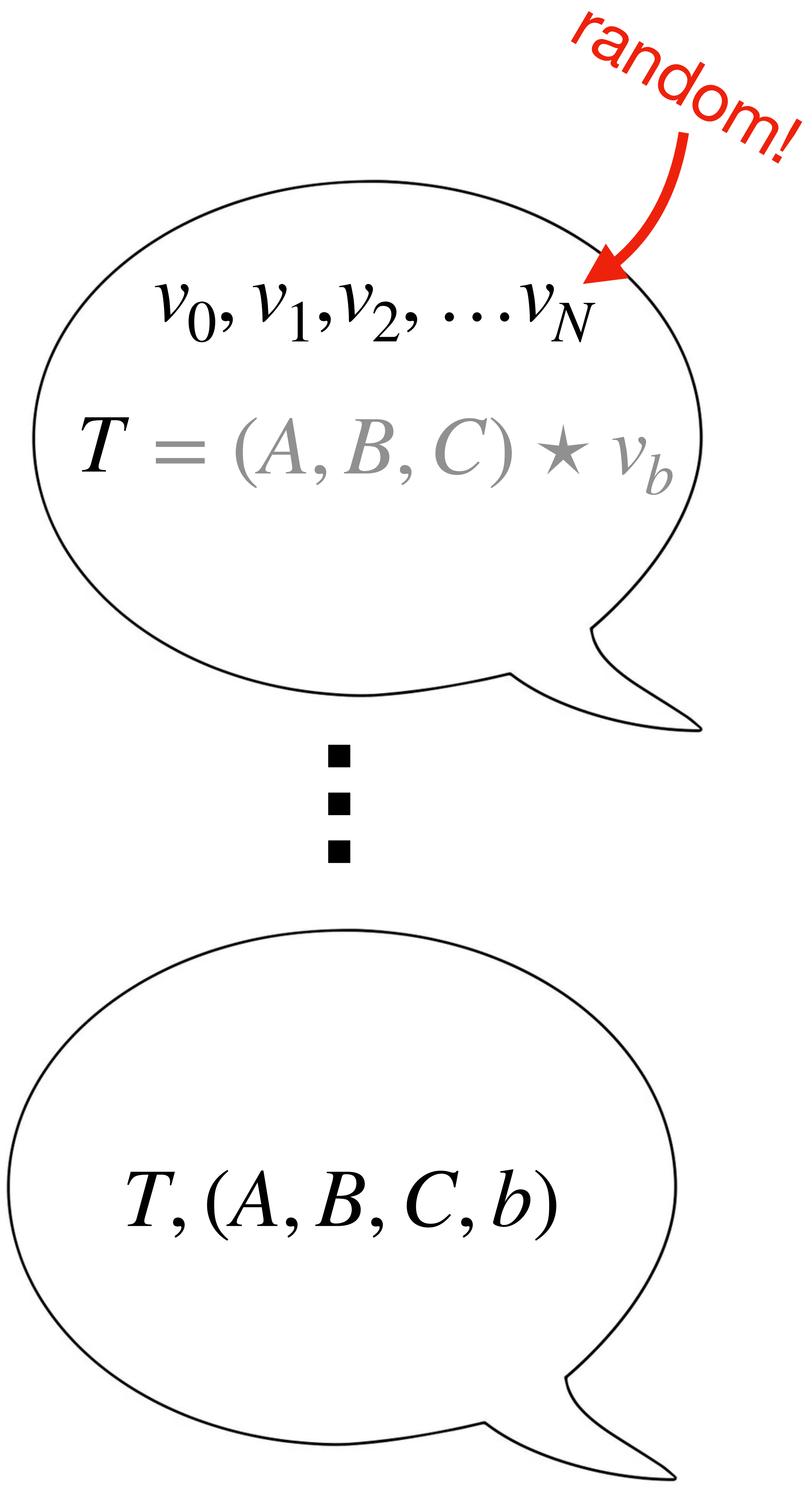
This gives a scheme that is



Repair

This gives a scheme that is

→ **statistically binding**

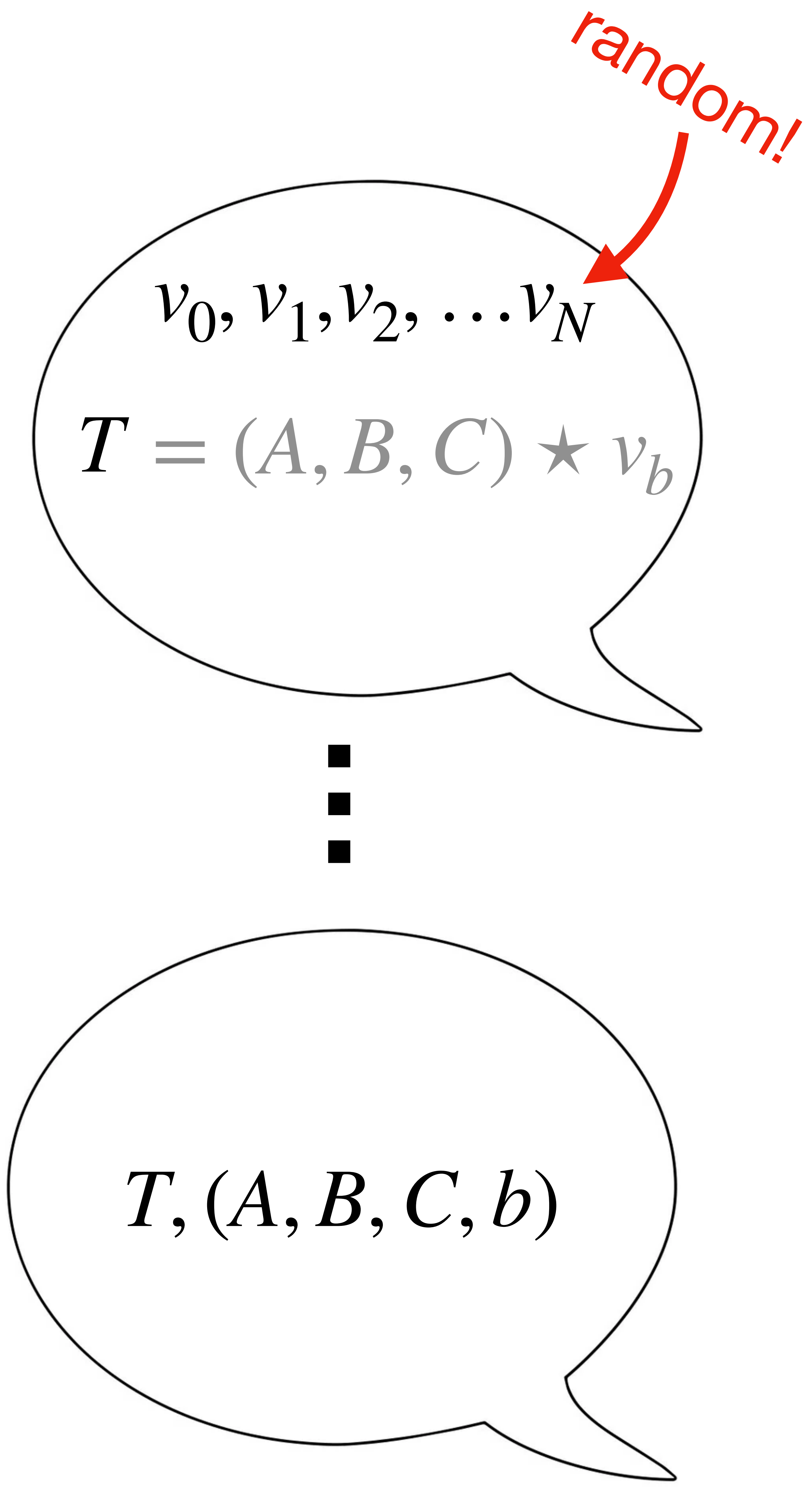


Repair

This gives a scheme that is

→ **statistically binding**

→ **computationally hiding**



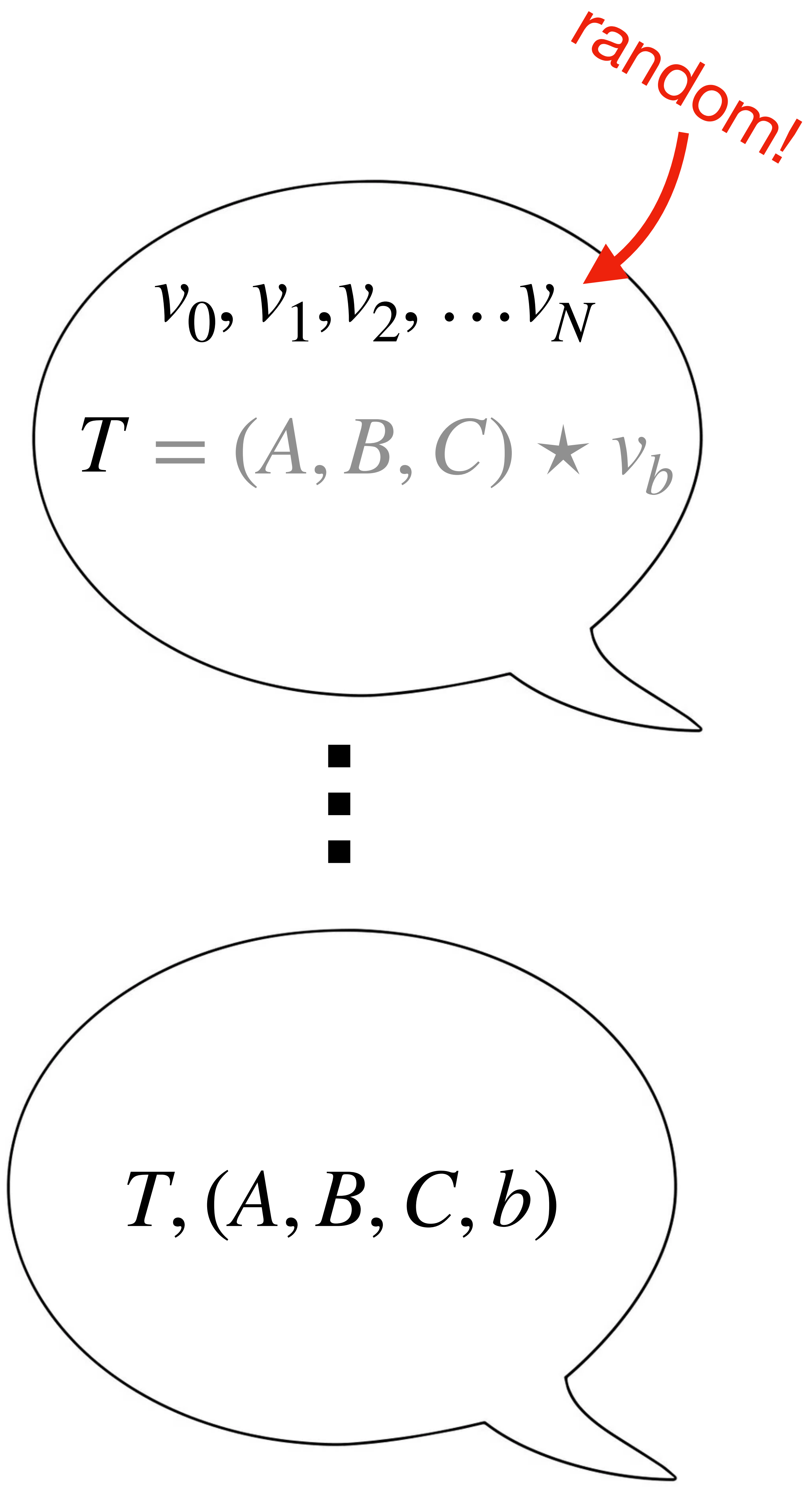
Repair

This gives a scheme that is

→ **statistically binding**

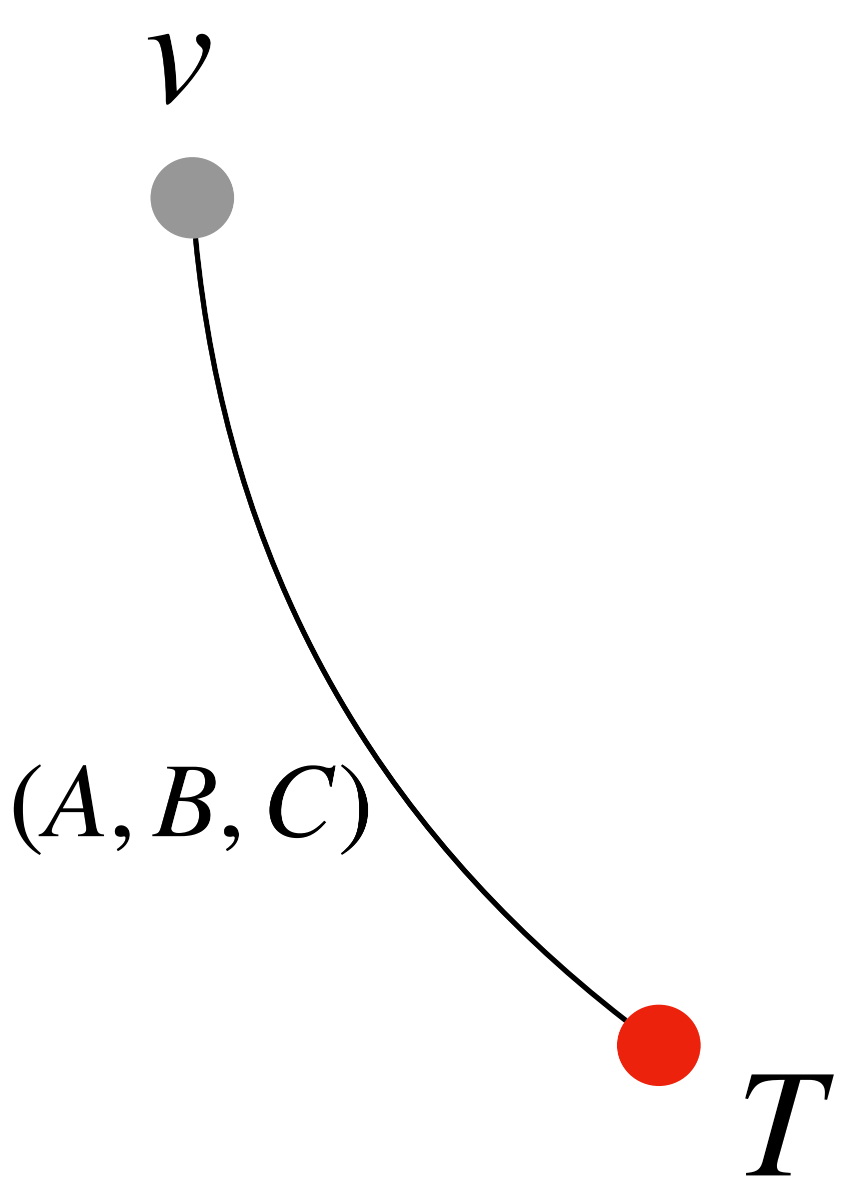
→ **computationally hiding**

→ complexity increases from $O(n^3)$ to $O(n^4)$



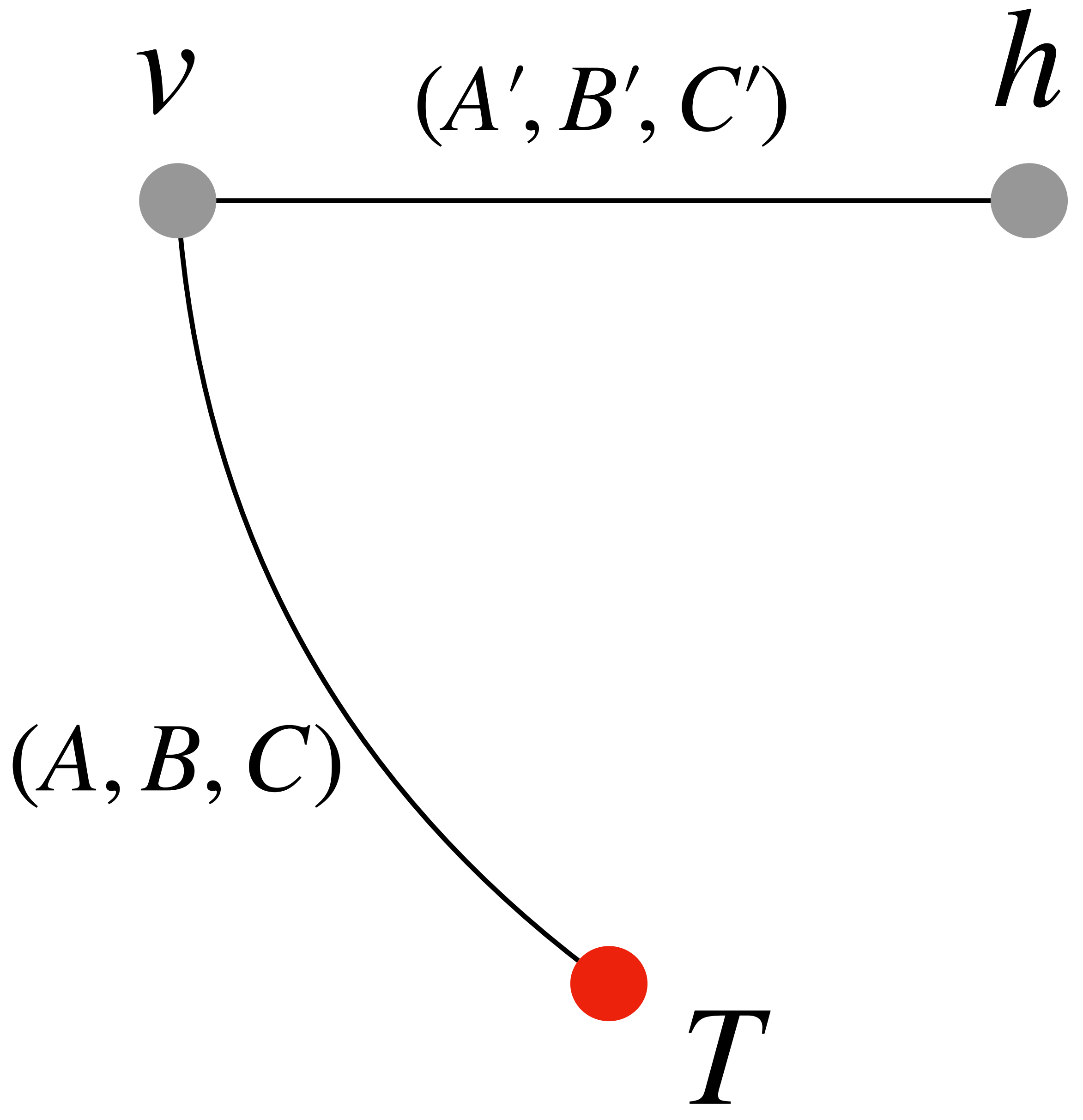
