Bruisable Onions: Anonymous Communication in the Asynchronous Model

Megumi Ando, Anna Lysyanskaya, and Eli Upfal

TCC 2024

Tufts UNIVERSITY | SCHOOL OF ENGINEERING Computer Science

BROWN

mando@cs.tufts.edu

{anna, eli}@cs.brown.edu

The Technical Problem: Anonymous Communication

Alice $\xrightarrow{c = 00101\ldots}$ **Bob**

$\mathsf{Enc}_{\mathsf{pk(Bob)}}(m) \to c$

$\mathsf{Dec}_{\mathsf{sk(Bob)}}(c) \to m$

A Practical Solution: Onion Routing [Chaum 81]

Notation:

$[\text{plaintext}]_{\text{key}}$ = encryption under key

$$\textbf{Alice} \xrightarrow{O_1} I_1 \longrightarrow I_2 \longrightarrow I_3 \longrightarrow I_4 \longrightarrow \textbf{Bob}$$

$$O_1 = [[[[[[m]_{\text{Bob}}, \text{Bob}]_{I_4}, I_4]_{I_3}, I_3]_{I_2}, I_2]_{I_1}$$

A Practical Solution: Onion Routing [Chaum 81]

Notation:

$[\text{plaintext}]_{\text{key}}$ = encryption under key

$$\textbf{Alice} \xrightarrow{O_1} I_1 \xrightarrow{O_2} I_2 \longrightarrow I_3 \longrightarrow I_4 \longrightarrow \textbf{Bob}$$

$$O_1 = [[[[[[m]_{\text{Bob}}, \text{Bob}]_{I_4}, I_4]_{I_3}, I_3]_{I_2}, I_2]_{I_1}$$

$$O_2 = [[[[m]_{\text{Bob}}, \text{Bob}]_{I_4}, I_4]_{I_3}, I_3]_{I_2}$$

A Practical Solution: Onion Routing [Chaum 81]

Notation:

$[\text{plaintext}]_{\text{key}}$ = encryption under key

$$\textbf{Alice} \xrightarrow{O_1} I_1 \xrightarrow{O_2} I_2 \xrightarrow{O_3} I_3 \longrightarrow I_4 \longrightarrow \textbf{Bob}$$

$$O_1 = [[[[[[m]_{\text{Bob}}, \text{Bob}]_{I_4}, I_4]_{I_3}, I_3]_{I_2}, I_2]_{I_1}$$

$$O_2 = [[[[m]_{\text{Bob}}, \text{Bob}]_{I_4}, I_4]_{I_3}, I_3]_{I_2}$$

$$O_3 = [[[m]_{\text{Bob}}, \text{Bob}]_{I_4}, I_4]_{I_3}$$

A Practical Solution: Onion Routing [Chaum 81]

Notation:

$[\text{plaintext}]_{\text{key}}$ = encryption under key

$$\text{Alice} \xrightarrow{O_1} I_1 \xrightarrow{O_2} I_2 \xrightarrow{O_3} I_3 \xrightarrow{O_4} I_4 \xrightarrow{\quad} \text{Bob}$$

$$O_1 = [[[[[m]_{\text{Bob}}, \text{Bob}]_{I_4}, I_4]_{I_3}, I_3]_{I_2}, I_2]_{I_1}$$

$$O_2 = [[[[m]_{\text{Bob}}, \text{Bob}]_{I_4}, I_4]_{I_3}, I_3]_{I_2}$$

$$O_3 = [[[m]_{\text{Bob}}, \text{Bob}]_{I_4}, I_4]_{I_3}$$

$$O_4 = [[m]_{\text{Bob}}, \text{Bob}]_{I_4}$$

A Practical Solution: Onion Routing [Chaum 81]

Notation:

$[\text{plaintext}]_{\text{key}}$ = encryption under key

$$\textbf{Alice} \xrightarrow{O_1} I_1 \xrightarrow{O_2} I_2 \xrightarrow{O_3} I_3 \xrightarrow{O_4} I_4 \xrightarrow{O_5} \textbf{Bob}$$

$$O_1 = [[[[[m]_{\text{Bob}}, \text{Bob}]_{I_4}, I_4]_{I_3}, I_3]_{I_2}, I_2]_{I_1}$$

$$O_2 = [[[[m]_{\text{Bob}}, \text{Bob}]_{I_4}, I_4]_{I_3}, I_3]_{I_2}$$

$$O_3 = [[[m]_{\text{Bob}}, \text{Bob}]_{I_4}, I_4]_{I_3}$$

$$O_4 = [[m]_{\text{Bob}}, \text{Bob}]_{I_4}$$

$$O_5 = [m]_{\text{Bob}}$$

A Practical Solution: Onion Routing [Chaum 81]

Notation:

$[\text{plaintext}]_{\text{key}}$ = encryption under key

$$\textbf{Alice} \xrightarrow{O_1} I_1 \xrightarrow{O_2} I_2 \xrightarrow{O_3} I_3 \xrightarrow{O_4} I_4 \xrightarrow{O_5} \textbf{Bob} \qquad m$$

$$O_1 = [[[[[m]_{\text{Bob}}, \text{Bob}]_{I_4}, I_4]_{I_3}, I_3]_{I_2}, I_2]_{I_1}$$

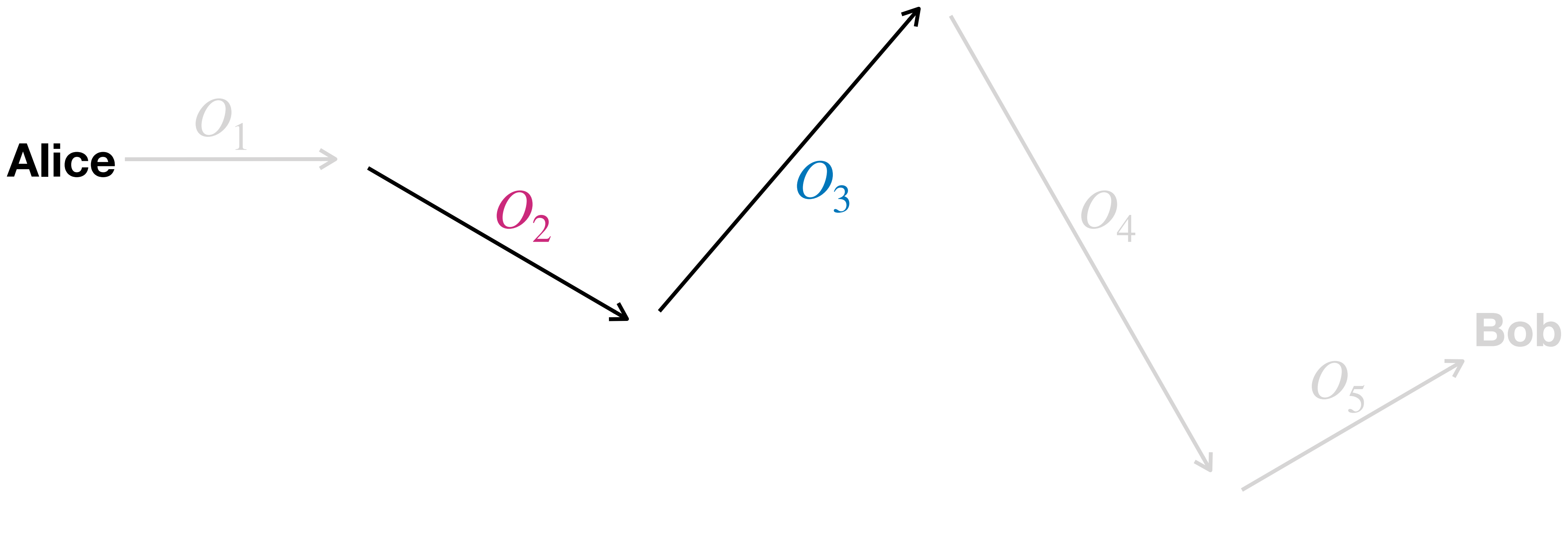
$$O_2 = [[[[m]_{\text{Bob}}, \text{Bob}]_{I_4}, I_4]_{I_3}, I_3]_{I_2}$$

$$O_3 = [[[m]_{\text{Bob}}, \text{Bob}]_{I_4}, I_4]_{I_3}$$

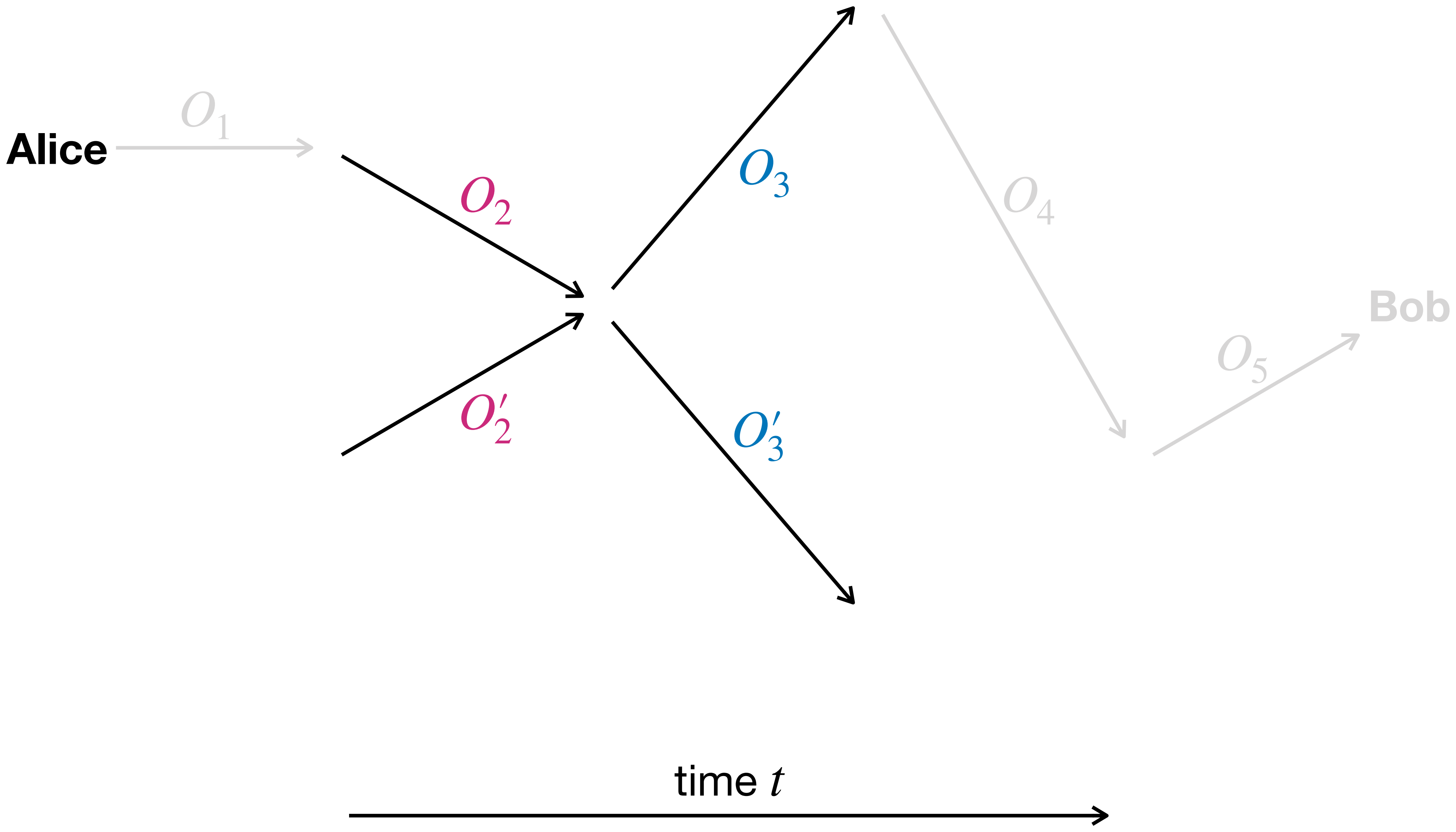
$$O_4 = [[m]_{\text{Bob}}, \text{Bob}]_{I_4}$$

$$O_5 = [m]_{\text{Bob}}$$

A Practical Solution: Onion Routing in Mixnets

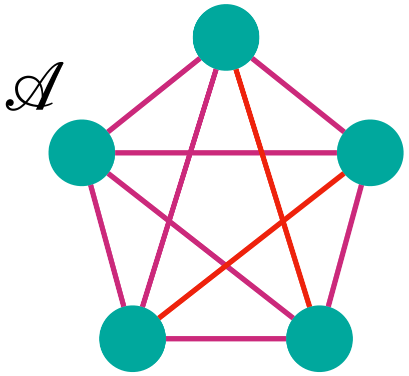


A Practical Solution: Onion Routing in Mixnets

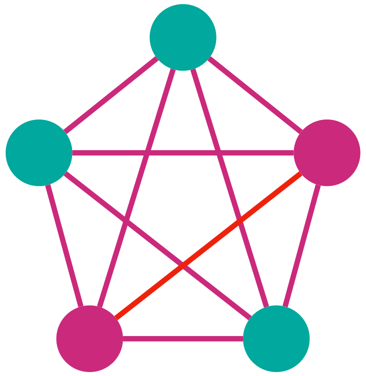


Standard Adversary Models

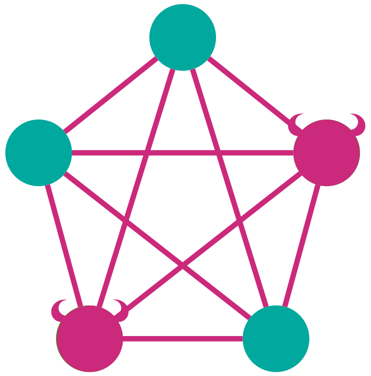
Adversary $\mathscr{A}$



Network:
all links



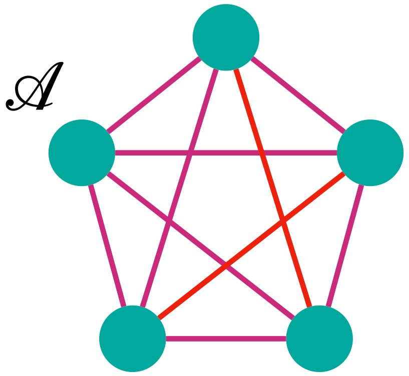
Passive:
all links, $\Theta(1)$ nodes



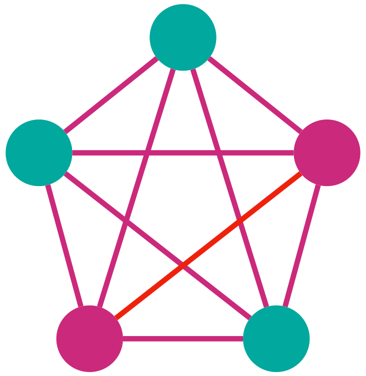
Active:
all links, controls $\Theta(1)$ nodes