Proofs of knowledge

Sometimes we will want to prove that we know a committed value without revealing it



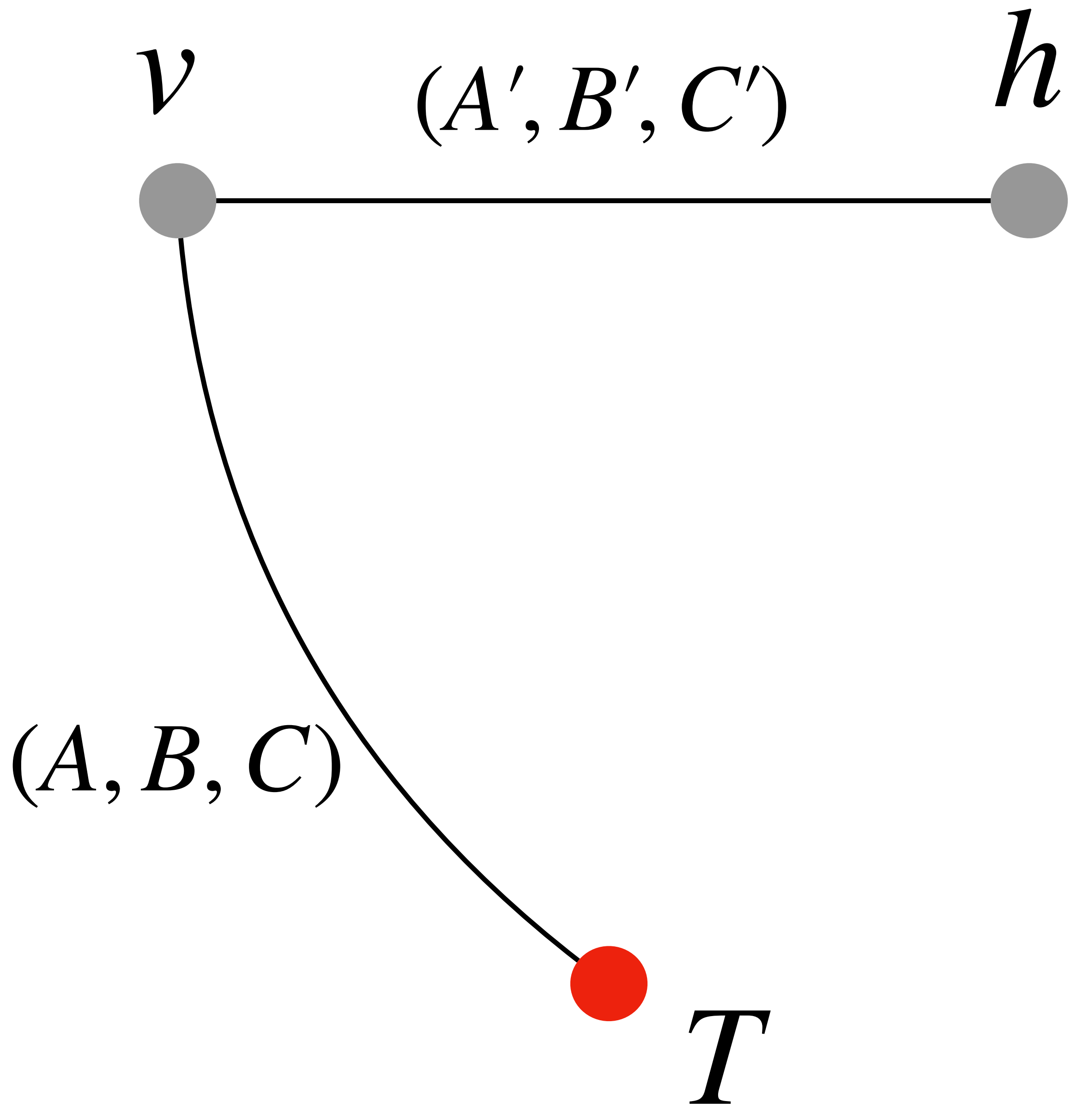
Proofs of knowledge

Sometimes we will want to prove that we know a committed value without revealing it



Proofs of knowledge

Sometimes we will want to prove that we know a committed value without revealing it

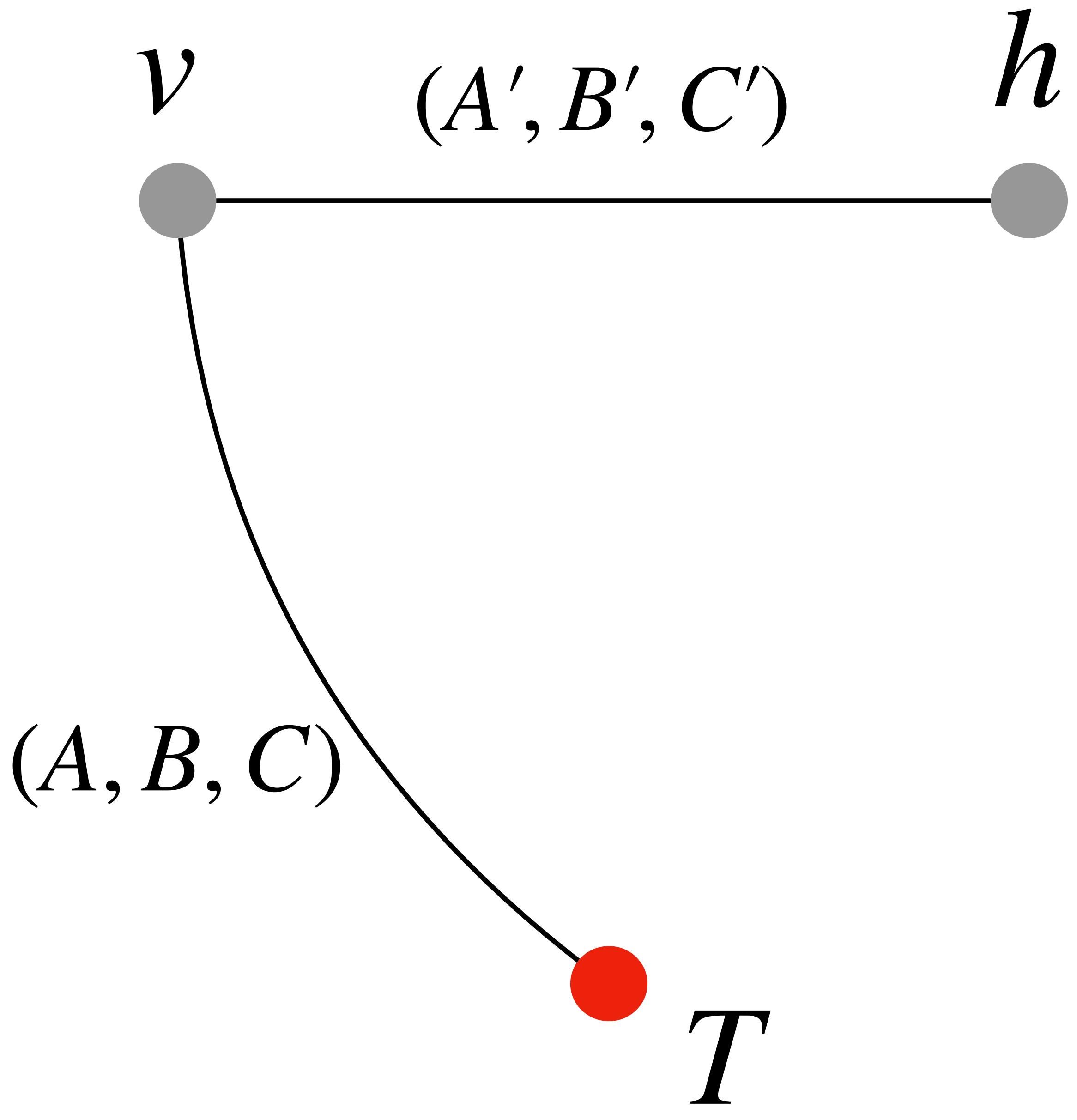


Prover:

Verifier:

Proofs of knowledge

Sometimes we will want to prove that we know a committed value without revealing it



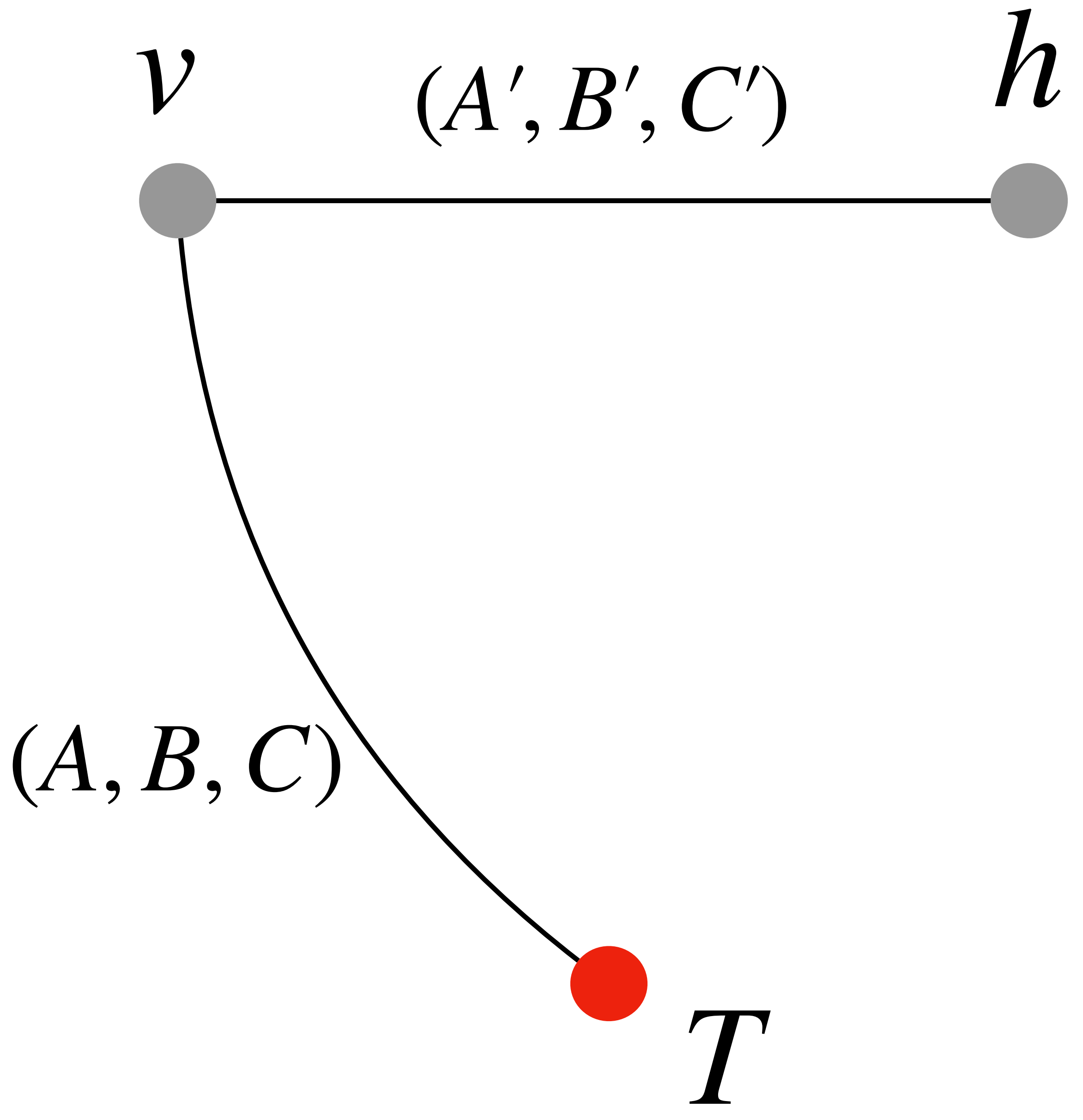
Prover:

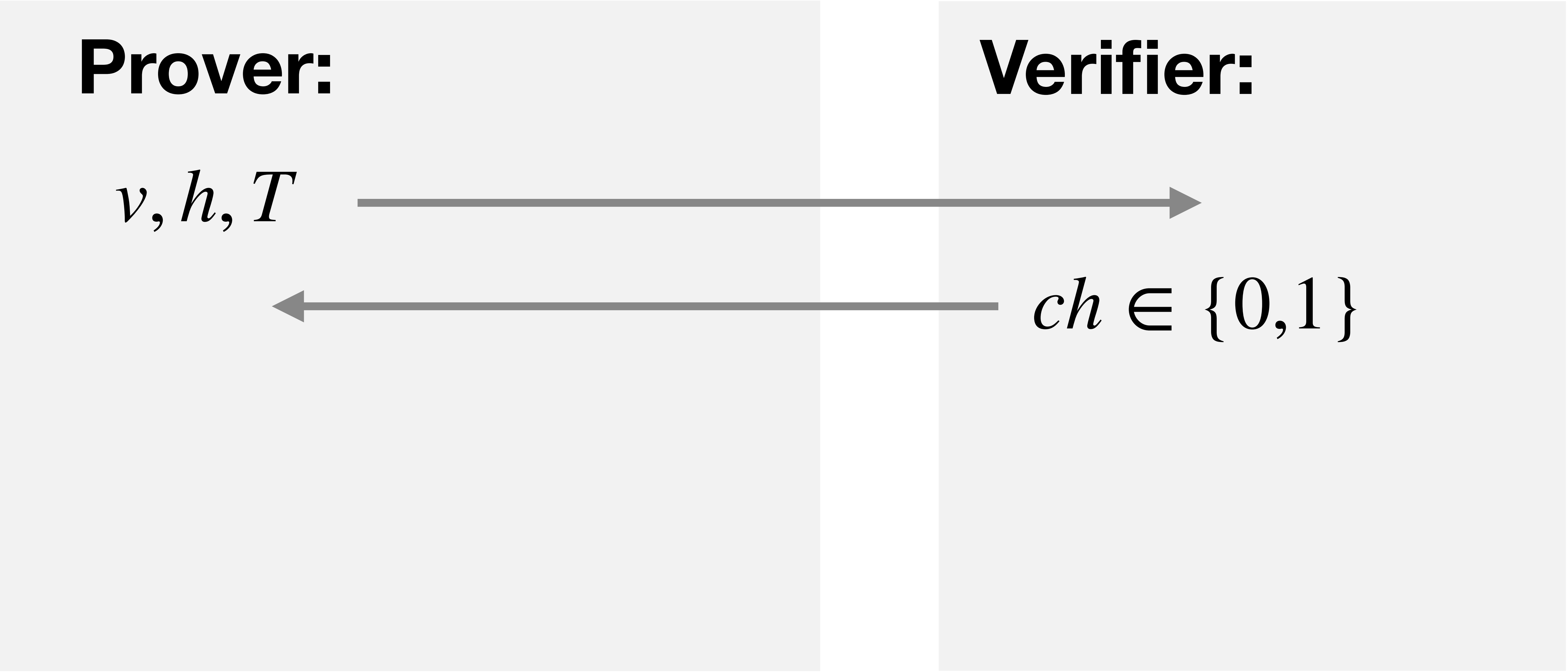
v, h, T →

Verifier:

Proofs of knowledge

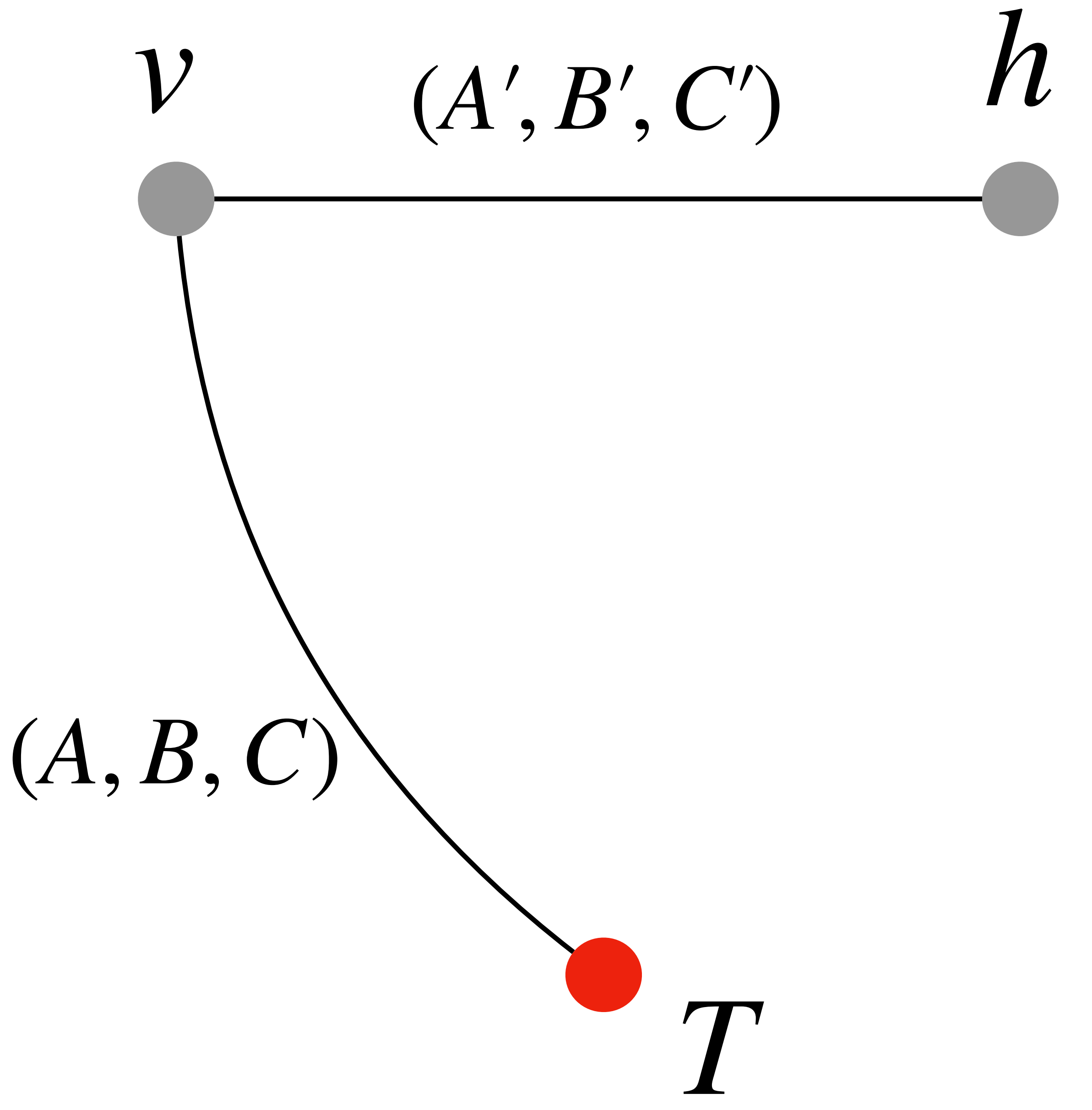
Sometimes we will want to prove that we know a committed value without revealing it





Proofs of knowledge

Sometimes we will want to prove that we know a committed value without revealing it



Prover:

Verifier:

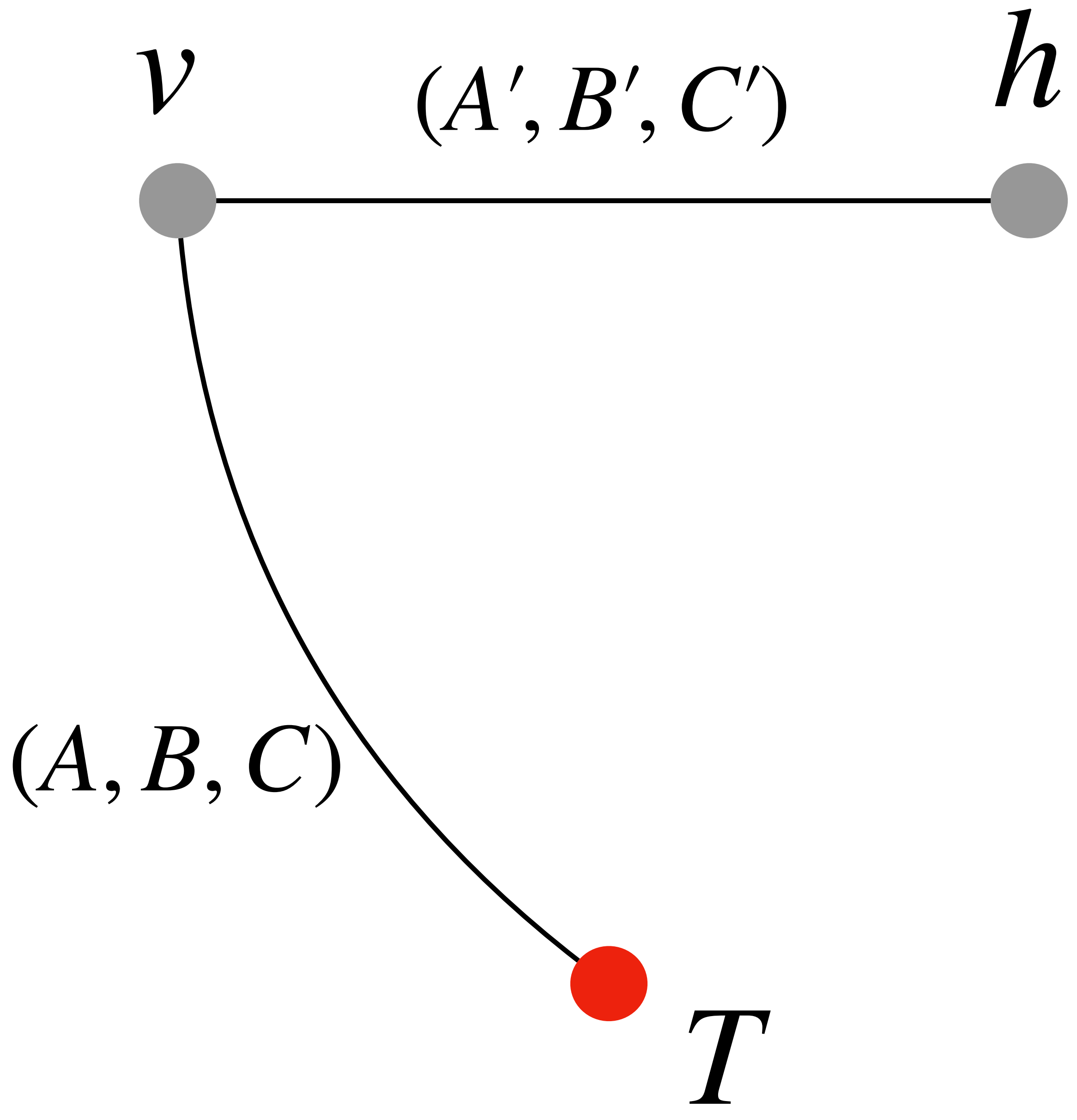
v, h, T →

← $ch \in \{0,1\}$

$$\begin{cases} (A', B', C'), & ch = 0 \\ (A'A, B'B, C'C), & ch = 1 \end{cases}$$ →

Proofs of knowledge

Sometimes we will want to prove that we know a committed value without revealing it



Prover:

Verifier:

v, h, T →

← $ch \in \{0,1\}$

$$\begin{cases} (A', B', C'), & ch = 0 \\ (A'A, B'B, C'C), & ch = 1 \end{cases}$$ →