Defining Anonymity

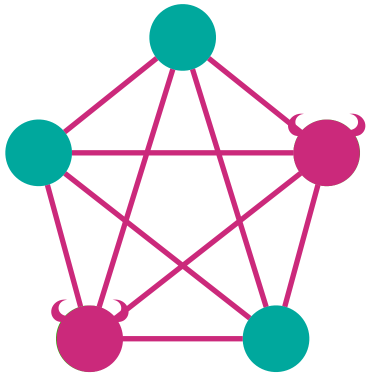
Adversary $\mathscr{A}$



Network:
all links

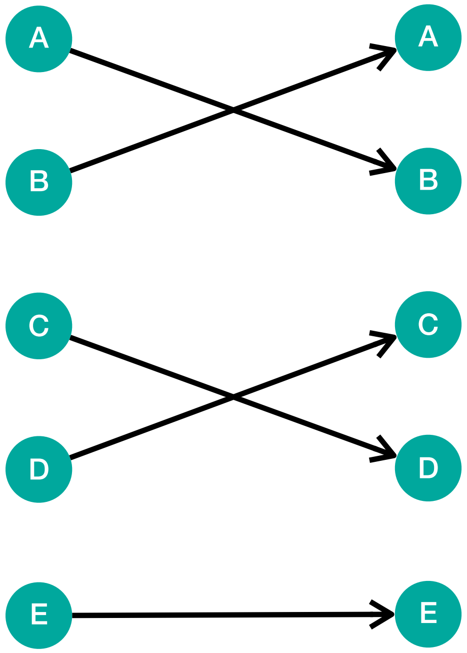


Passive:
all links, $\Theta(1)$ nodes

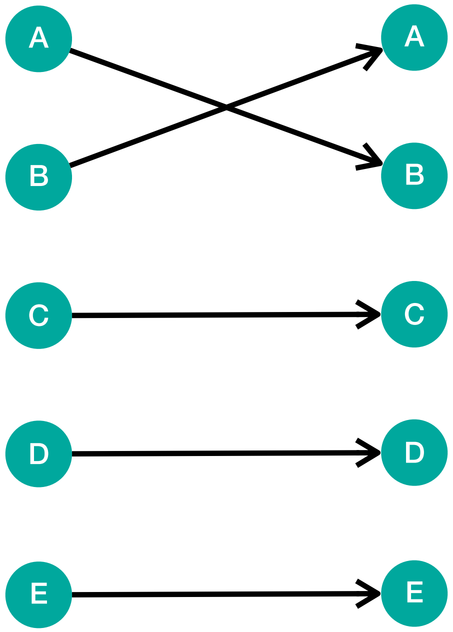


Active:
all links, controls $\Theta(1)$ nodes

“sends to”



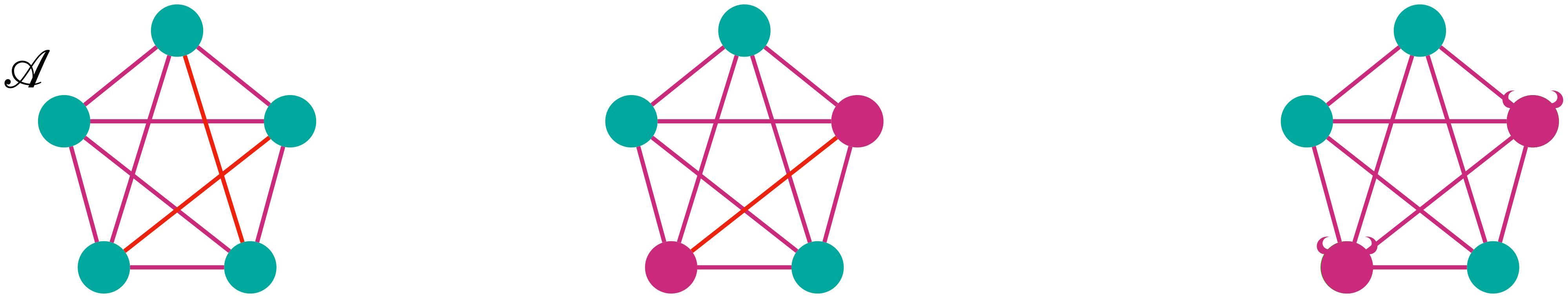
input 0

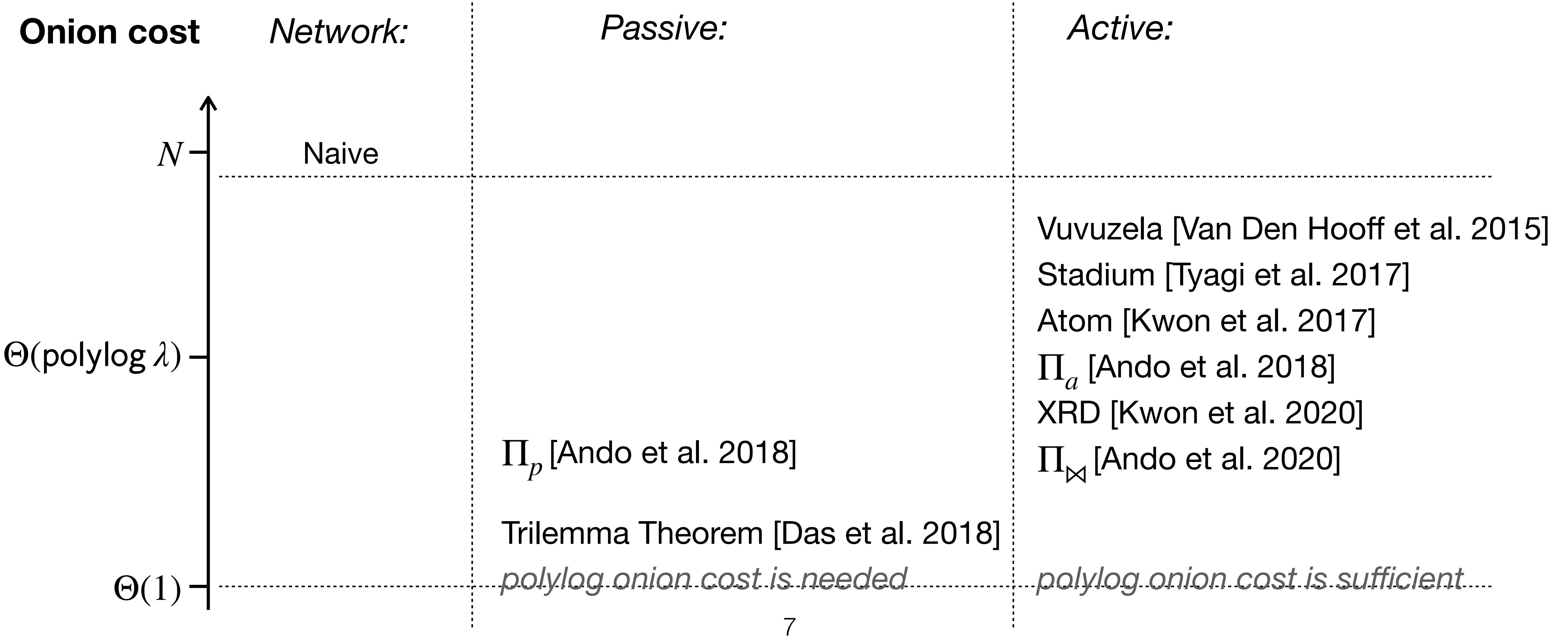


input 1

Existing Related Work: All in Synchronous Setting

Adversary $\mathscr{A}$





Vuvuzela [Van Den Hooff et al. 2015]

Stadium [Tyagi et al. 2017]

Atom [Kwon et al. 2017]

Π_a [Ando et al. 2018]

XRD [Kwon et al. 2020]

$\Pi_{\bowtie}$ [Ando et al. 2020]

Π_p [Ando et al. 2018]

Trilemma Theorem [Das et al. 2018]

polylog onion cost is needed

polylog onion cost is sufficient

Challenges in Achieving Anonymity in the Asynchronous Setting

Challenges in Achieving Anonymity in the Asynchronous Setting

1. ***No global clock:*** when to batch-process onions

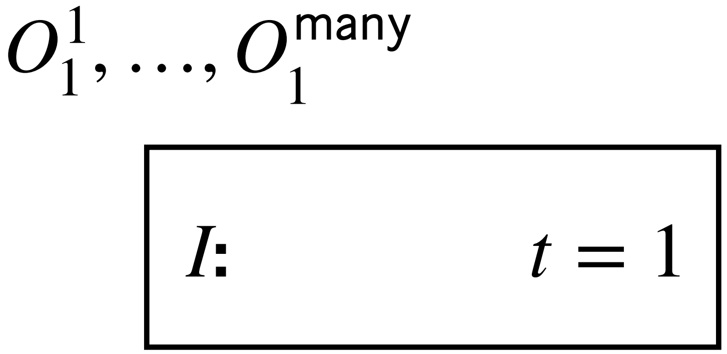
Challenges in Achieving Anonymity in the Asynchronous Setting

1. ***No global clock:*** when to batch-process onions

I**:** $t = 1$

Challenges in Achieving Anonymity in the Asynchronous Setting

1. ***No global clock:*** when to batch-process onions

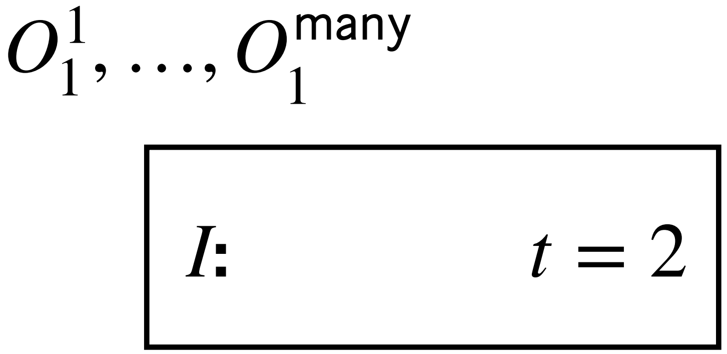


$O_1^1, \ldots, O_1^{\text{many}}$

I: $t = 1$

Challenges in Achieving Anonymity in the Asynchronous Setting

1. ***No global clock:*** when to batch-process onions

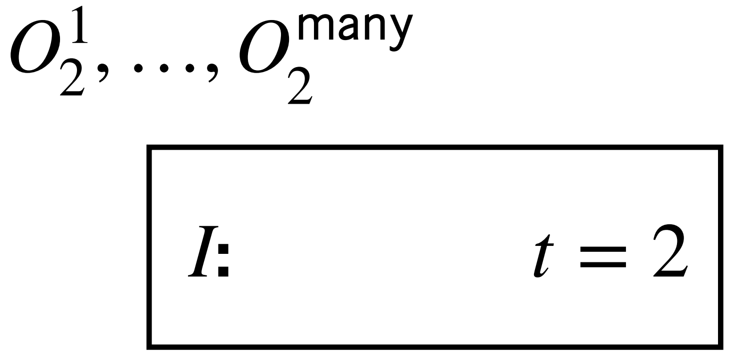


$O_1^1, \ldots, O_1^{\text{many}}$

I: $t = 2$

Challenges in Achieving Anonymity in the Asynchronous Setting

1. ***No global clock:*** when to batch-process onions

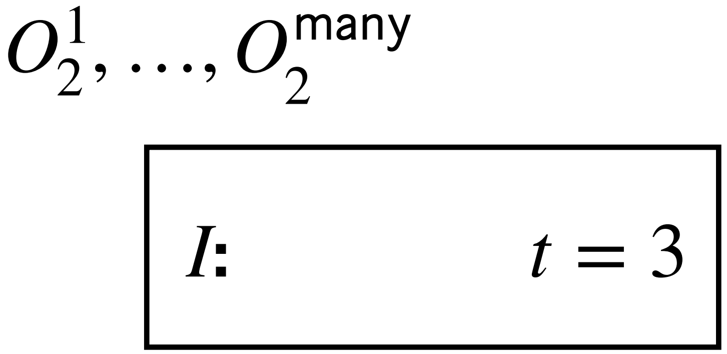


$O_2^1, \ldots, O_2^{\text{many}}$

I: $t = 2$

Challenges in Achieving Anonymity in the Asynchronous Setting

1. ***No global clock:*** when to batch-process onions



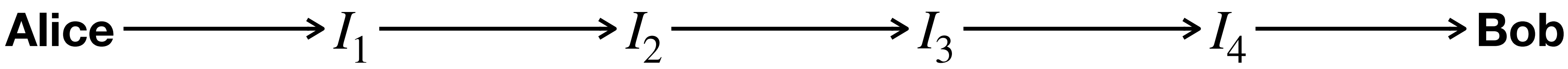
$O_2^1, \ldots, O_2^{\text{many}}$

I: $t = 3$

Challenges in Achieving Anonymity in the Asynchronous Setting

1. ***No global clock:*** when to batch-process onions

2. ***Timing attacks:*** chronically late onions don't shuffle



Alice → I_1 → I_2 → I_3 → I_4 → **Bob**

Challenges in Achieving Anonymity in the Asynchronous Setting

1. ***No global clock:*** when to batch-process onions

2. ***Timing attacks:*** chronically late onions don't shuffle

$$\textbf{Alice} \xrightarrow{O_1} I_1 \longrightarrow I_2 \longrightarrow I_3 \longrightarrow I_4 \longrightarrow \textbf{Bob}$$