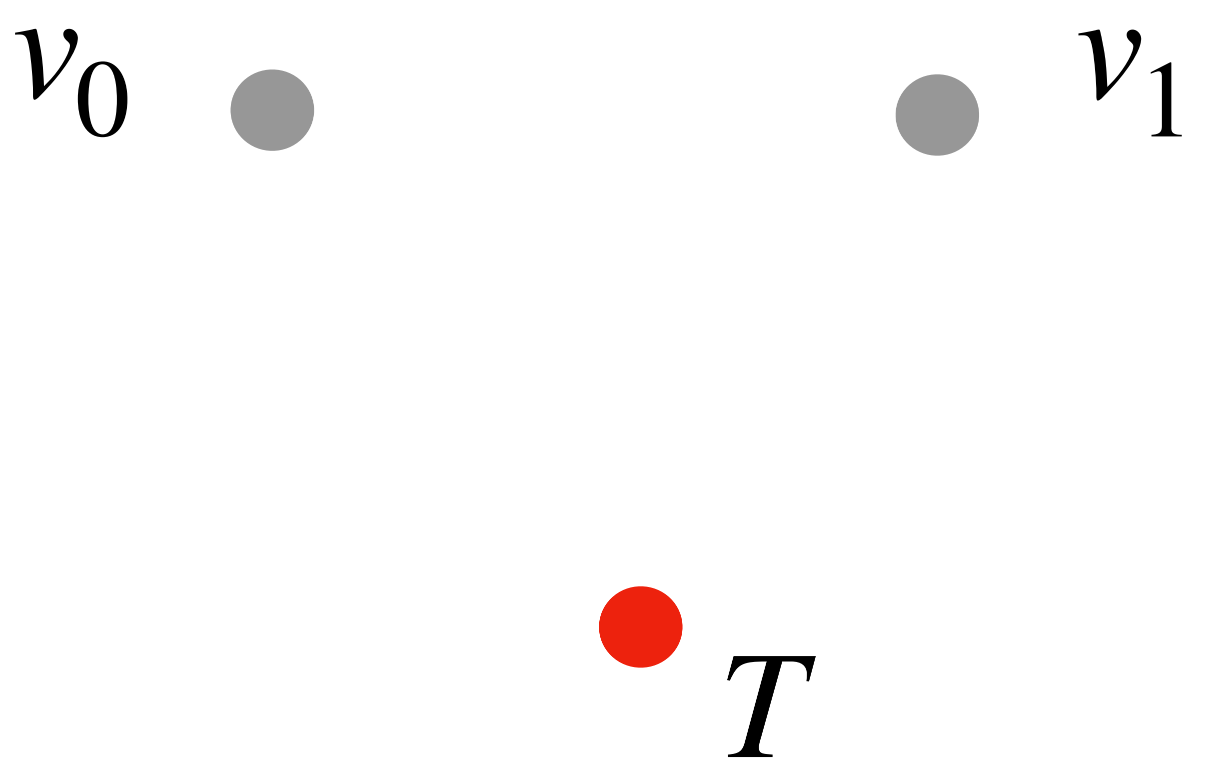
→ we need to keep v secret

Proofs of knowledge

(A, B, C, b) secret

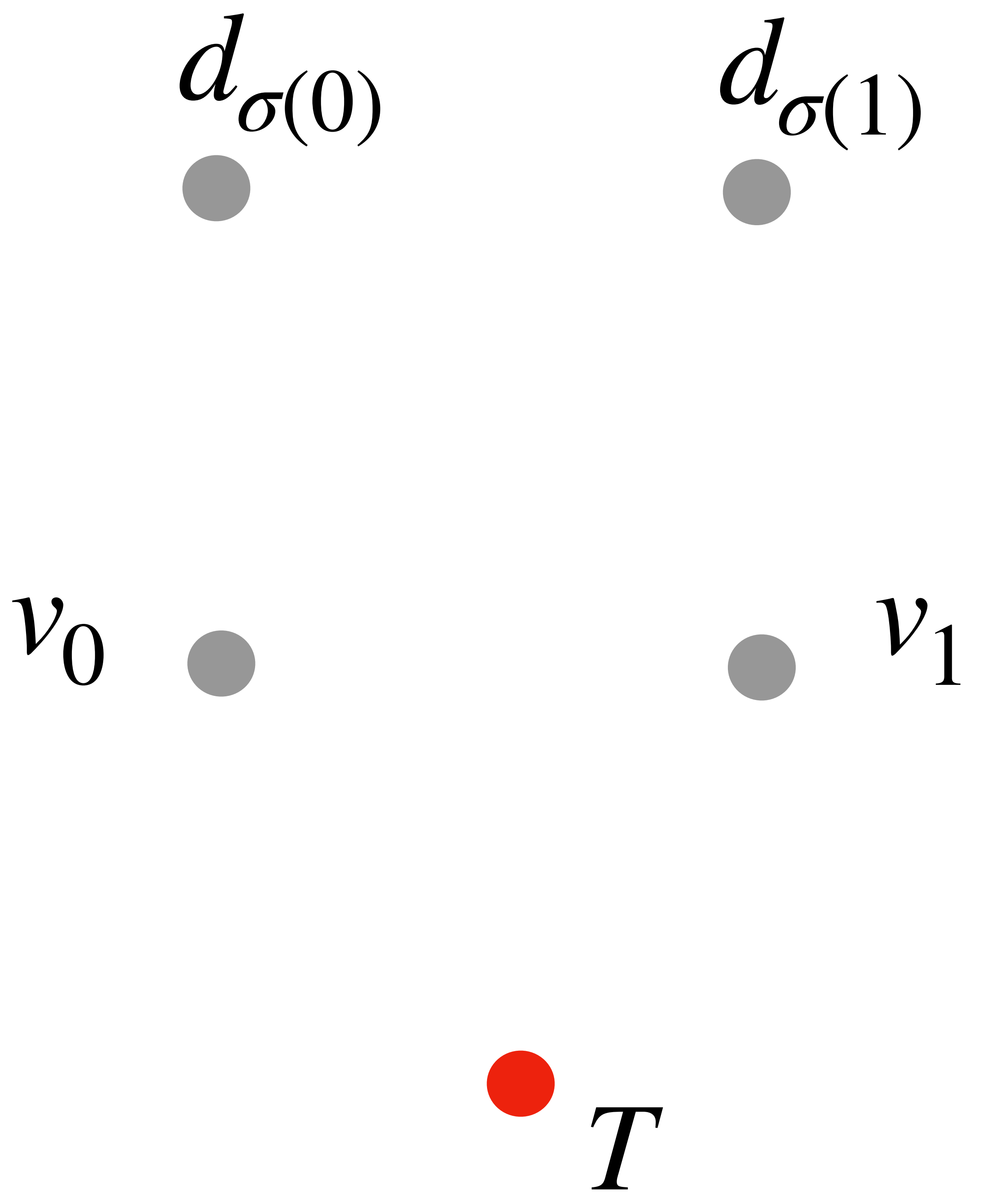
Proofs of knowledge

(A, B, C, b) secret

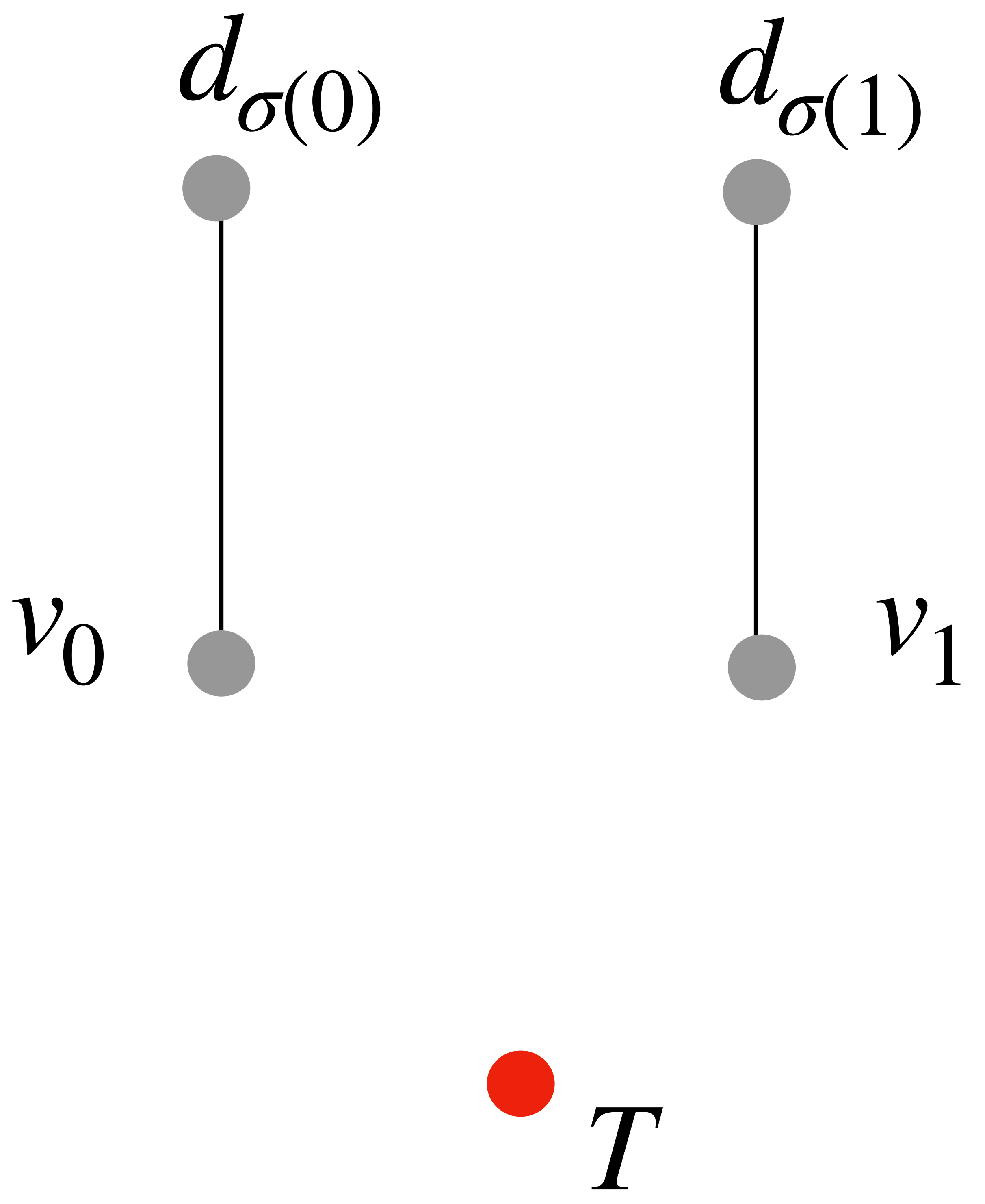


Proofs of knowledge

(A, B, C, b) secret

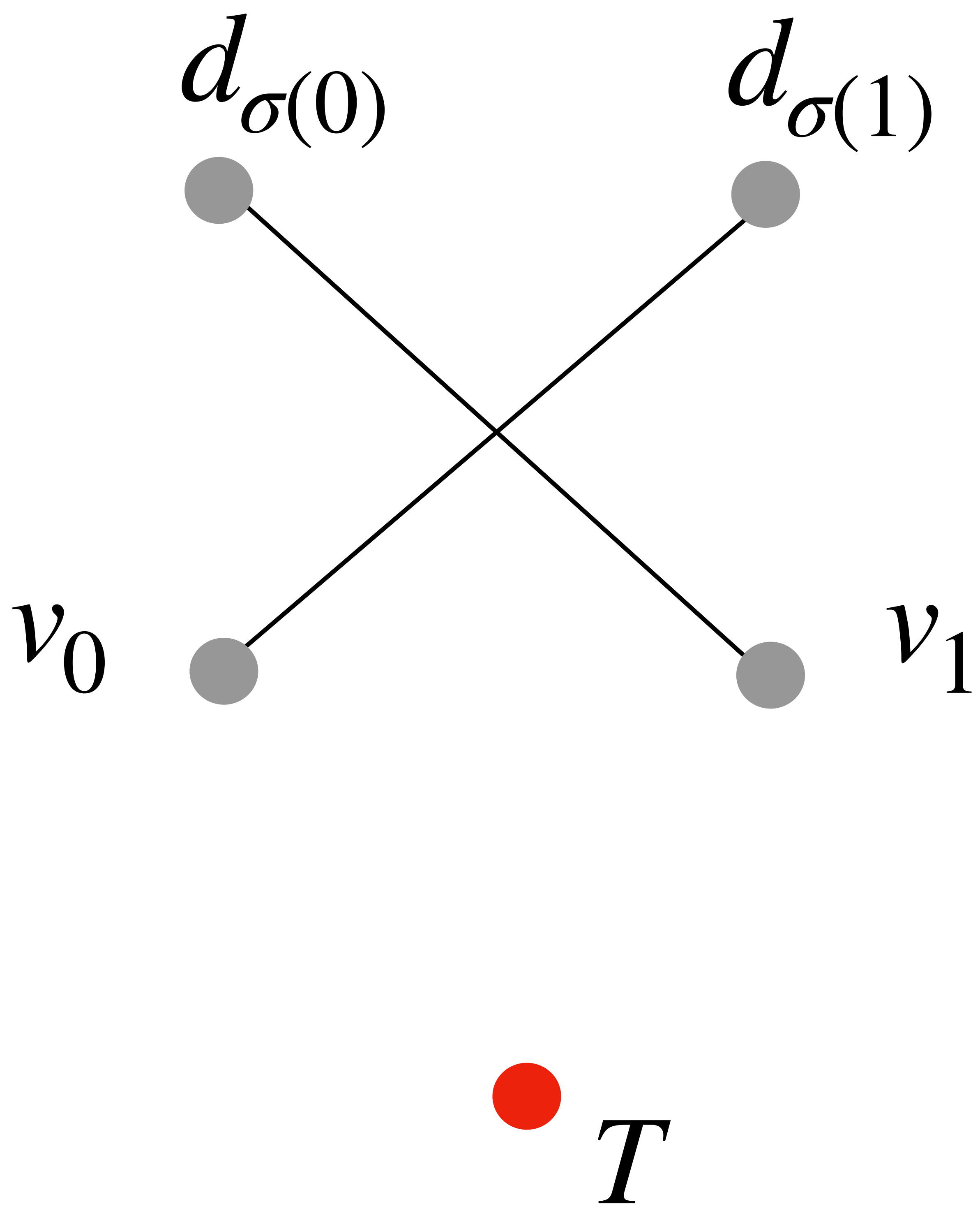


Proofs of knowledge



(A, B, C, b) secret

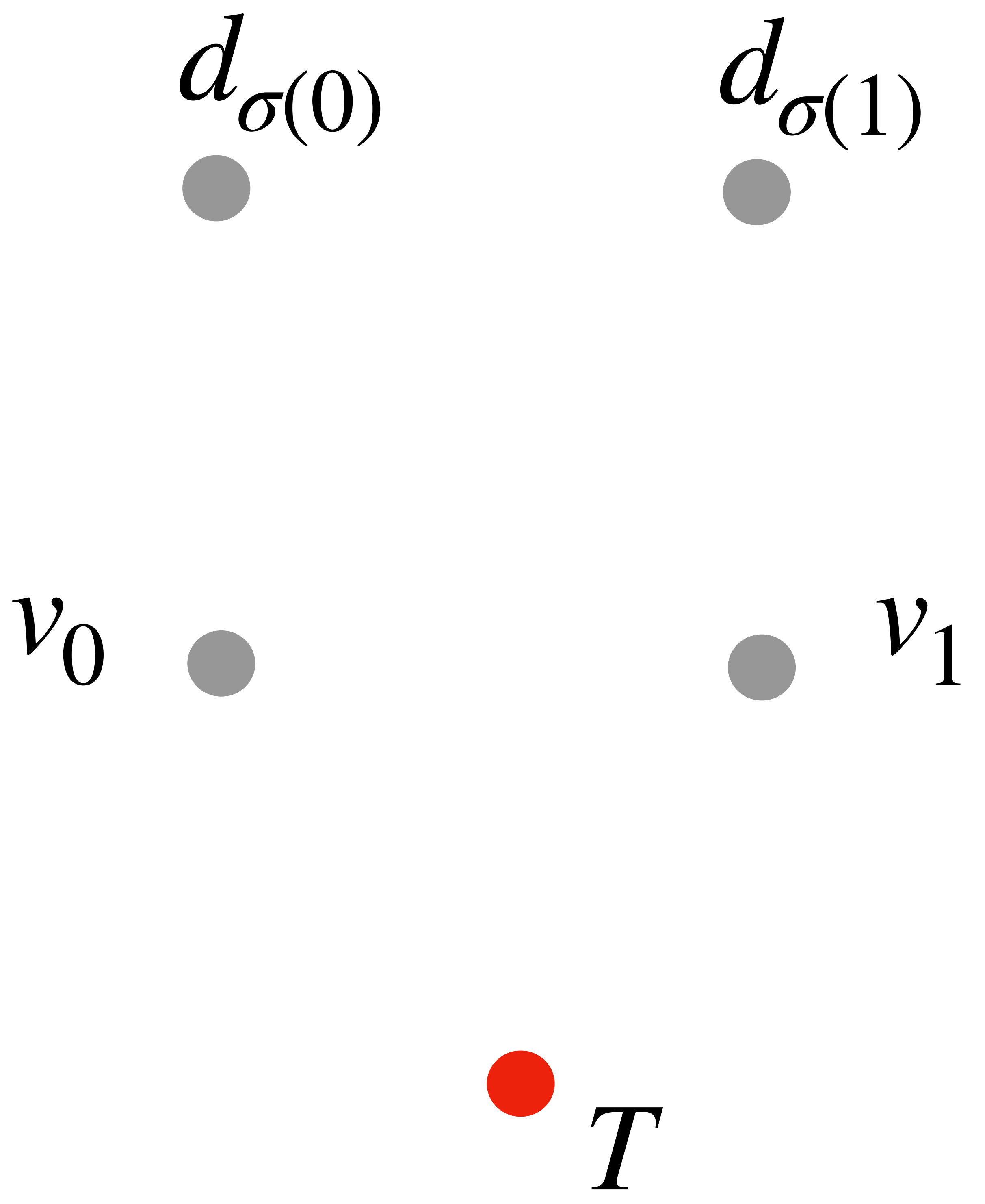
Proofs of knowledge



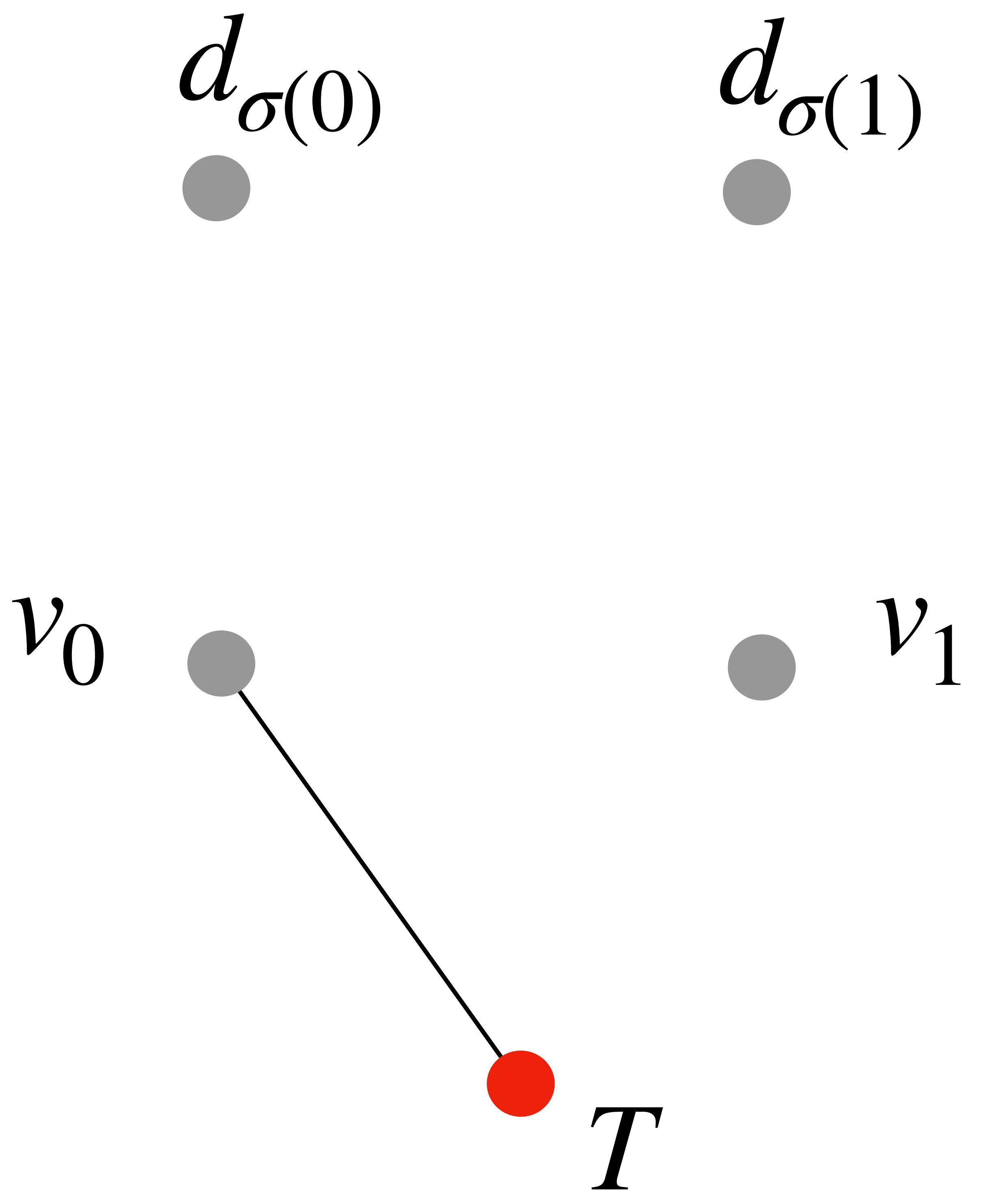
(A, B, C, b) secret

Proofs of knowledge

(A, B, C, b) secret

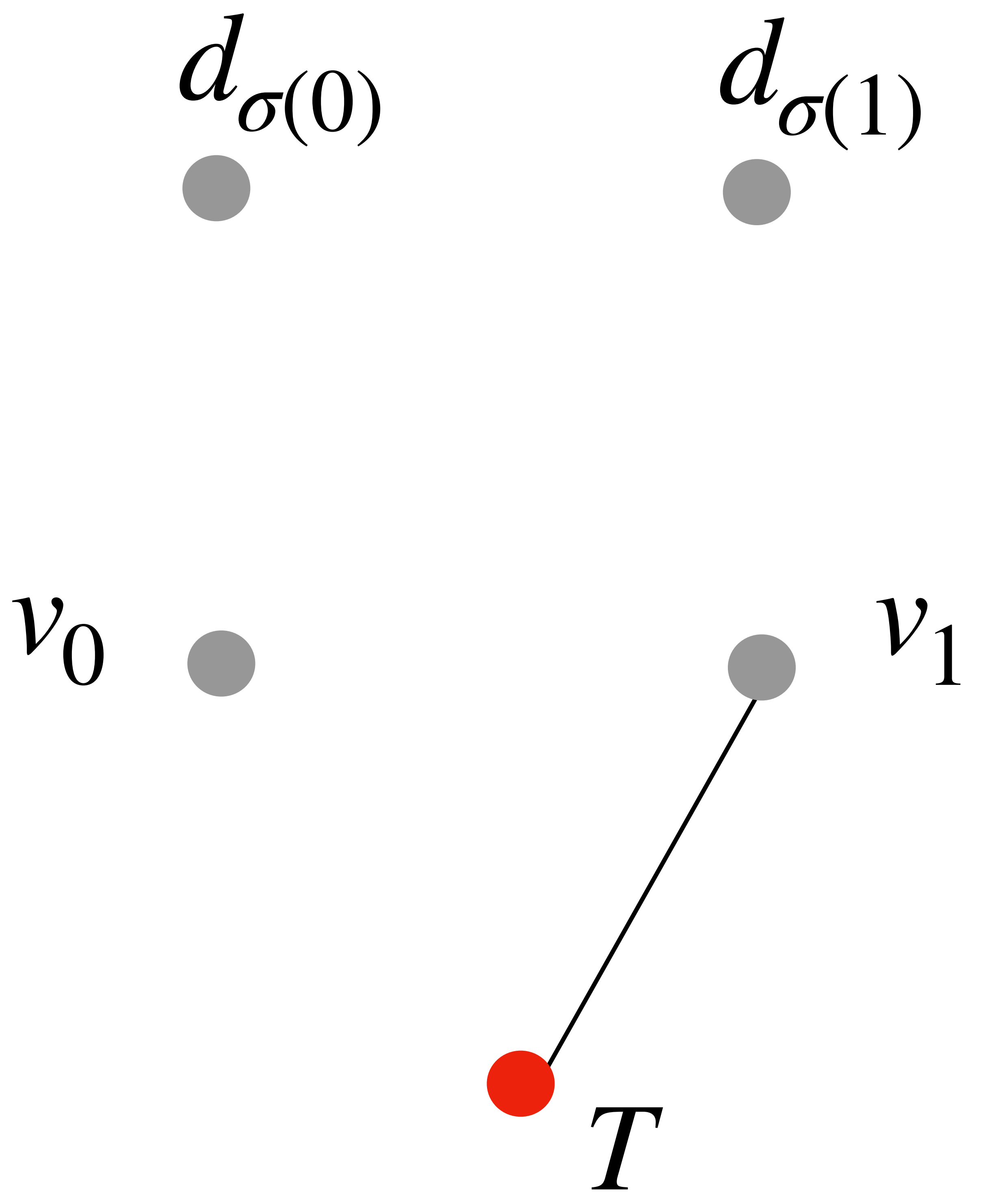


Proofs of knowledge



(A, B, C, b) secret

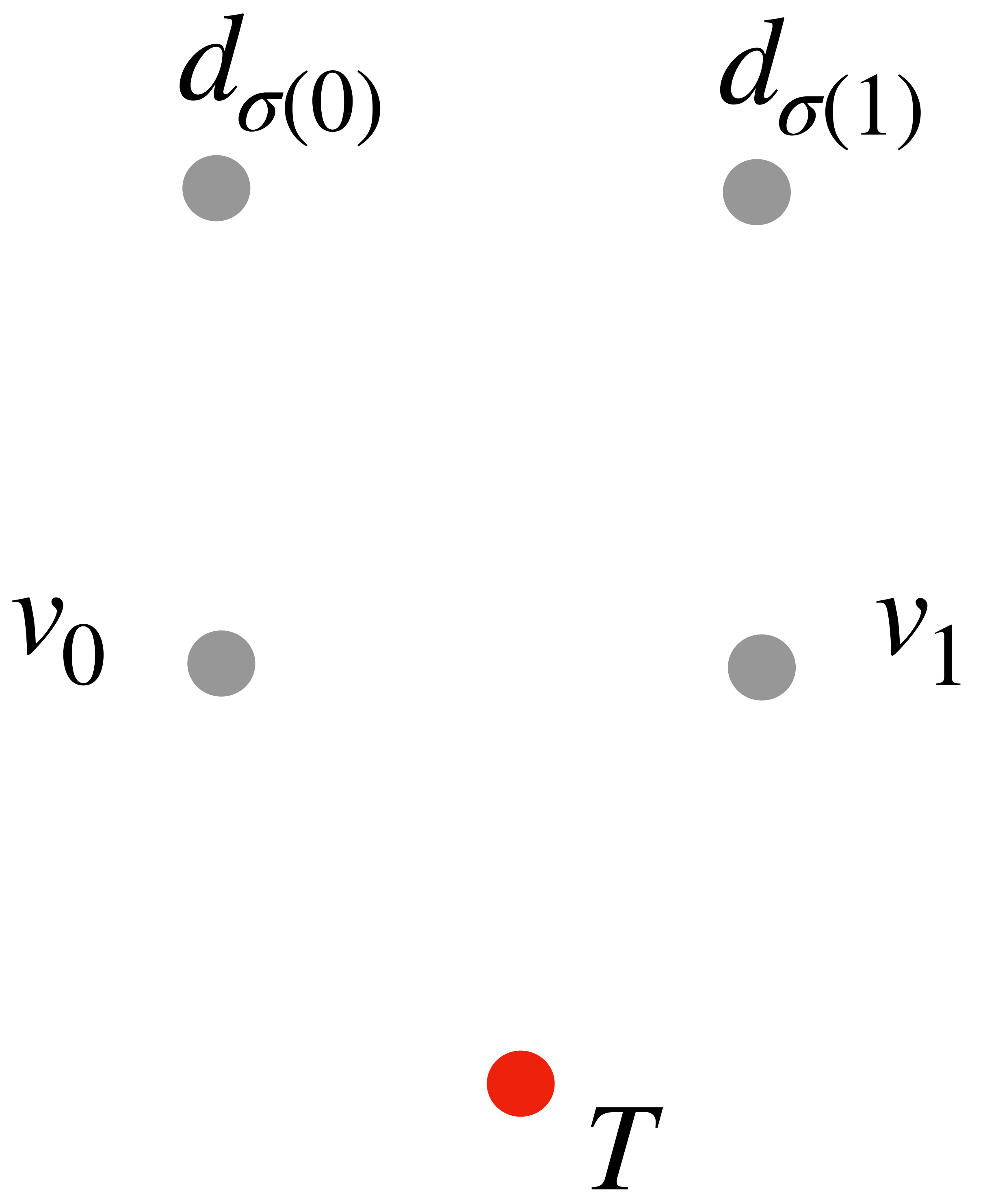
Proofs of knowledge