$t = 2$

Challenges in Achieving Anonymity in the Asynchronous Setting

1. ***No global clock:*** when to batch-process onions

2. ***Timing attacks:*** chronically late onions don't shuffle

$$\textbf{Alice} \xrightarrow{O_1} I_1 \xrightarrow{O_2} I_2 \longrightarrow I_3 \longrightarrow I_4 \longrightarrow \textbf{Bob}$$

$$t = 3$$

Challenges in Achieving Anonymity in the Asynchronous Setting

1. ***No global clock:*** when to batch-process onions

2. ***Timing attacks:*** chronically late onions don't shuffle

$$\textbf{Alice} \xrightarrow{O_1} I_1 \xrightarrow{O_2} I_2 \xrightarrow{O_3} I_3 \xrightarrow{\quad} I_4 \xrightarrow{\quad} \textbf{Bob}$$

$$t = 4$$

Challenges in Achieving Anonymity in the Asynchronous Setting

1. ***No global clock:*** when to batch-process onions

2. ***Timing attacks:*** chronically late onions don't shuffle

$$\begin{tikzcd}
\textbf{Alice} \arrow[r, "O_1"] & I_1 \arrow[r, "O_2"] & I_2 \arrow[r, "O_3"] & I_3 \arrow[r, "O_4"] & I_4 \arrow[r] & \textbf{Bob}
\end{tikzcd}$$

$t = 5$

Challenges in Achieving Anonymity in the Asynchronous Setting

1. ***No global clock:*** when to batch-process onions

2. ***Timing attacks:*** chronically late onions don’t shuffle

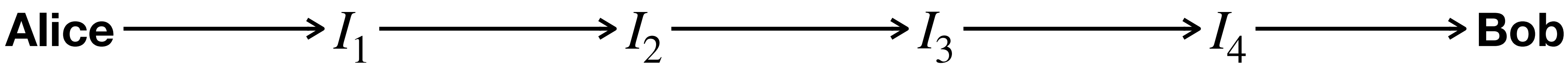
$$\textbf{Alice} \xrightarrow{O_1} I_1 \xrightarrow{O_2} I_2 \xrightarrow{O_3} I_3 \xrightarrow{O_4} I_4 \xrightarrow{O_5} \textbf{Bob}$$

chronically late;
didn’t shuffle

Main Idea: Bruisable Onions

1. ***No global clock:*** when to batch-process onions

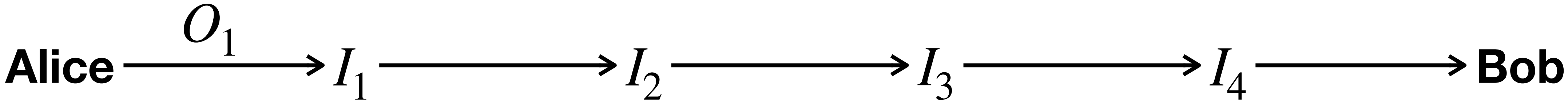
2. ***Timing attacks:*** chronically late onions don't shuffle



Main Idea: Bruisable Onions

1. ***No global clock:*** when to batch-process onions

2. ***Timing attacks:*** chronically late onions don't shuffle

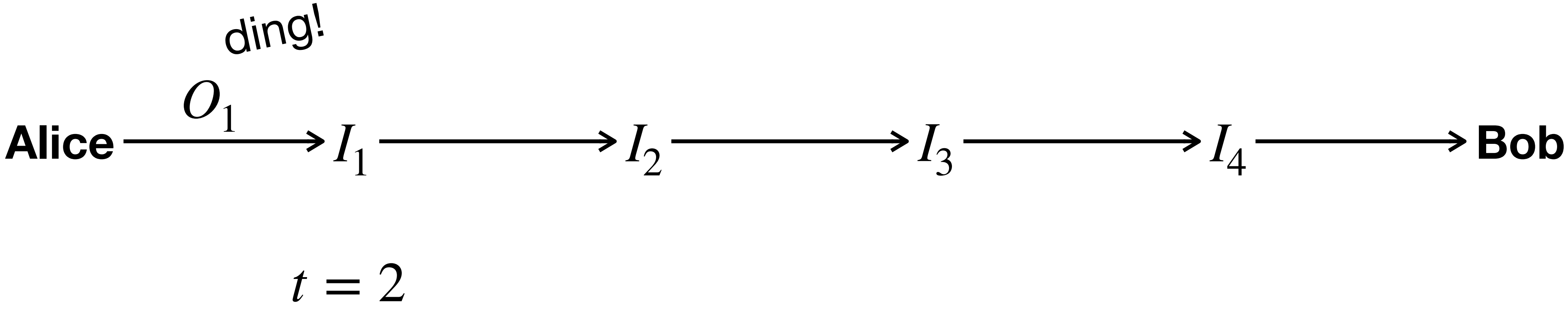


$t = 2$

Main Idea: Bruisable Onions

1. ***No global clock:*** when to batch-process onions

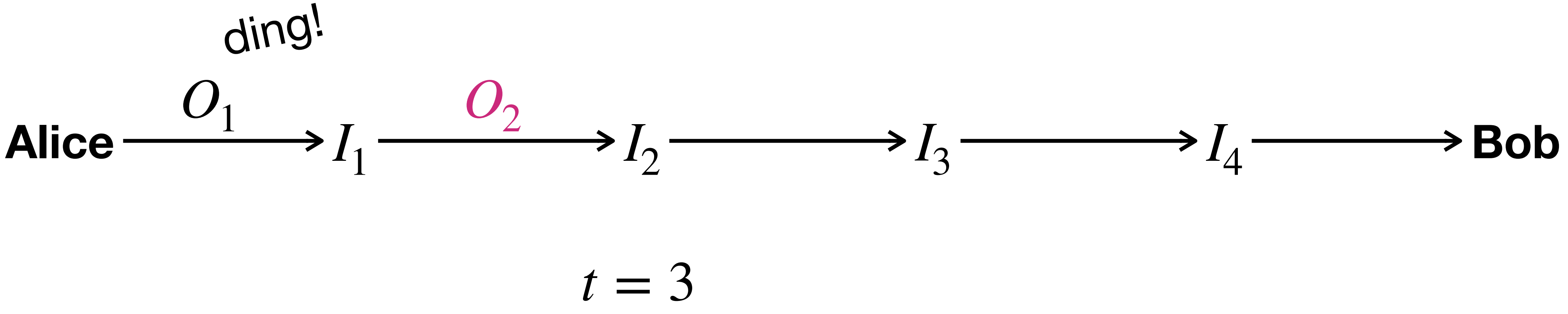
2. ***Timing attacks:*** chronically late onions don't shuffle



Main Idea: Bruisable Onions

1. ***No global clock:*** when to batch-process onions

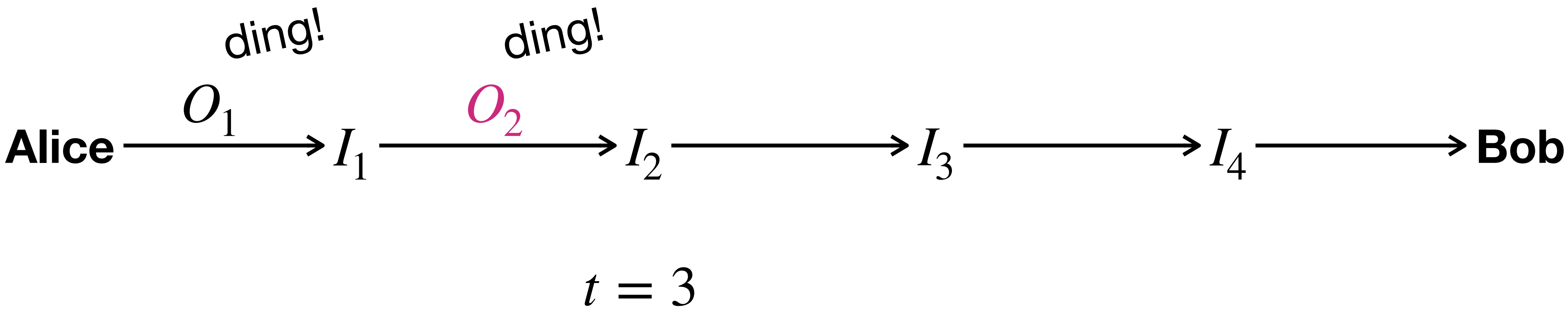
2. ***Timing attacks:*** chronically late onions don’t shuffle



Main Idea: Bruisable Onions

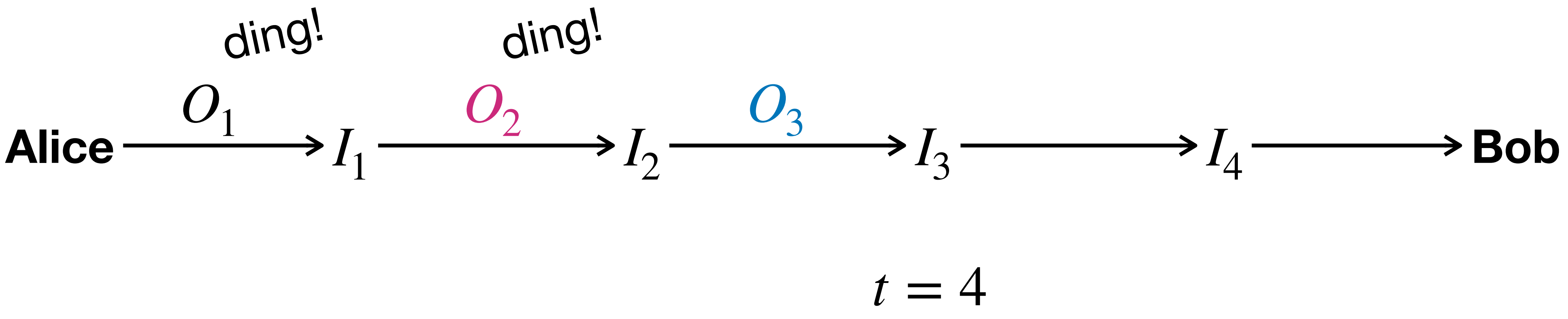
1. ***No global clock:*** when to batch-process onions

2. ***Timing attacks:*** chronically late onions don't shuffle



Main Idea: Bruisable Onions

1. ***No global clock:*** when to batch-process onions
2. ***Timing attacks:*** chronically late onions don't shuffle



ding!

ding!