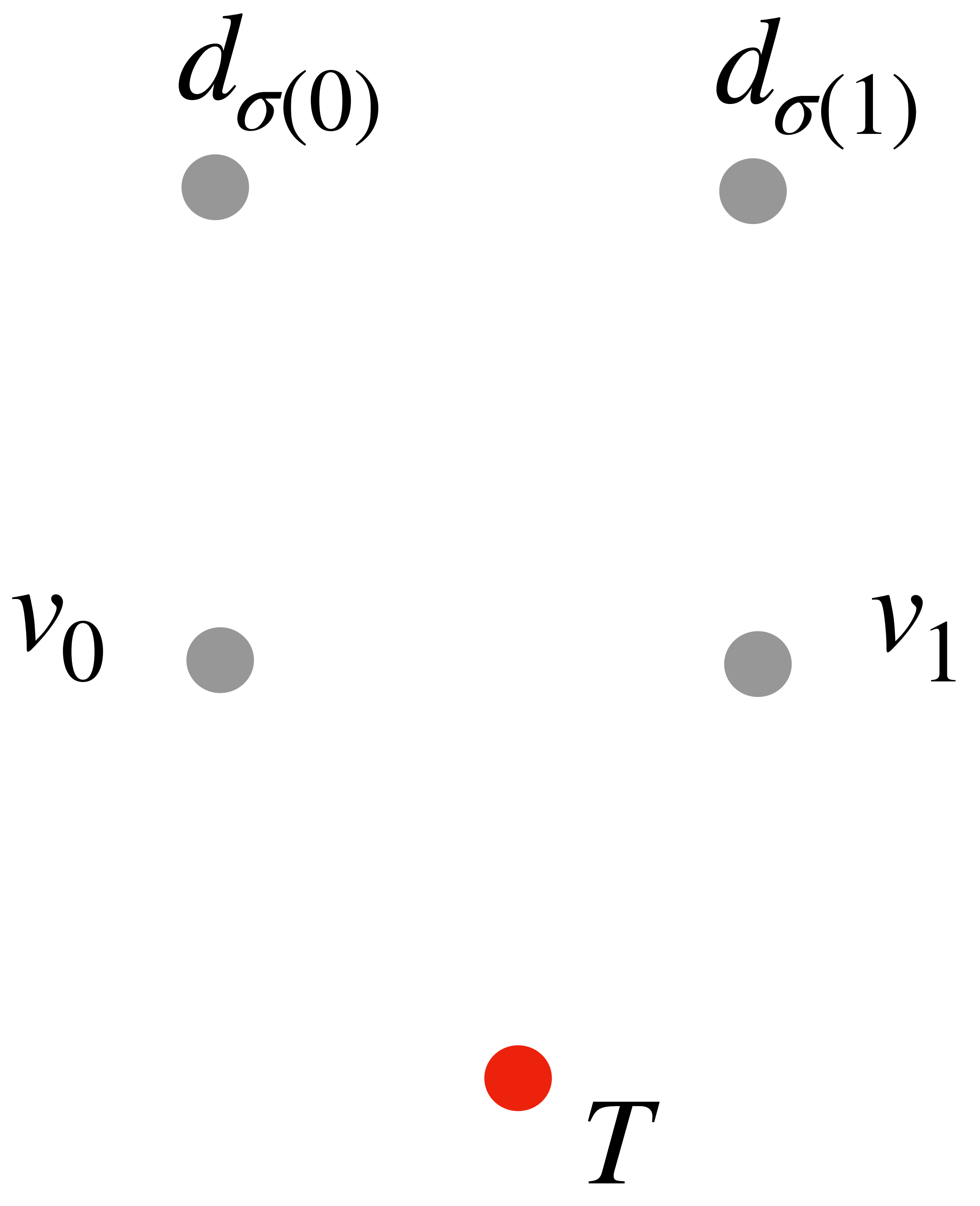
(A, B, C, b) secret

Proofs of knowledge

(A, B, C, b) secret



Proofs of knowledge

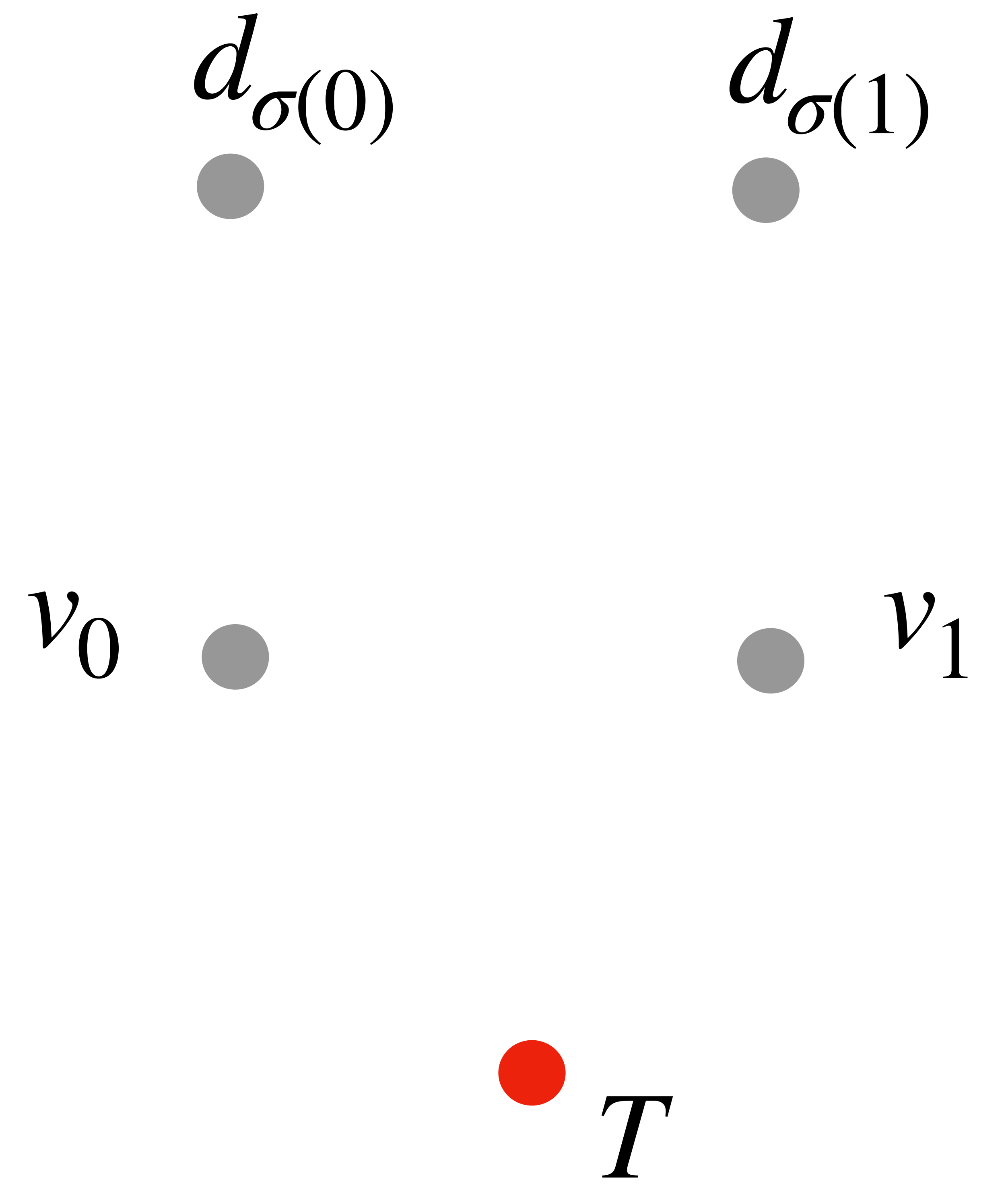


(A, B, C, b) secret

Prover:

Verifier:

Proofs of knowledge



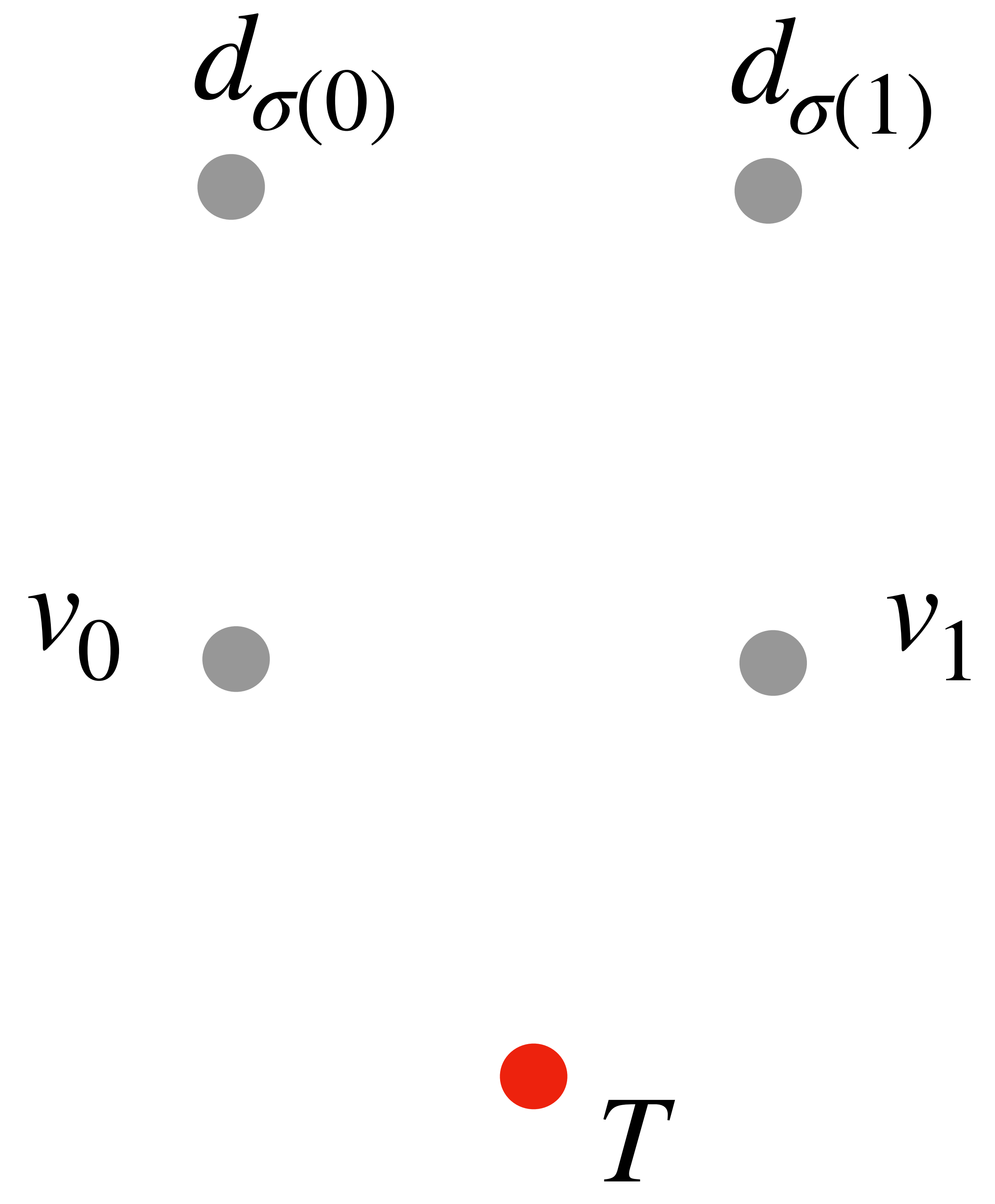
(A, B, C, b) secret

Prover:

$(v_0, v_1), (d_{\sigma(0)}, d_{\sigma(1)}), T$ →

Verifier:

Proofs of knowledge



(A, B, C, b) secret

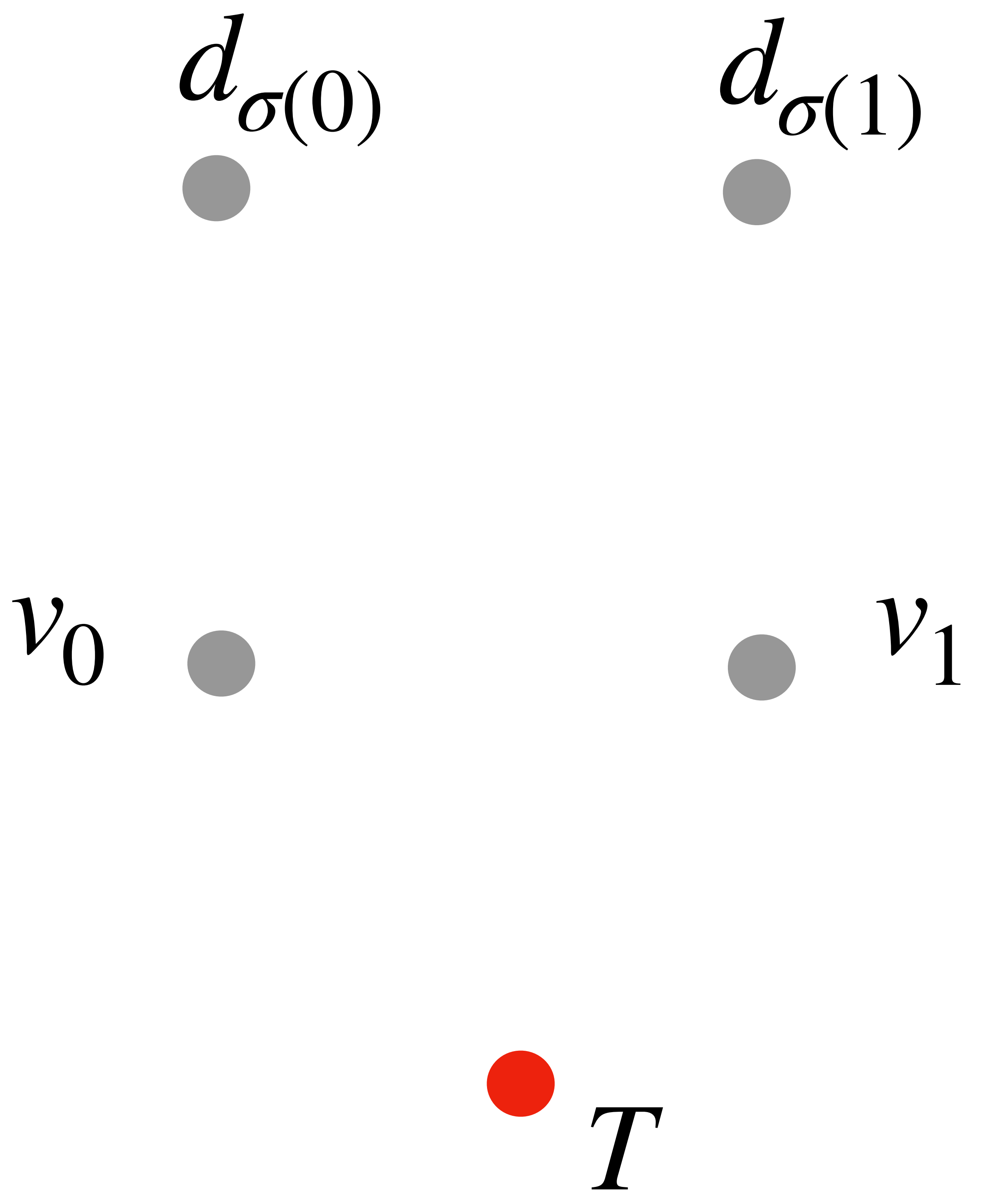
Prover:

$(v_0, v_1), (d_{\sigma(0)}, d_{\sigma(1)}), T$ →

Verifier:

← $ch \in \{0,1\}$

Proofs of knowledge



(A, B, C, b) secret

Prover:

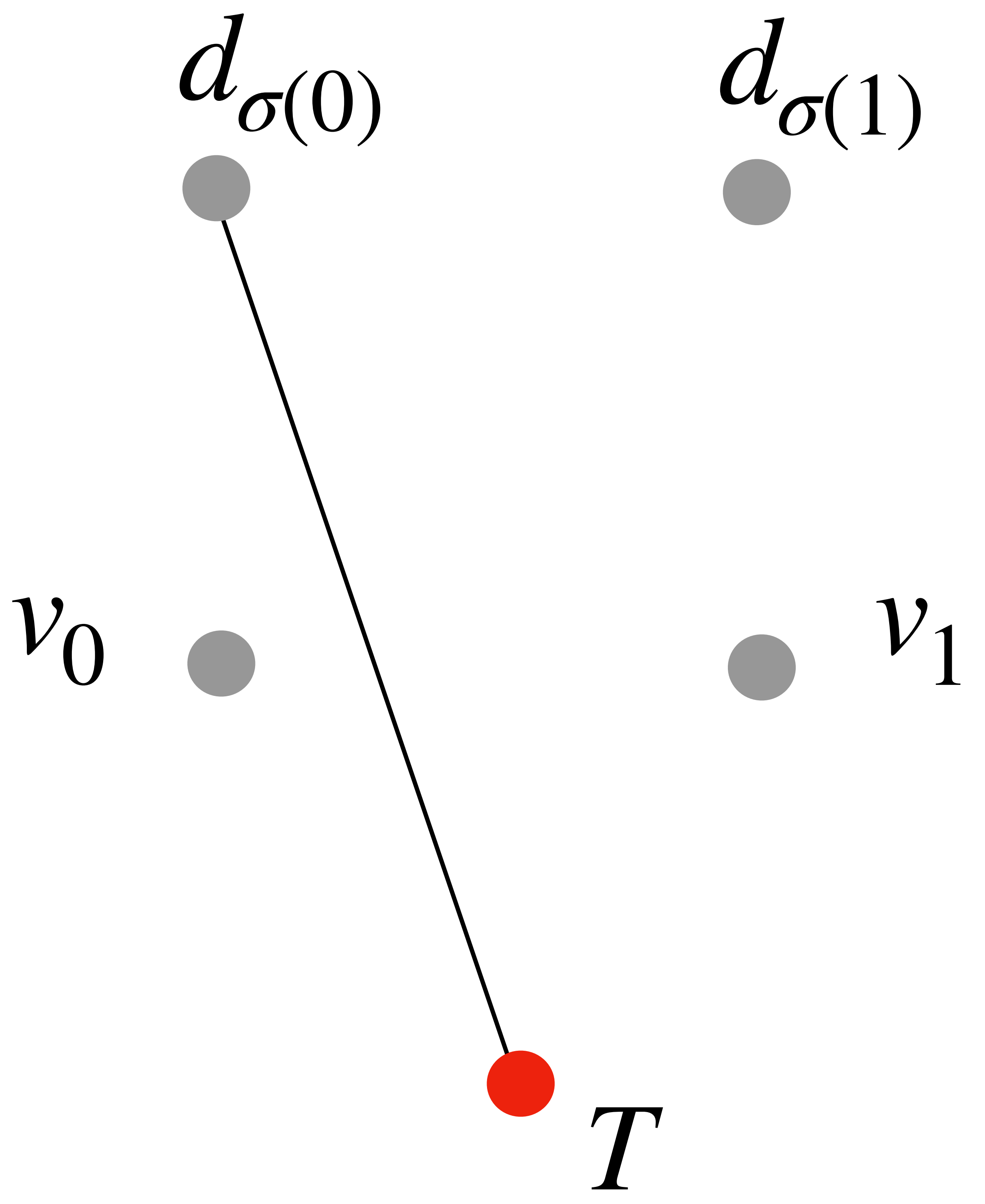
Verifier:

$(v_0, v_1), (d_{\sigma(0)}, d_{\sigma(1)}), T$ →

← $ch \in \{0,1\}$

$$\begin{cases} \sigma, (A', B', C'), & ch = 0 \\ (A'A, B'B, C'C), \sigma(b), & ch = 1 \end{cases}$$ →

Proofs of knowledge



(A, B, C, b) secret

Prover:

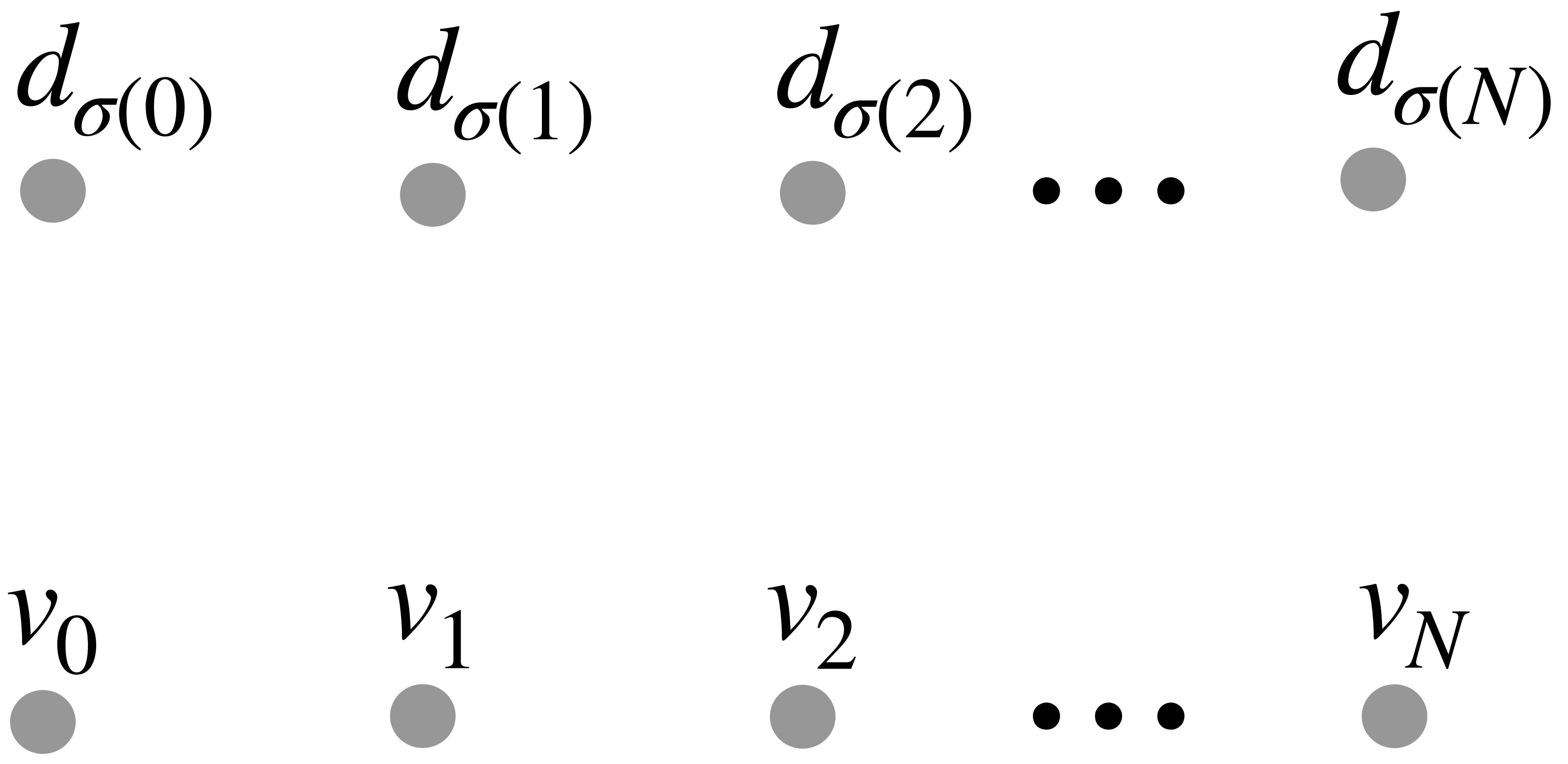
Verifier:

$(v_0, v_1), (d_{\sigma(0)}, d_{\sigma(1)}), T$ →

← $ch \in \{0,1\}$

$$\begin{cases} \sigma, (A', B', C'), & ch = 0 \\ (A'A, B'B, C'C), \sigma(b), & ch = 1 \end{cases}$$ →