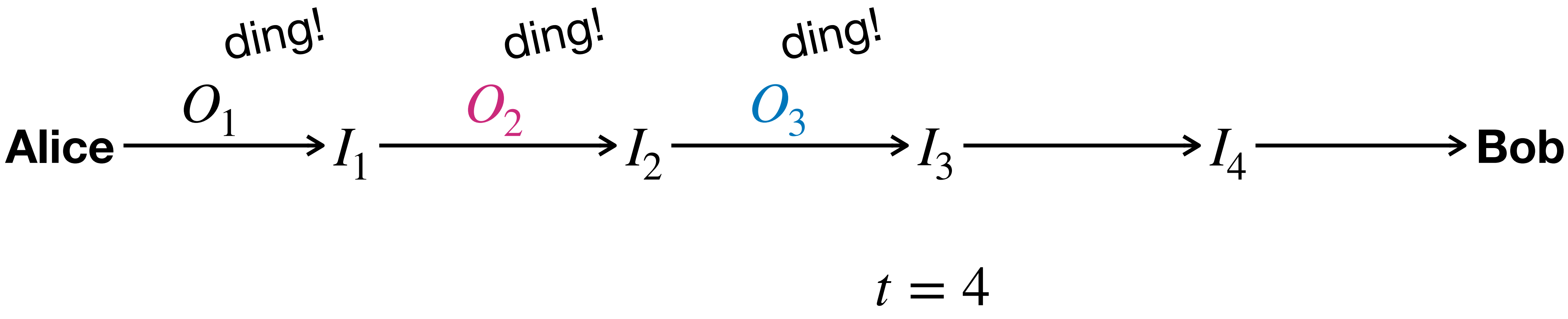
Alice $\xrightarrow{O_1}$ I_1 $\xrightarrow{O_2}$ I_2 $\xrightarrow{O_3}$ I_3 $\longrightarrow$ I_4 $\longrightarrow$ Bob

$t = 4$

Main Idea: Bruisable Onions

1. ***No global clock:*** when to batch-process onions

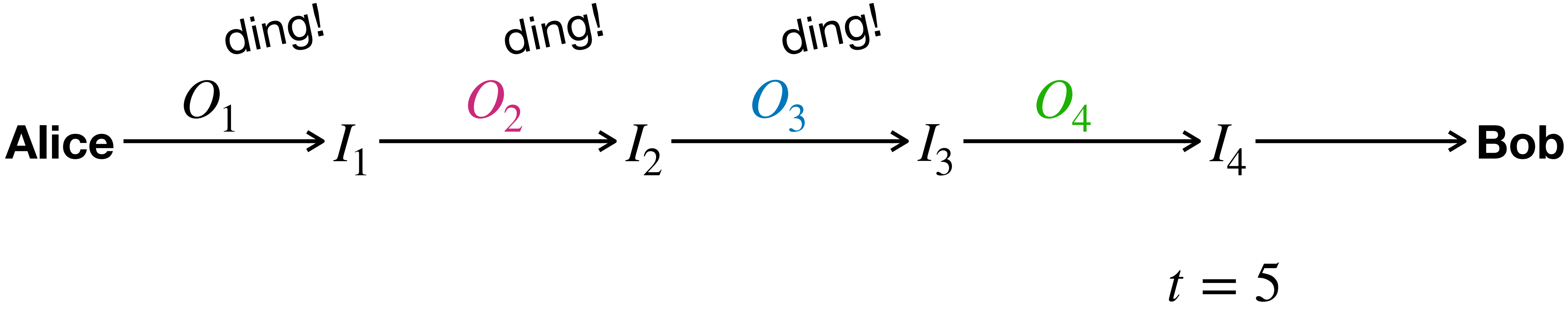
2. ***Timing attacks:*** chronically late onions don't shuffle



Main Idea: Bruisable Onions

1. ***No global clock:*** when to batch-process onions

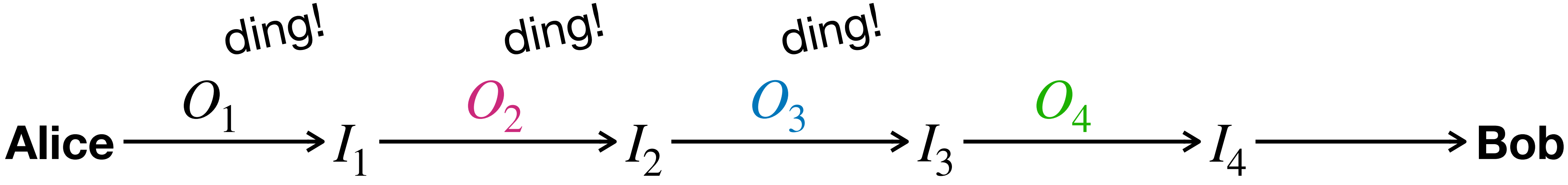
2. ***Timing attacks:*** chronically late onions don't shuffle



Main Idea: Bruisable Onions

1. ***No global clock:*** when to batch-process onions

2. ***Timing attacks:*** chronically late onions don't shuffle



ding! ding! ding!

$$\textbf{Alice} \xrightarrow{O_1} I_1 \xrightarrow{O_2} I_2 \xrightarrow{O_3} I_3 \xrightarrow{O_4} I_4 \longrightarrow \textbf{Bob}$$

$t = 5$

too bruised;
can't be delivered!

Our Contributions

1. Formal definitions for bruisable onion encryption
2. Bruisable onion construction: Tulip Onion Encryption
3. First provably anonymous onion routing protocol for the asynchronous setting: Π_t

Onion Encryption I/O [Camenisch-Lysyanskaya 05]

Onion Encryption I/O [Camenisch-Lysyanskaya 05]

Definition. A standard onion encryption scheme is a triple:

Onion Encryption I/O [Camenisch-Lysyanskaya 05]

Definition. A standard onion encryption scheme is a triple:

- *Key generation algorithm:* $\mathsf{KeyGen}(1^\lambda, P_t) \rightarrow (\mathsf{pk}(P_t), \mathsf{sk}(P_t))$

Onion Encryption I/O [Camenisch-Lysyanskaya 05]

Definition. A standard onion encryption scheme is a triple:

- *Key generation algorithm:* KeyGen$(1^\lambda, P_t) \rightarrow$ (pk(P_t), sk(P_t))
- *Onion forming algorithm:*

  FormOnion$(m, \vec{I} = (I_1, I_2, I_3, I_4, R)$, pk$(\overrightarrow{Q}), (Y_1, Y_2, Y_3, Y_4)))\rightarrow$

  $(O_1, O_2, O_3, O_4, O_5)$

$$\begin{tikzcd} S \arrow[r, "O_1"] & I_1 \arrow[r, "O_2"] & I_2 \arrow[r, "O_3"] & I_3 \arrow[r, "O_4"] & I_4 \arrow[r, "O_5"] & R \end{tikzcd}$$

Onion Encryption I/O [Camenisch-Lysyanskaya 05]

Definition. A standard onion encryption scheme is a triple:

- *Key generation algorithm:* KeyGen$(1^\lambda, P_t) \to$ (pk(P_t), sk(P_t))
- *Onion forming algorithm:*
  FormOnion$(m, \vec{I} = (I_1, I_2, I_3, I_4, R)$, pk$(\overrightarrow{Q})$, $(Y_1, Y_2, Y_3, Y_4)))\to$
  $(O_1, O_2, O_3, O_4, O_5)$
- *Onion processing algorithm:* PeelOnion$(O_i$, sk$(Q_t)) \to (Y_t, O_{t+1}, Q_{t+1})$

$$\begin{tikzcd}
S \arrow[r, "O_1"] & I_1 \arrow[r, "O_2"] & I_2 \arrow[r, "O_3"] & I_3 \arrow[r, "O_4"] & I_4 \arrow[r, "O_5"] & R
\end{tikzcd}$$

Onion Encryption I/O [Camenisch-Lysyanskaya 05]

Definition. A standard onion encryption scheme is a triple:

- *Key generation algorithm:* $\mathsf{KeyGen}(1^\lambda, P_t) \rightarrow (\mathsf{pk}(P_t), \mathsf{sk}(P_t))$
- *Onion forming algorithm:*
  $\mathsf{FormOnion}(m, \vec{I} = (I_1, I_2, I_3, I_4, R), \mathsf{pk}(\overrightarrow{Q}), (Y_1, Y_2, Y_3, Y_4))) \rightarrow$
  $(O_1, O_2, O_3, O_4, O_5)$
- *Onion processing algorithm:* $\mathsf{PeelOnion}(O_i, \mathsf{sk}(Q_t)) \rightarrow (Y_t, O_{t+1}, Q_{t+1})$

$$\begin{tikzcd}
S \arrow[r, "O_1"] & I_1 \arrow[r, "O_2"] & I_2 \arrow[r, "O_3"] & I_3 \arrow[r, "O_4"] & I_4 \arrow[r, "O_5"] & R
\end{tikzcd}$$

$\mathsf{PeelOnion}(O_1, \mathsf{sk}(I_1)) \rightarrow (Y_1, O_2, I_2)$

Onion Encryption I/O [Camenisch-Lysyanskaya 05]

Definition. A standard onion encryption scheme is a triple:

- *Key generation algorithm:* KeyGen$(1^\lambda, P_t) \to$ (pk(P_t), sk(P_t))
- *Onion forming algorithm:*
  FormOnion$(m, \vec{I} = (I_1, I_2, I_3, I_4, R)$, pk$(\overrightarrow{Q}), (Y_1, Y_2, Y_3, Y_4)))\to$
  $(O_1, O_2, O_3, O_4, O_5)$
- *Onion processing algorithm:* PeelOnion$(O_i$, sk$(Q_t)) \to (Y_t, O_{t+1}, Q_{t+1})$

$$\begin{tikzcd} S \arrow[r, "O_1"] & I_1 \arrow[r, "O_2"] & I_2 \arrow[r, "O_3"] & I_3 \arrow[r, "O_4"] & I_4 \arrow[r, "O_5"] & R \end{tikzcd}$$

PeelOnion$(O_2$, sk$(I_2)) \to (Y_2, O_3, I_3)$

Contribution 1.

Game-based security definition

Bruisable Onion Encryption: I/O

Bruisable Onion Encryption: I/O

Definition. A bruisable onion encryption scheme is a quadruple:

Bruisable Onion Encryption: I/O

Definition. A bruisable onion encryption scheme is a quadruple:

- KeyGen$(1^\lambda, P_t) \rightarrow (\mathsf{pk}(P_t), \mathsf{sk}(P_t))$

Bruisable Onion Encryption: I/O

Definition. A bruisable onion encryption scheme is a quadruple:

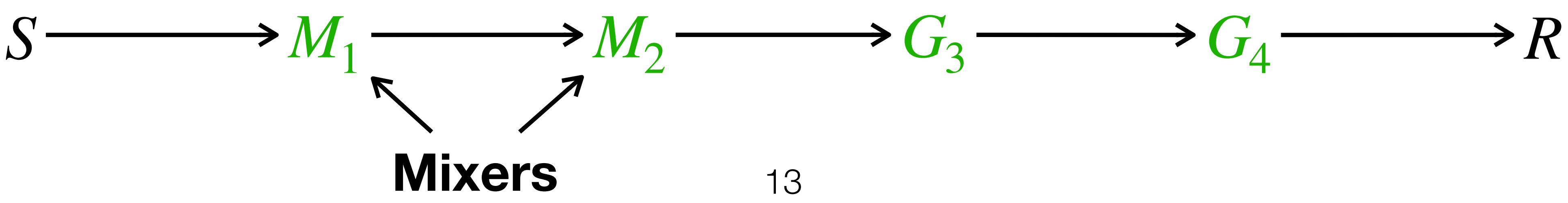
- KeyGen$(1^\lambda, P_t) \to$ (pk(P_t), sk(P_t))
- FormOnion$(m, \overrightarrow{Q} = (M_1, M_2, G_3, G_4, R)$, pk$(\overrightarrow{Q})$, $(Y_1, Y_2, Y_3, Y_4)) \to$ $(\overrightarrow{O}_1, \overrightarrow{O}_2, \overrightarrow{O}_3, \overrightarrow{O}_4, \overrightarrow{O}_5)$

$$\begin{tikzcd} S \arrow[r] & M_1 \arrow[r] & M_2 \arrow[r] & G_3 \arrow[r] & G_4 \arrow[r] & R \end{tikzcd}$$

Bruisable Onion Encryption: I/O

Definition. A bruisable onion encryption scheme is a quadruple:

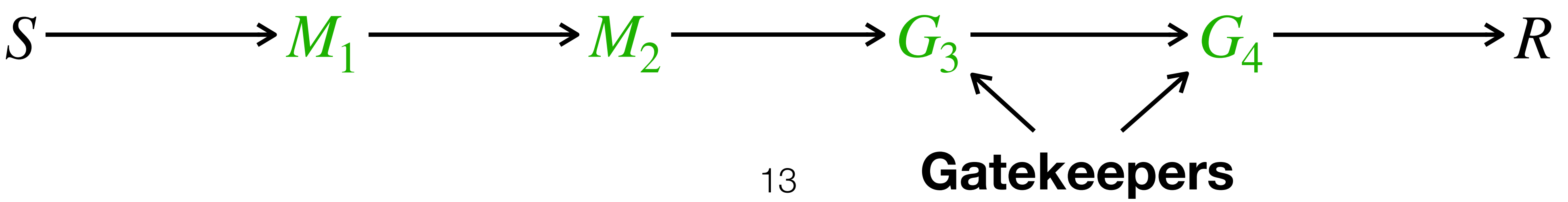
- KeyGen$(1^\lambda, P_t) \rightarrow$ (pk(P_t), sk(P_t))
- FormOnion$(m, \overrightarrow{Q} = (M_1, M_2, G_3, G_4, R)$, pk$(\overrightarrow{Q})$, $(Y_1, Y_2, Y_3, Y_4)) \rightarrow$ $(\overrightarrow{O}_1, \overrightarrow{O}_2, \overrightarrow{O}_3, \overrightarrow{O}_4, \overrightarrow{O}_5)$



Bruisable Onion Encryption: I/O

Definition. A bruisable onion encryption scheme is a quadruple:

- KeyGen$(1^\lambda, P_t) \to$ (pk(P_t), sk(P_t))
- FormOnion$(m, \overrightarrow{Q} = (M_1, M_2, G_3, G_4, R)$, pk$(\overrightarrow{Q})$, $(Y_1, Y_2, Y_3, Y_4)) \to$ $(\overrightarrow{O}_1, \overrightarrow{O}_2, \overrightarrow{O}_3, \overrightarrow{O}_4, \overrightarrow{O}_5)$



Bruisable Onion Encryption: I/O

Definition. A bruisable onion encryption scheme is a quadruple:

- KeyGen$(1^\lambda, P_t) \to (\mathsf{pk}(P_t), \mathsf{sk}(P_t))$
- FormOnion$(m, \overrightarrow{Q} = (M_1, M_2, G_3, G_4, R), \mathsf{pk}(\overrightarrow{Q}), (Y_1, Y_2, Y_3, Y_4)) \to (\overrightarrow{O}_1, \overrightarrow{O}_2, \overrightarrow{O}_3, \overrightarrow{O}_4, \overrightarrow{O}_5)$
- PeelOnion$(O_t, \mathsf{sk}(Q_t)) \to (t, Y_t, O_{t+1}, Q_{t+1})$

$$\begin{tikzcd} S \arrow[r] & M_1 \arrow[r] & M_2 \arrow[r] & G_3 \arrow[r] & G_4 \arrow[r] & R \end{tikzcd}$$

Bruisable Onion Encryption: I/O

Definition. A bruisable onion encryption scheme is a quadruple:

- KeyGen$(1^\lambda, P_t) \rightarrow$ (pk(P_t), sk(P_t))
- FormOnion$(m, \overrightarrow{Q} = (M_1, M_2, G_3, G_4, R)$, pk$(\overrightarrow{Q})$, $(Y_1, Y_2, Y_3, Y_4)) \rightarrow$ $(\overrightarrow{O}_1, \overrightarrow{O}_2, \overrightarrow{O}_3, \overrightarrow{O}_4, \overrightarrow{O}_5)$
- PeelOnion$(O_t$, sk$(Q_t)) \rightarrow (t, Y_t, O_{t+1}, Q_{t+1})$
- *Onion bruising algorithm:* BruiseOnion$(O_t$, sk$(Q_t)) \rightarrow O_{t+1}$

$$S \longrightarrow M_1 \longrightarrow M_2 \longrightarrow G_3 \longrightarrow G_4 \longrightarrow R$$

Bruisable Onion Encryption: I/O

Definition. A bruisable onion encryption scheme is a quadruple:

- $\mathsf{KeyGen}(1^\lambda, P_t) \rightarrow (\mathsf{pk}(P_t), \mathsf{sk}(P_t))$
- $\mathsf{FormOnion}(m, \overrightarrow{Q} = (M_1, M_2, G_3, G_4, R), \mathsf{pk}(\overrightarrow{Q}), (Y_1, Y_2, Y_3, Y_4)) \rightarrow (\overrightarrow{O}_1, \overrightarrow{O}_2, \overrightarrow{O}_3, \overrightarrow{O}_4, \overrightarrow{O}_5)$
- $\mathsf{PeelOnion}(O_t, \mathsf{sk}(Q_t)) \rightarrow (t, Y_t, O_{t+1}, Q_{t+1})$
- *Onion bruising algorithm:* $\mathsf{BruiseOnion}(O_t, \mathsf{sk}(Q_t)) \rightarrow O_{t+1}$

$O_1 \in (O_1)$

$$\begin{tikzcd}
S \arrow[r] & M_1 \arrow[r] & M_2 \arrow[r] & G_3 \arrow[r] & G_4 \arrow[r] & R
\end{tikzcd}$$

$\mathsf{PeelOnion}(O_{1,0}, \mathsf{sk}(M_1)) \rightarrow (1, Y_1, O_{2,0}, P_2)$

Bruisable Onion Encryption: I/O

Definition. A bruisable onion encryption scheme is a quadruple:

- KeyGen$(1^\lambda, P_t) \rightarrow$ (pk(P_t), sk(P_t))
- FormOnion$(m, \overrightarrow{Q} = (M_1, M_2, G_3, G_4, R)$, pk$(\overrightarrow{Q})$, $(Y_1, Y_2, Y_3, Y_4)) \rightarrow (\overrightarrow{O}_1, \overrightarrow{O}_2, \overrightarrow{O}_3, \overrightarrow{O}_4, \overrightarrow{O}_5)$
- PeelOnion$(O_t$, sk$(Q_t)) \rightarrow (t, Y_t, O_{t+1}, Q_{t+1})$
- *Onion bruising algorithm:* BruiseOnion$(O_t$, sk$(Q_t)) \rightarrow O_{t+1}$

$$\begin{tikzcd} S \arrow[r] & M_1 \arrow[r, "O_{2,0} \in \begin{pmatrix} O_{2,0} \\ O_{2,1} \end{pmatrix}"] & M_2 \arrow[r] & G_3 \arrow[r] & G_4 \arrow[r] & R \end{tikzcd}$$

PeelOnion$(O_{2,0}$, sk$(M_2)) \rightarrow (2, Y_2, O_{3,0}, P_3)$

Bruisable Onion Encryption: I/O

Definition. A bruisable onion encryption scheme is a quadruple:

- $\mathsf{KeyGen}(1^\lambda, P_t) \to (\mathsf{pk}(P_t), \mathsf{sk}(P_t))$
- $\mathsf{FormOnion}(m, \overrightarrow{Q} = (M_1, M_2, G_3, G_4, R), \mathsf{pk}(\overrightarrow{Q}), (Y_1, Y_2, Y_3, Y_4)) \to (\overrightarrow{O}_1, \overrightarrow{O}_2, \overrightarrow{O}_3, \overrightarrow{O}_4, \overrightarrow{O}_5)$
- $\mathsf{PeelOnion}(O_t, \mathsf{sk}(Q_t)) \to (t, Y_t, O_{t+1}, Q_{t+1})$
- *Onion bruising algorithm:* $\mathsf{BruiseOnion}(O_t, \mathsf{sk}(Q_t)) \to O_{t+1}$

$$\begin{tikzcd}
S \arrow[r] & M_1 \arrow[r] & M_2 \arrow[r, "O_{3,1} \in \begin{pmatrix} O_{3,0} \\ O_{3,1} \\ O_{3,2} \end{pmatrix}"] & G_3 \arrow[r, "O_{4,1} \in \begin{pmatrix} O_{4,0} \\ O_{4,1} \\ O_{4,2} \end{pmatrix}"] & G_4 \arrow[r, "O_5 \in \left(O_5\right)"] & R
\end{tikzcd}$$

$\mathsf{BruiseOnion}(O_{2,0}, \mathsf{sk}(M_2)) \to O_{3,1}$

Bruisable Onion Encryption: Security

Bruisable Onion Encryption: Security

Intuition. Information behind an honest party remains hidden, including bruise count

Bruisable Onion Encryption: Security

Intuition. Information behind an honest party remains hidden, including bruise count

Informal Definition. Adversary cannot distinguish between:

1. $O_{1,0}^{b=0}$ formed on all onion parameters:

   - Message m
   - Complete path (M_1, M_2, G_3, G_4, R) and associated keys and metadata

Bruisable Onion Encryption: Security

Intuition. Information behind an honest party remains hidden, including bruise count

Informal Definition. Adversary cannot distinguish between:

1. $O_{1,0}^{b=0}$ formed on all onion parameters:
    - Message m
    - Complete path (M_1, M_2, G_3, G_4, R) and associated keys and metadata
2. $O_{1,0}^{b=1}$ formed without any information after honest intermediary, e.g., M_1:
    - Dummy message $\perp$
    - Partial path, e.g, $(M_1, \perp, \perp, \perp, \perp)$, and associated keys and metadata

Bruisable Onion Encryption: Security

Intuition. Information behind an honest party remains hidden, including bruise count

Informal Definition. Adversary cannot distinguish between:

challenge bit

1. $O_{1,0}^{b=0}$ formed on all onion parameters:
    - Message m
    - Complete path (M_1, M_2, G_3, G_4, R) and associated keys and metadata
2. $O_{1,0}^{b=1}$ formed without any information after honest intermediary, e.g., M_1:
    - Dummy message $\perp$
    - Partial path, e.g, $(M_1, \perp, \perp, \perp, \perp)$, and associated keys and metadata

Bruisable Onion Encryption: Security

Intuition. Information behind an honest party remains hidden, including bruise count

Informal Definition. Adversary cannot distinguish between:

challenge bit

1. $O^{b=0}_{1,0}$ formed on all onion parameters:

onion layer

- Message m
- Complete path (M_1, M_2, G_3, G_4, R) and associated keys and metadata

2. $O^{b=1}_{1,0}$ formed without any information after honest intermediary, e.g., M_1:

- Dummy message $\perp$
- Partial path, e.g, $(M_1, \perp, \perp, \perp, \perp)$, and associated keys and metadata

Bruisable Onion Encryption: Security

Intuition. Information behind an honest party remains hidden, including bruise count

Informal Definition. Adversary cannot distinguish between:

1. $O_{2,n}^{b=0}$ formed on all onion parameters:
   - Message m
   - Complete path (M_1, M_2, G_3, G_4, R) and associated keys and metadata
2. $O_{2,0}^{b=1}$ formed without rest of the routing path, e.g., before M_1 or after G_3:
   - Dummy message $\perp$
   - Partial path, e.g, $(\perp, M_2, G_3, \perp, \perp)$, and associated keys and metadata

Bruisable Onion Encryption: Security

Intuition. Information behind an honest party remains hidden, including bruise count

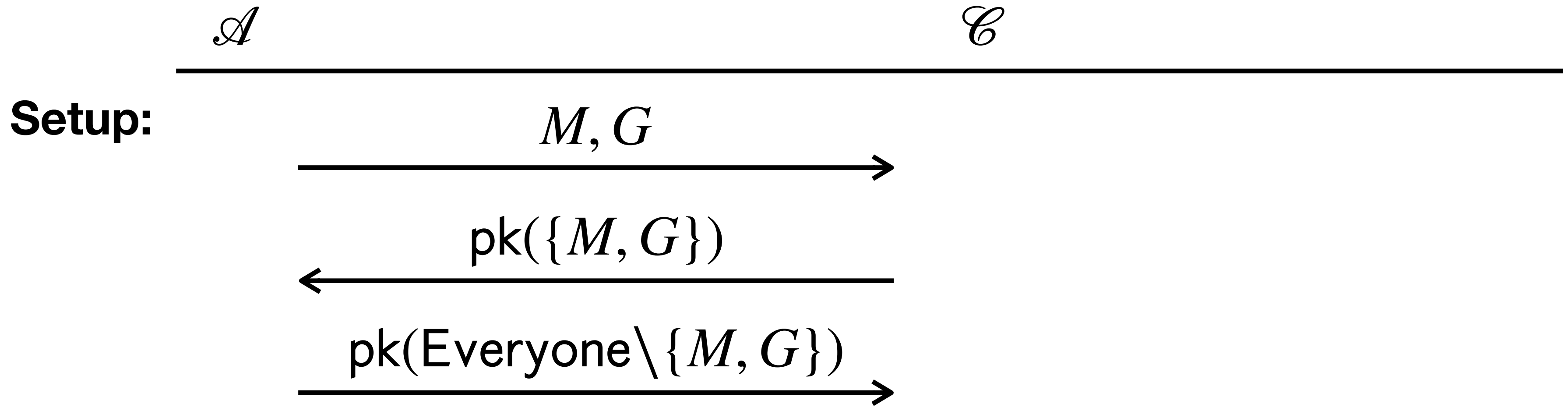
Informal Definition. Adversary cannot distinguish between:

1. $O_{2,n}^{b=0}$ formed on all onion parameters:
   ↖ bruise count
   - Message m
   - Complete path (M_1, M_2, G_3, G_4, R) and associated keys and metadata

2. $O_{2,0}^{b=1}$ formed without rest of the routing path, e.g., before M_1 or after G_3:
   - Dummy message $\perp$
   - Partial path, e.g, $(\perp, M_2, G_3, \perp, \perp)$, and associated keys and metadata

Bruisable Onion Encryption: Security

Formal Definition. CCA2-like, defined w.r.t. Onion Security Game:



$\mathscr{A}$ $\mathscr{C}$

Setup:

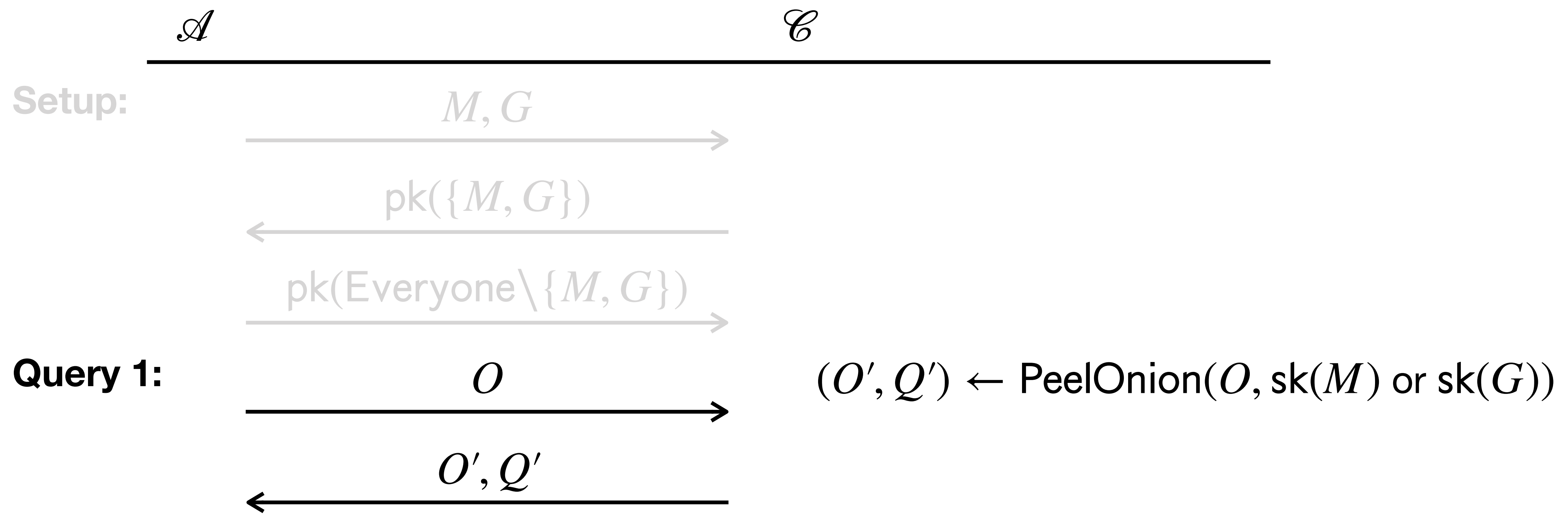
M, G

pk$(\{M, G\})$

pk(Everyone$\setminus\{M, G\}$)