Proofs of knowledge

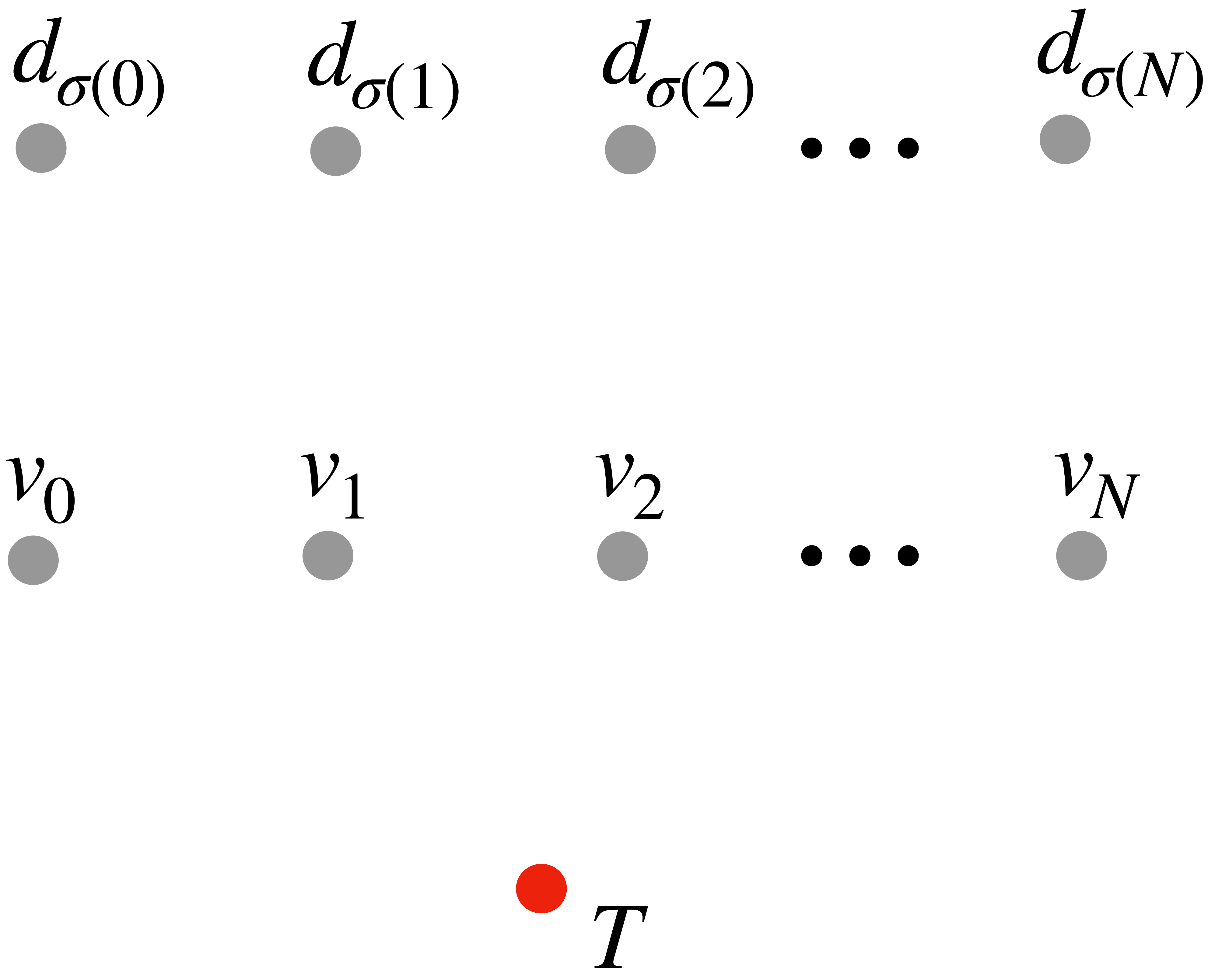


(A, B, C, b) secret

→ if $ch = 0$ then reveal the isomorphisms between v_i and $d_{\sigma(i)}$

→ if $ch = 1$ then reveal the isomorphism from $d_{\sigma(b)}$ to T

Proofs of knowledge

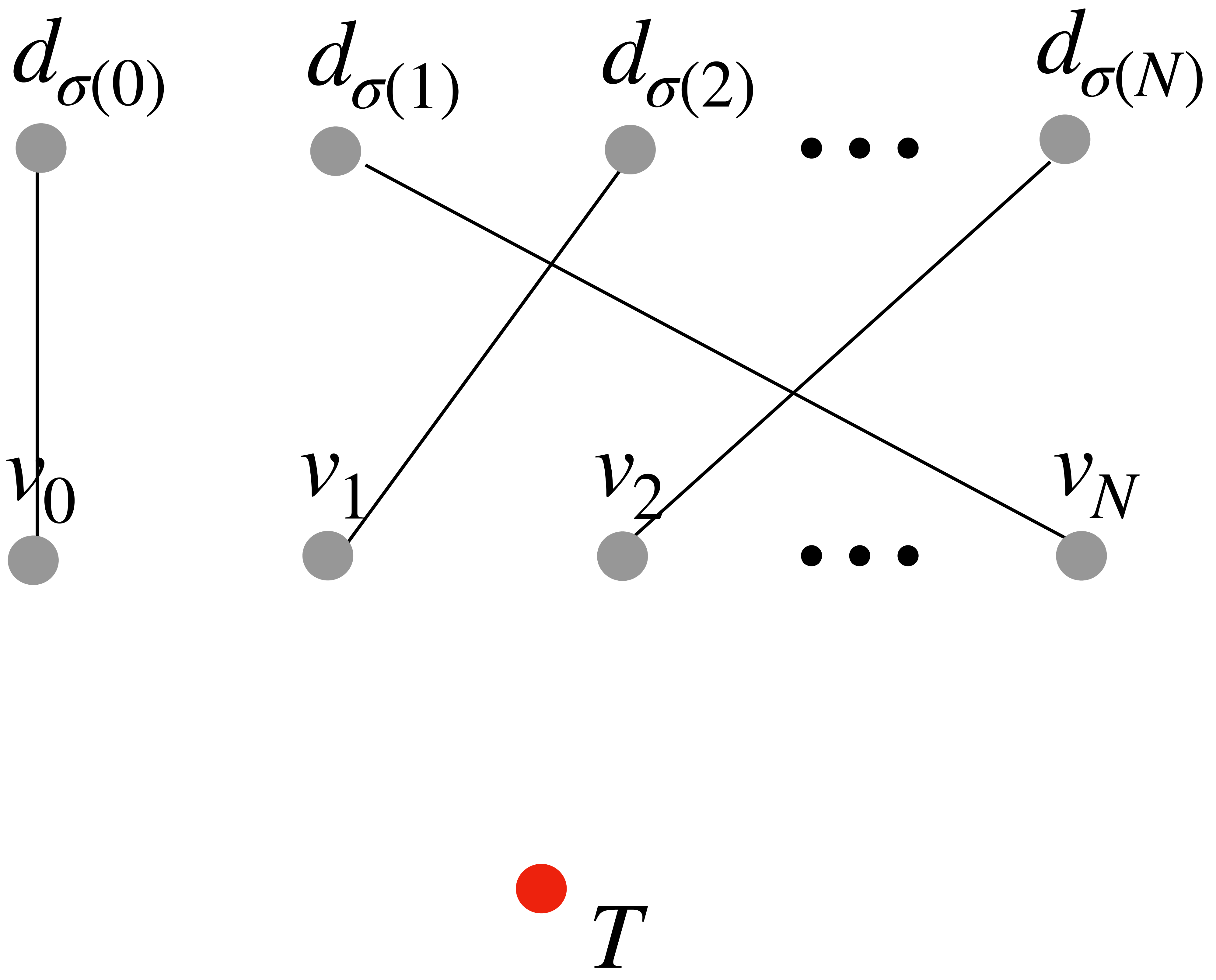


(A, B, C, b) secret

→ if $ch = 0$ then reveal the isomorphisms between v_i and $d_{\sigma(i)}$

→ if $ch = 1$ then reveal the isomorphism from $d_{\sigma(b)}$ to T

Proofs of knowledge

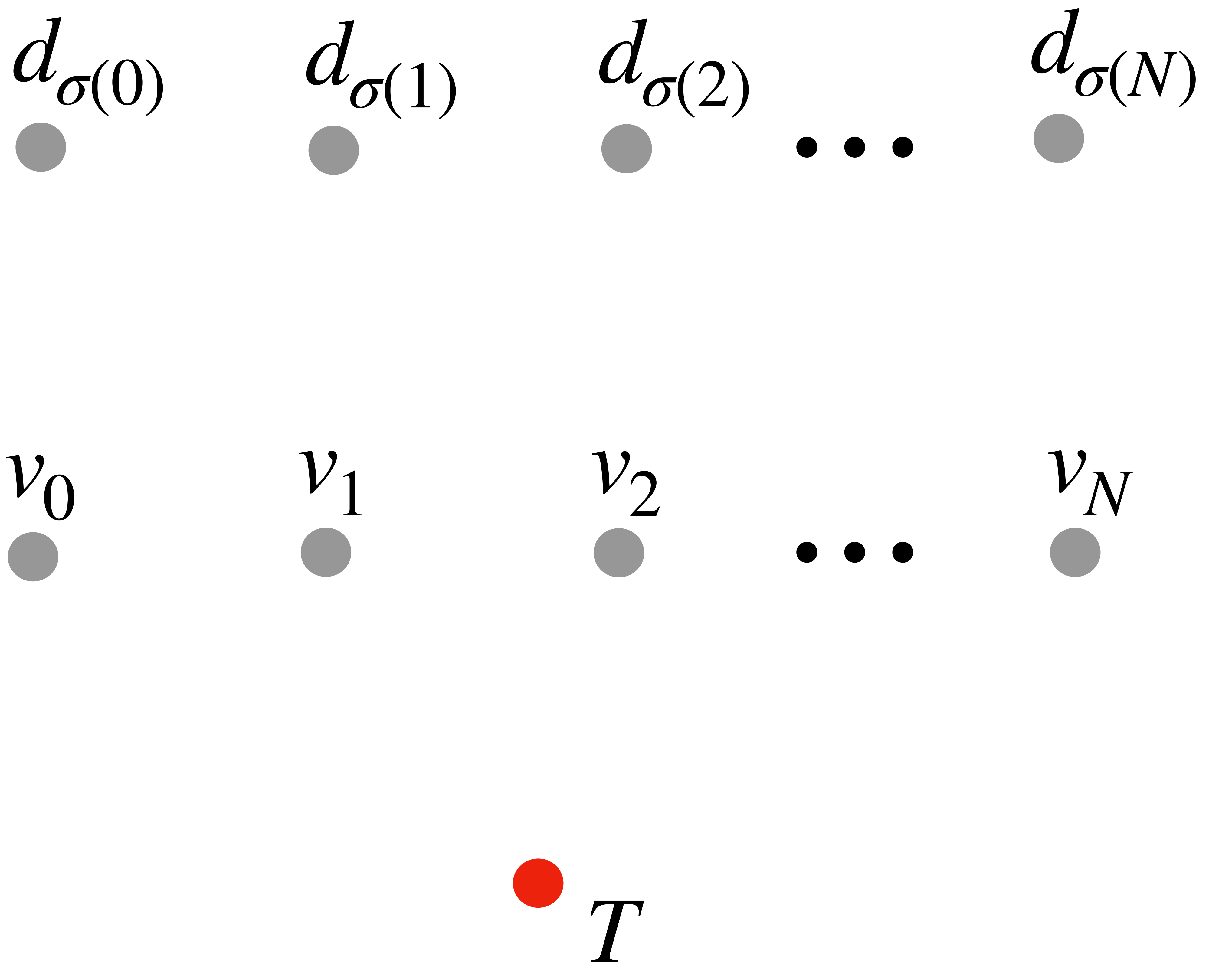


(A, B, C, b) secret

→ if $ch = 0$ then reveal the isomorphisms between v_i and $d_{\sigma(i)}$

→ if $ch = 1$ then reveal the isomorphism from $d_{\sigma(b)}$ to T

Proofs of knowledge

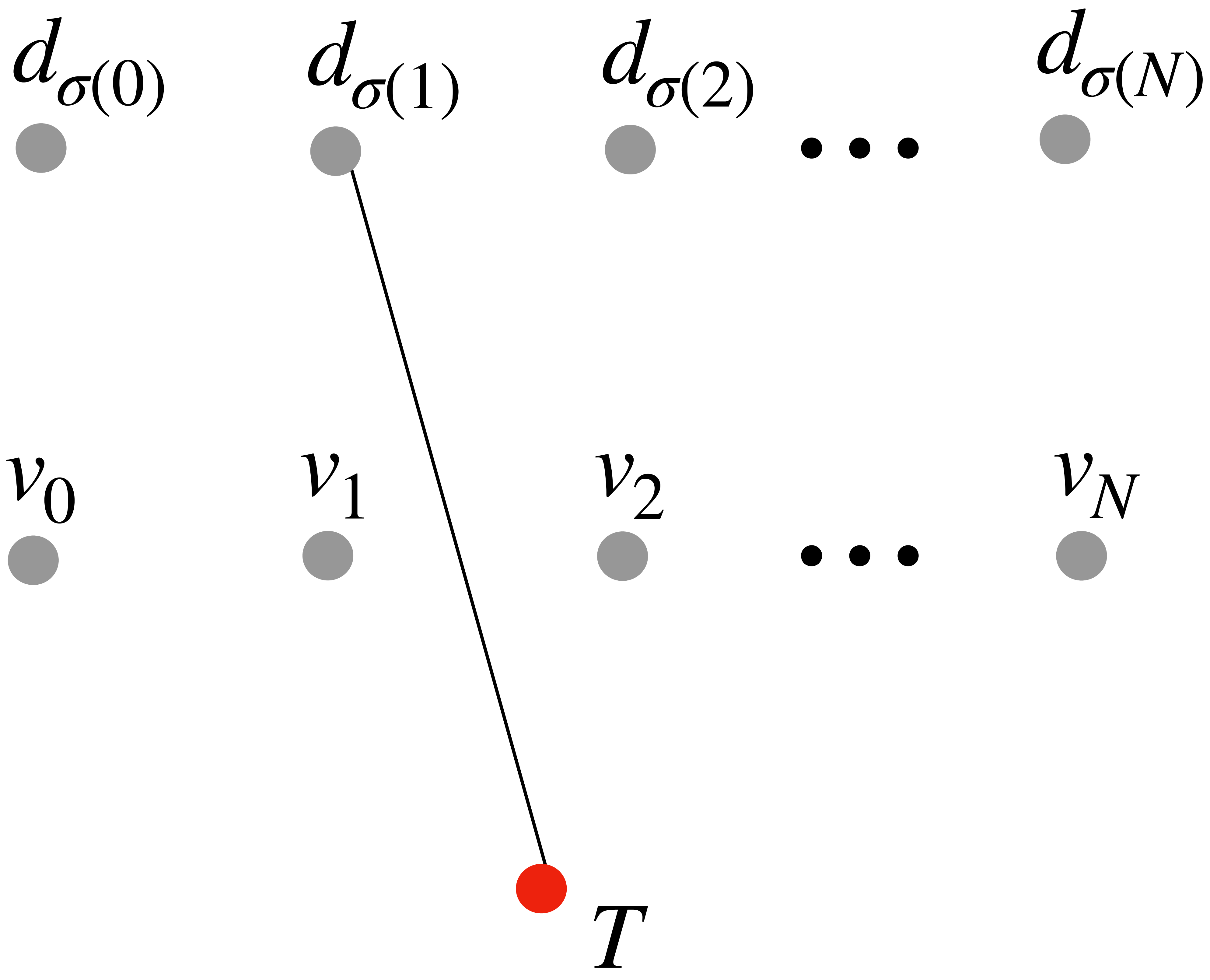


(A, B, C, b) secret

→ if $ch = 0$ then reveal the isomorphisms between v_i and $d_{\sigma(i)}$

→ if $ch = 1$ then reveal the isomorphism from $d_{\sigma(b)}$ to T

Proofs of knowledge



(A, B, C, b) secret

→ if $ch = 0$ then reveal the isomorphisms between v_i and $d_{\sigma(i)}$

→ if $ch = 1$ then reveal the isomorphism from $d_{\sigma(b)}$ to T

Summary:

Solving the Tensor Isomorphism Problem

for special orbits with low rank points:

Cryptanalysis and repair

of an Asiacrypt 2023 commitment scheme

Summary:

Solving the Tensor Isomorphism Problem

for special orbits with low rank points:

Cryptanalysis and repair

of an Asiacrypt 2023 commitment scheme

Summary:

Solving the Tensor Isomorphism Problem

for special orbits with low rank points:

Cryptanalysis and repair

of an Asiacrypt 2023 commitment scheme

Summary:

Solving the Tensor Isomorphism Problem

for special orbits with low rank points:

Cryptanalysis and repair

of an Asiacrypt 2023 commitment scheme

Summary:

Solving the Tensor Isomorphism Problem

for special orbits with low rank points:

Cryptanalysis and repair

of an Asiacrypt 2023 commitment scheme

Summary:

Solving the Tensor Isomorphism Problem

for special orbits with low rank points:

Cryptanalysis and repair

of an Asiacrypt 2023 commitment scheme

Summary:

Solving the Tensor Isomorphism Problem

for special orbits with low rank points:

Cryptanalysis and repair

of an Asiacrypt 2023 commitment scheme

Summary:

Solving the Tensor Isomorphism Problem

for special orbits with low rank points:

Cryptanalysis and repair

of an Asiacrypt 2023 commitment scheme

Summary:

Solving the Tensor Isomorphism Problem

for special orbits with low rank points:

Cryptanalysis and repair

of an Asiacrypt 2023 commitment scheme

→ ePrint:2024/337

Summary:

Solving the Tensor Isomorphism Problem

for special orbits with low rank points:

Cryptanalysis and repair

of an Asiacrypt 2023 commitment scheme

→ ePrint:2024/337

Thank you!