Bruisable Onion Encryption: Security

Formal Definition. CCA2-like, defined w.r.t. Onion Security Game:



$\mathscr{A}$ $\mathscr{C}$

Setup:

$\xrightarrow{M, G}$

$\xleftarrow{\mathsf{pk}(\{M, G\})}$

$\xrightarrow{\mathsf{pk}(\mathsf{Everyone} \setminus \{M, G\})}$

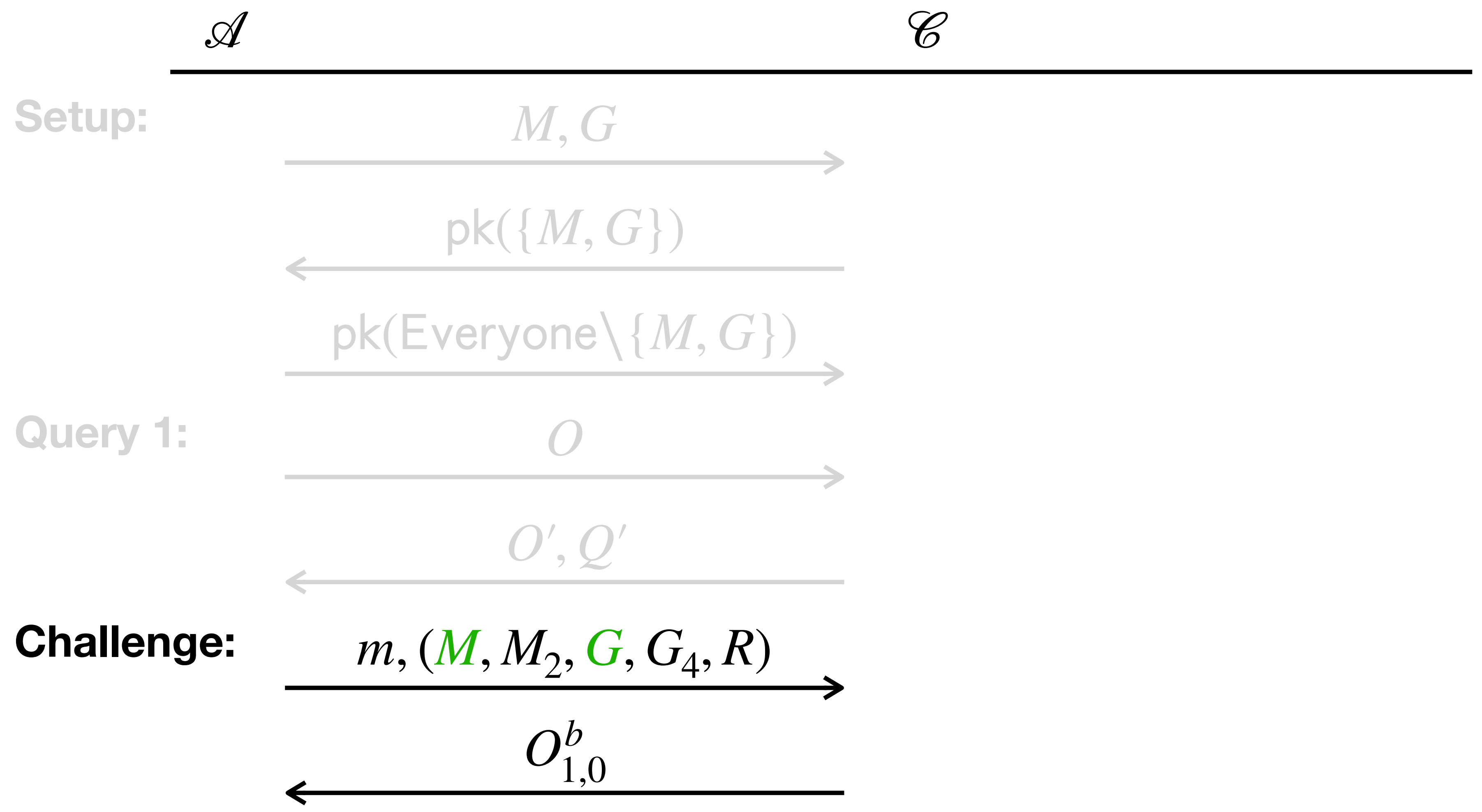
Query 1:

$\xrightarrow{O}$ $(O', Q') \leftarrow \mathsf{PeelOnion}(O, \mathsf{sk}(M) \text{ or } \mathsf{sk}(G))$

$\xleftarrow{O', Q'}$

Bruisable Onion Encryption: Security

Formal Definition. CCA2-like, defined w.r.t. Onion Security Game:



$\mathscr{A}$ $\mathscr{C}$

Setup: M, G →

← $\mathsf{pk}(\{M, G\})$

$\mathsf{pk}(\text{Everyone} \setminus \{M, G\})$ →

Query 1: O →

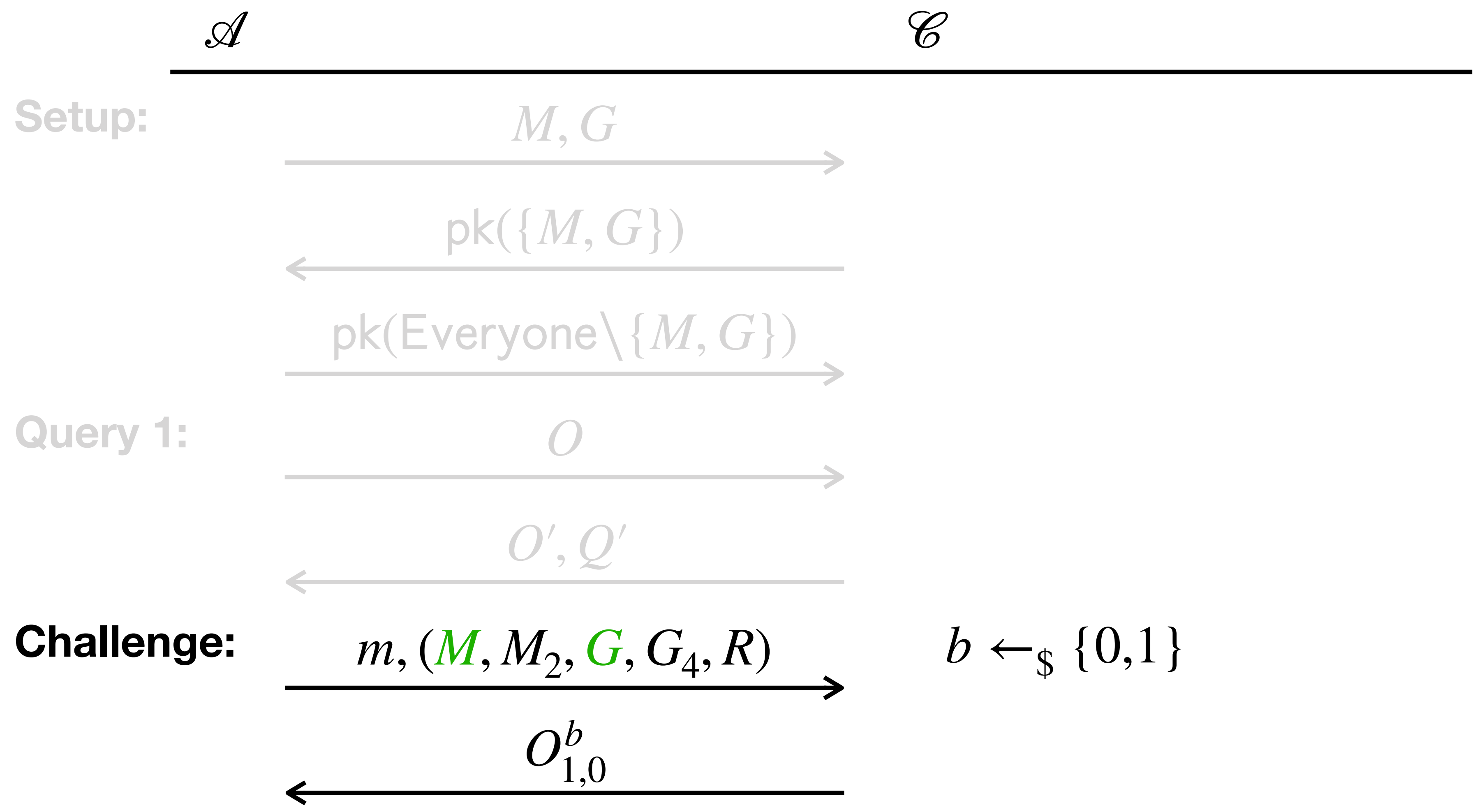
← O', Q'

Challenge: $m, (M, M_2, G, G_4, R)$ →

← $O^b_{1,0}$

Bruisable Onion Encryption: Security

Formal Definition. CCA2-like, defined w.r.t. Onion Security Game:



$\mathscr{A}$ $\mathscr{C}$

Setup: $\xrightarrow{M, G}$

$\xleftarrow{\text{pk}(\{M, G\})}$

$\xrightarrow{\text{pk}(\text{Everyone} \setminus \{M, G\})}$

Query 1: $\xrightarrow{O}$

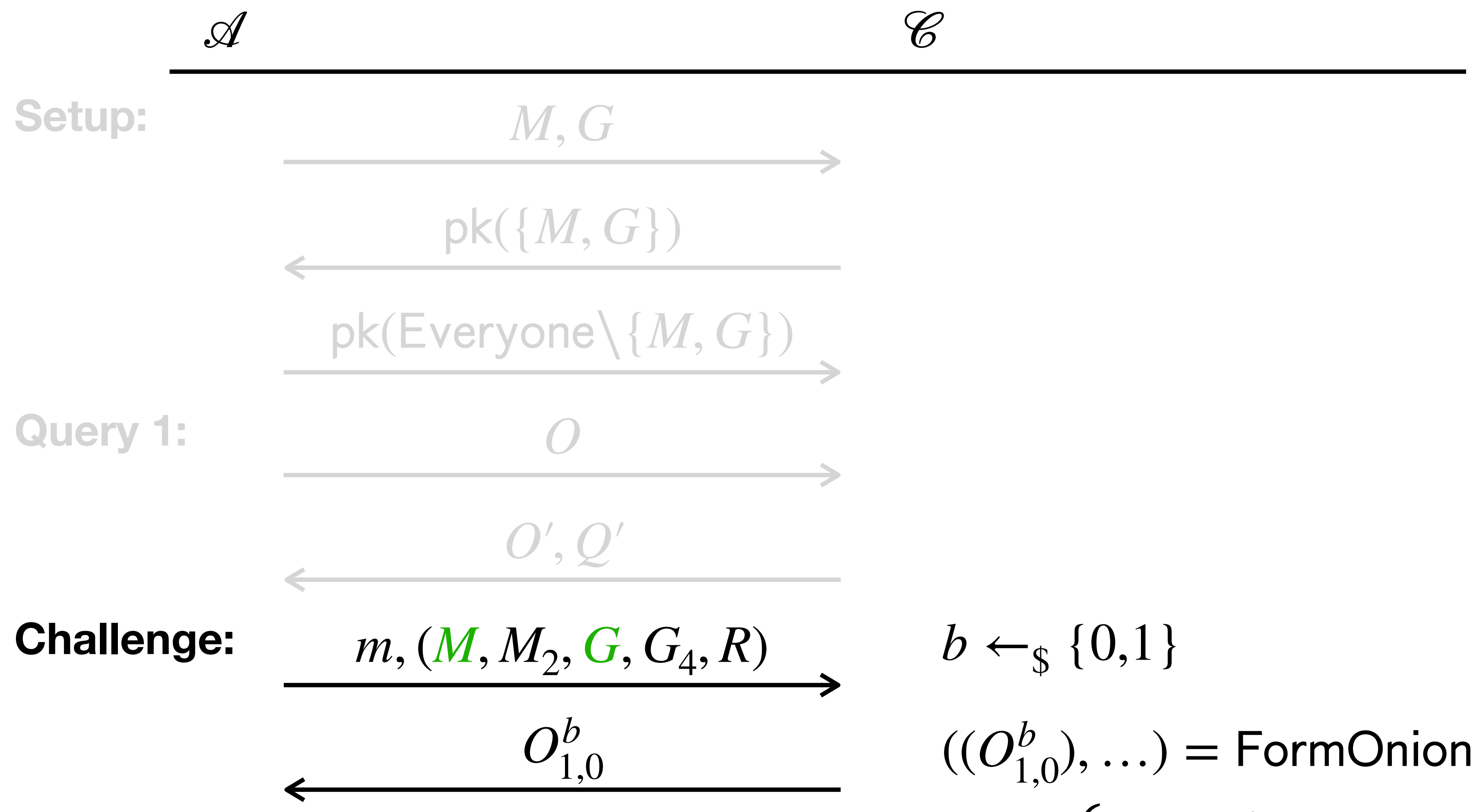
$\xleftarrow{O', Q'}$

Challenge: $\xrightarrow{m, (M, M_2, G, G_4, R)}$ $b \leftarrow_{\$} \{0,1\}$

$\xleftarrow{O^b_{1,0}}$

Bruisable Onion Encryption: Security

Formal Definition. CCA2-like, defined w.r.t. Onion Security Game:



$\mathscr{A}$ $\mathscr{C}$

Setup:

M, G →

← pk($\{M, G\}$)

pk(Everyone$\setminus\{M, G\}$) →

Query 1:

O →

← O', Q'

Challenge:

$m, (M, M_2, G, G_4, R)$ →

← $O^b_{1,0}$

$b \leftarrow_\$ \{0,1\}$

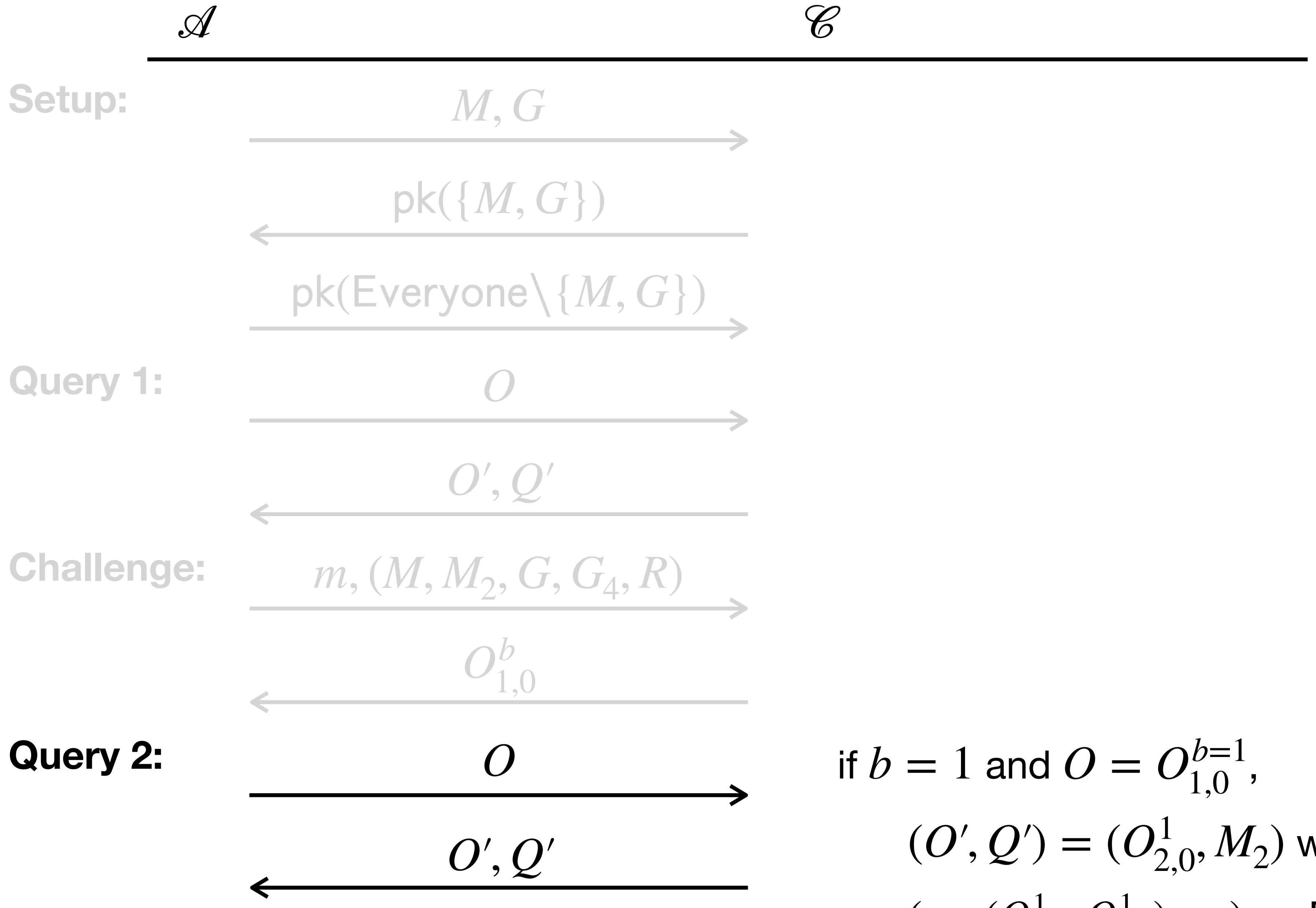
$((O^b_{1,0}), \ldots) =$ FormOnion($m_b, \overrightarrow{Q}_b$) where

$$m_b = \begin{cases} m, & b = 0 \\ \perp, & b = 1 \end{cases}$$

$$\overrightarrow{Q}_b = \begin{cases} (M, M_2, G, G_4, R), & b = 0 \\ (M, \perp, \perp, \perp, \perp), & b = 1 \end{cases}$$

Bruisable Onion Encryption: Security

Formal Definition. CCA2-like, defined w.r.t. Onion Security Game:



$\mathscr{A}$ $\mathscr{C}$

Setup:

M, G

pk($\{M, G\}$)

pk(Everyone$\setminus\{M, G\}$)

Query 1:

O

O', Q'

Challenge:

$m, (M, M_2, G, G_4, R)$

$O^b_{1,0}$

Query 2:

O

O', Q'

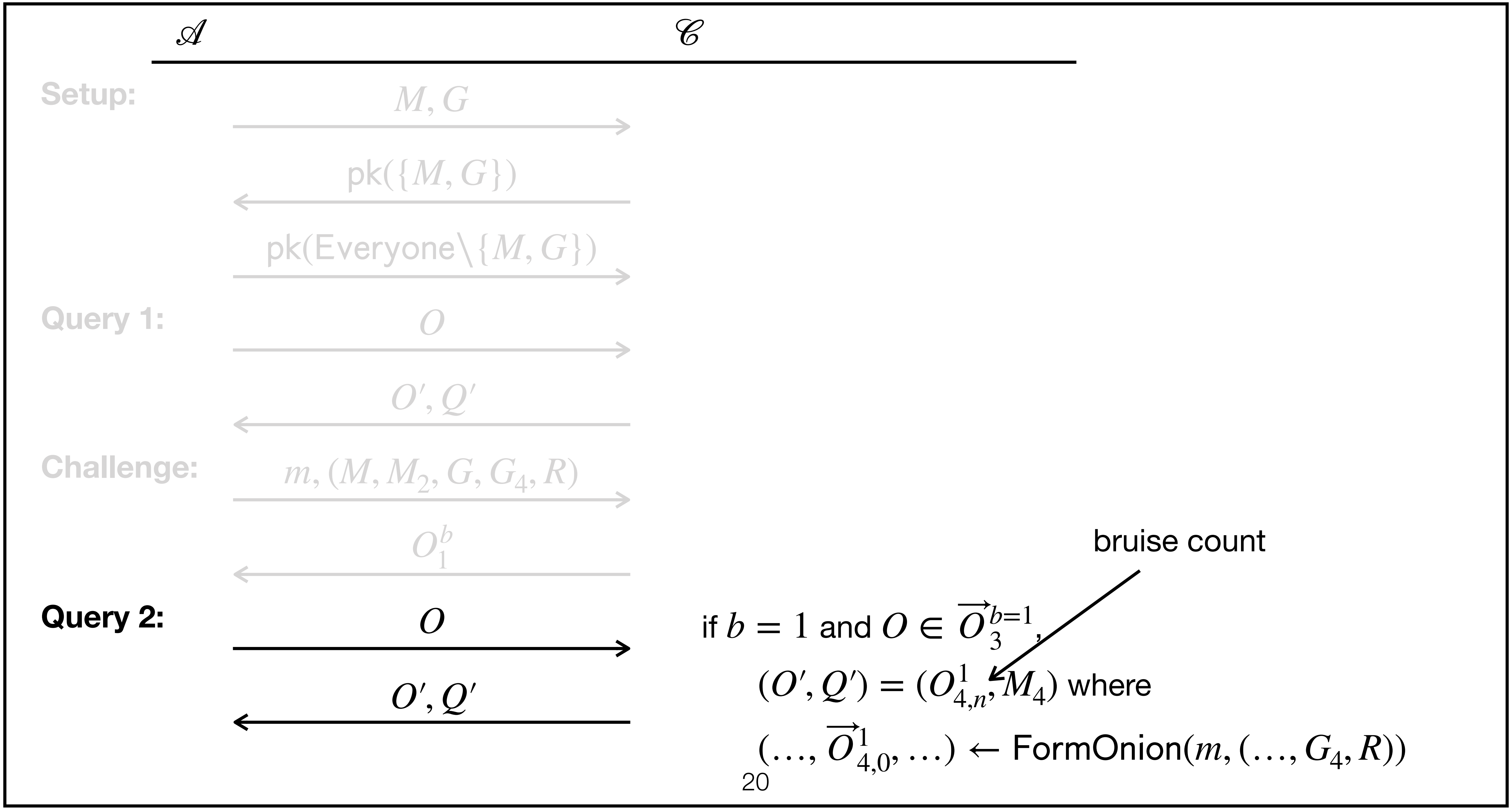
if $b = 1$ and $O = O^{b=1}_{1,0}$,

$(O', Q') = (O^1_{2,0}, M_2)$ where

$(\ldots, (O^1_{2,0}, O^1_{2,1}), \ldots) \leftarrow$ FormOnion($\perp$, ($\perp$, $M_2, G, \ldots$))

Bruisable Onion Encryption: Security

Formal Definition. CCA2-like, defined w.r.t. Onion Security Game:



$\mathcal{A}$ $\mathcal{C}$

Setup: $\xrightarrow{M, G}$

$\xleftarrow{\text{pk}(\{M, G\})}$

$\xrightarrow{\text{pk}(\text{Everyone}\setminus\{M, G\})}$

Query 1: $\xrightarrow{O}$

$\xleftarrow{O', Q'}$

Challenge: $\xrightarrow{m, (M, M_2, G, G_4, R)}$

$\xleftarrow{O_1^b}$

Query 2: $\xrightarrow{O}$

$\xleftarrow{O', Q'}$

if $b = 1$ and $O \in \overrightarrow{O}_3^{b=1}$,

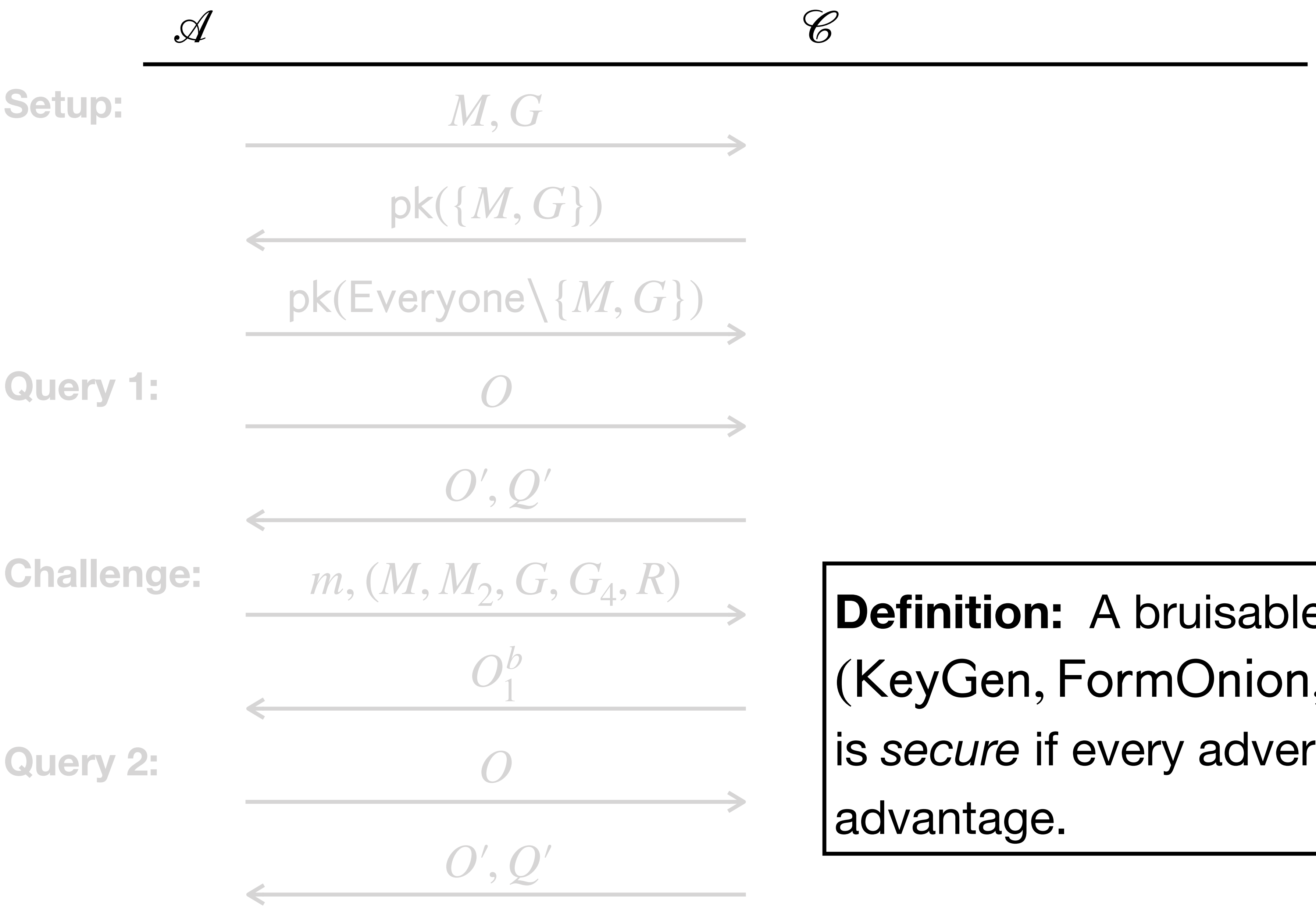
$(O', Q') = (O_{4,n}^1, M_4)$ where

$(\ldots, \overrightarrow{O}_{4,0}^1, \ldots) \leftarrow$ FormOnion$(m, (\ldots, G_4, R))$

bruise count

Bruisable Onion Encryption: Security

Formal Definition. CCA2-like, defined w.r.t. Onion Security Game:



$\mathscr{A}$ $\mathscr{C}$

Setup:

$\xrightarrow{M, G}$

$\xleftarrow{\mathsf{pk}(\{M, G\})}$

$\xrightarrow{\mathsf{pk}(\text{Everyone} \setminus \{M, G\})}$

Query 1:

$\xrightarrow{O}$

$\xleftarrow{O', Q'}$

Challenge:

$\xrightarrow{m, (M, M_2, G, G_4, R)}$

$\xleftarrow{O_1^b}$

Query 2:

$\xrightarrow{O}$

$\xleftarrow{O', Q'}$

Definition: A bruisable onion encryption scheme (KeyGen, FormOnion, PeelOnion, BruiseOnion) is *secure* if every adversary wins with negligible advantage.

Output guess b' and wins if $b' = b$

Contribution 2.

Our construction: Tulip Encryption Scheme

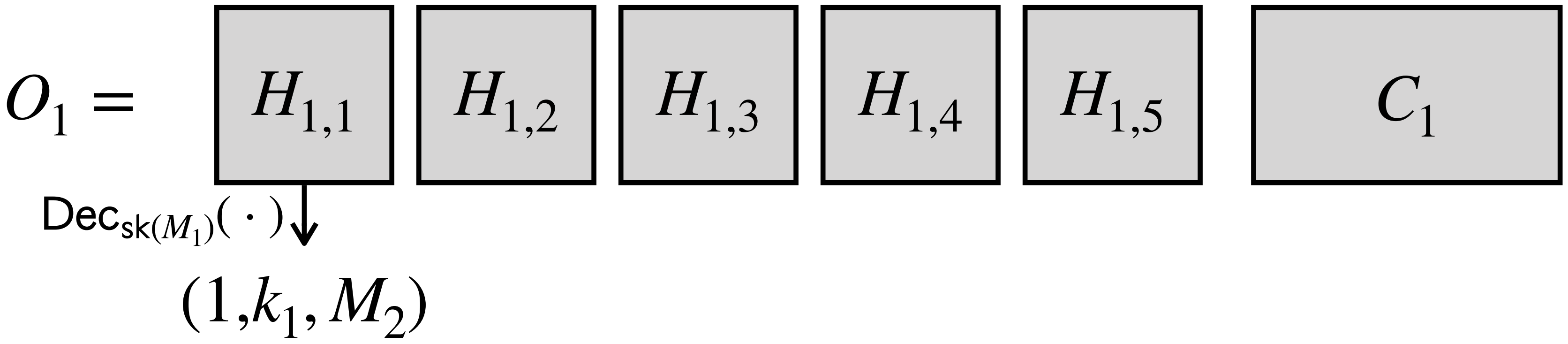
Tulip Onion Encryption

$$O_1 = \boxed{H_{1,1}}\ \boxed{H_{1,2}}\ \boxed{H_{1,3}}\ \boxed{H_{1,4}}\ \boxed{H_{1,5}}\quad \boxed{\quad C_1 \quad}$$

header blocks = encrypted path *content* = encrypted payload

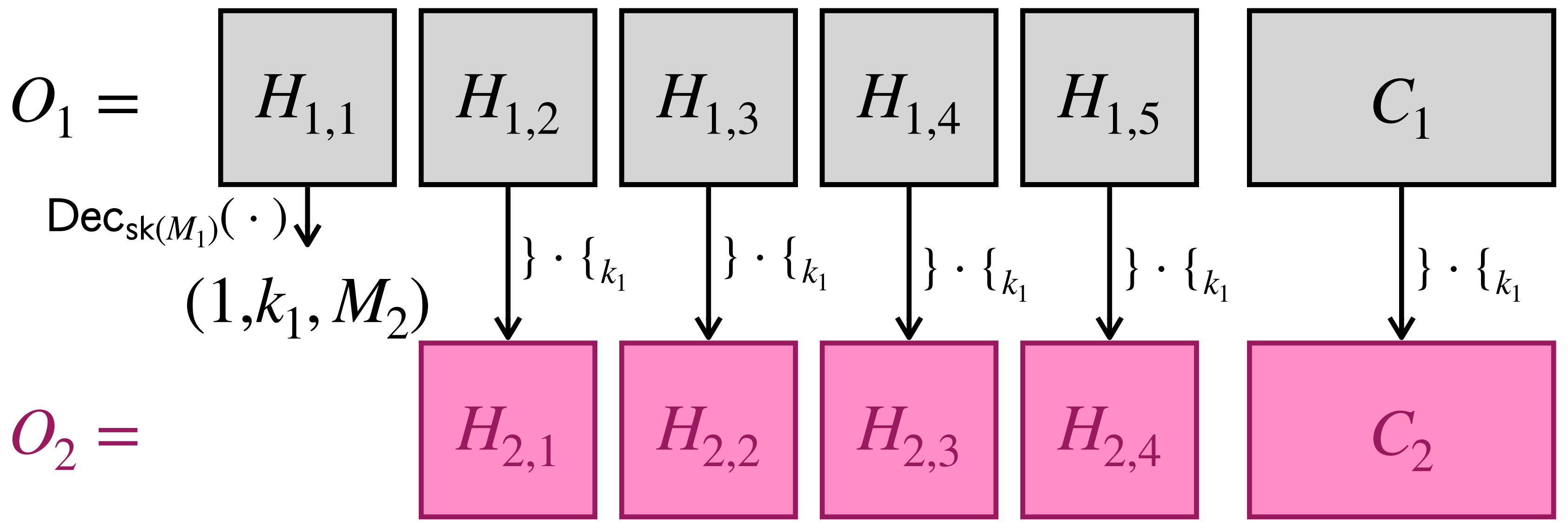
$$\textbf{Alice} \xrightarrow{O_1} M_1 \xrightarrow{O_2} M_2 \xrightarrow{O_3} G_3 \xrightarrow{O_4} G_4 \xrightarrow{O_5} \textbf{Bob} \quad m$$

Tulip Onion Encryption



$$\text{Alice} \xrightarrow{O_1} M_1 \xrightarrow{O_2} M_2 \xrightarrow{O_3} G_3 \xrightarrow{O_4} G_4 \xrightarrow{O_5} \text{Bob} \quad m$$

Tulip Onion Encryption



$O_1 =$ $H_{1,1}$ $H_{1,2}$ $H_{1,3}$ $H_{1,4}$ $H_{1,5}$ C_1

$\mathsf{Dec}_{\mathsf{sk}(M_1)}(\cdot)$

$(1, k_1, M_2)$

$\}\cdot\{_{k_1}$ $\}\cdot\{_{k_1}$ $\}\cdot\{_{k_1}$ $\}\cdot\{_{k_1}$ $\}\cdot\{_{k_1}$

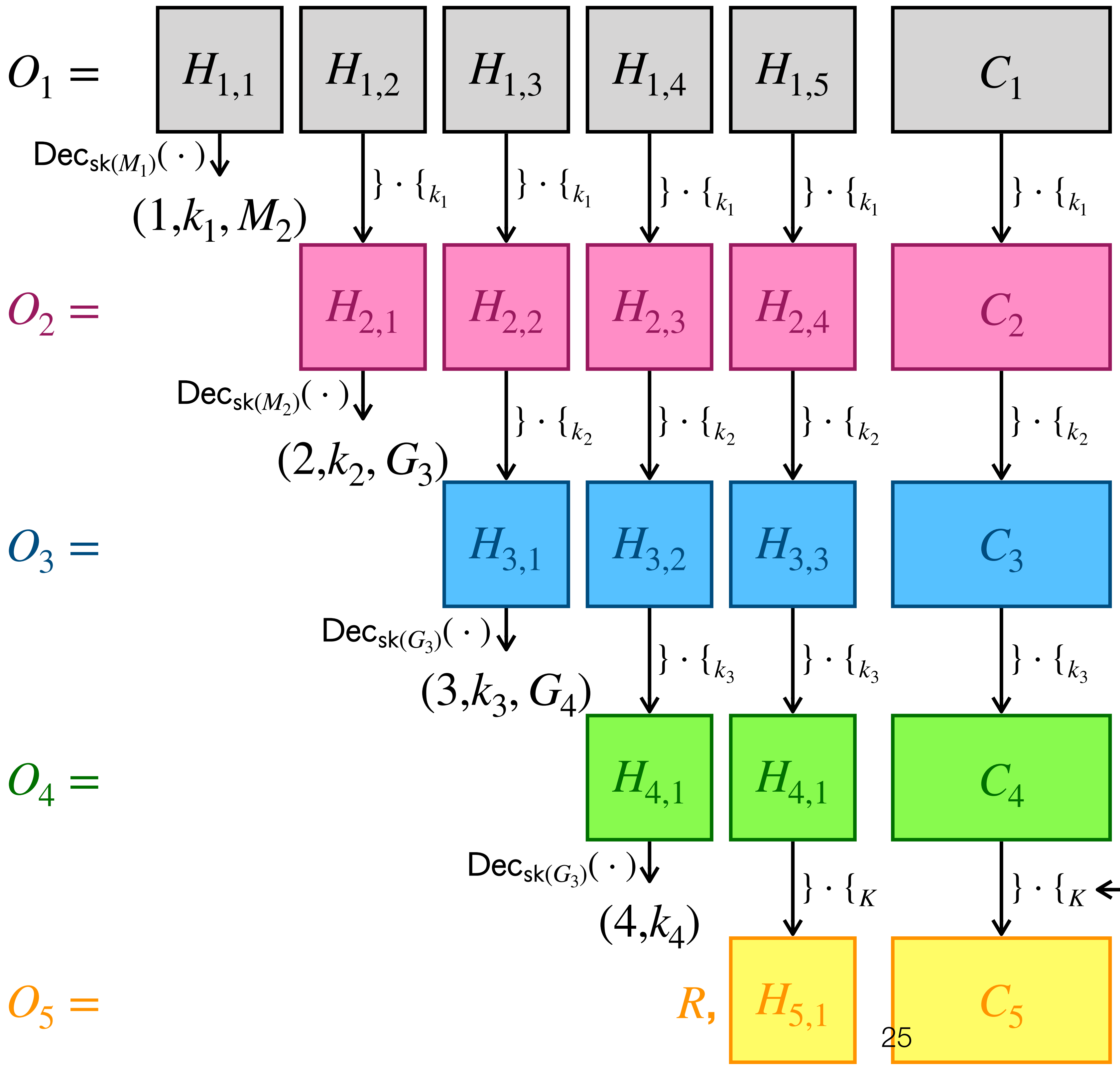
$O_2 =$ $H_{2,1}$ $H_{2,2}$ $H_{2,3}$ $H_{2,4}$ C_2

Notation:

$\}\cdot\{_{\text{key}}$ = decryption under key

Alice $\xrightarrow{O_1} M_1 \xrightarrow{O_2} M_2 \xrightarrow{O_3} G_3 \xrightarrow{O_4} G_4 \xrightarrow{O_5}$ **Bob** m

Tulip Onion Encryption



$O_1 =$ $H_{1,1}$ $H_{1,2}$ $H_{1,3}$ $H_{1,4}$ $H_{1,5}$ C_1

$\text{Dec}_{\text{sk}(M_1)}(\cdot)$ $(1, k_1, M_2)$ $\}\cdot\{_{k_1}$ $\}\cdot\{_{k_1}$ $\}\cdot\{_{k_1}$ $\}\cdot\{_{k_1}$ $\}\cdot\{_{k_1}$

$O_2 =$ $H_{2,1}$ $H_{2,2}$ $H_{2,3}$ $H_{2,4}$ C_2

$\text{Dec}_{\text{sk}(M_2)}(\cdot)$ $(2, k_2, G_3)$ $\}\cdot\{_{k_2}$ $\}\cdot\{_{k_2}$ $\}\cdot\{_{k_2}$ $\}\cdot\{_{k_2}$

$O_3 =$ $H_{3,1}$ $H_{3,2}$ $H_{3,3}$ C_3

$\text{Dec}_{\text{sk}(G_3)}(\cdot)$ $(3, k_3, G_4)$ $\}\cdot\{_{k_3}$ $\}\cdot\{_{k_3}$ $\}\cdot\{_{k_3}$

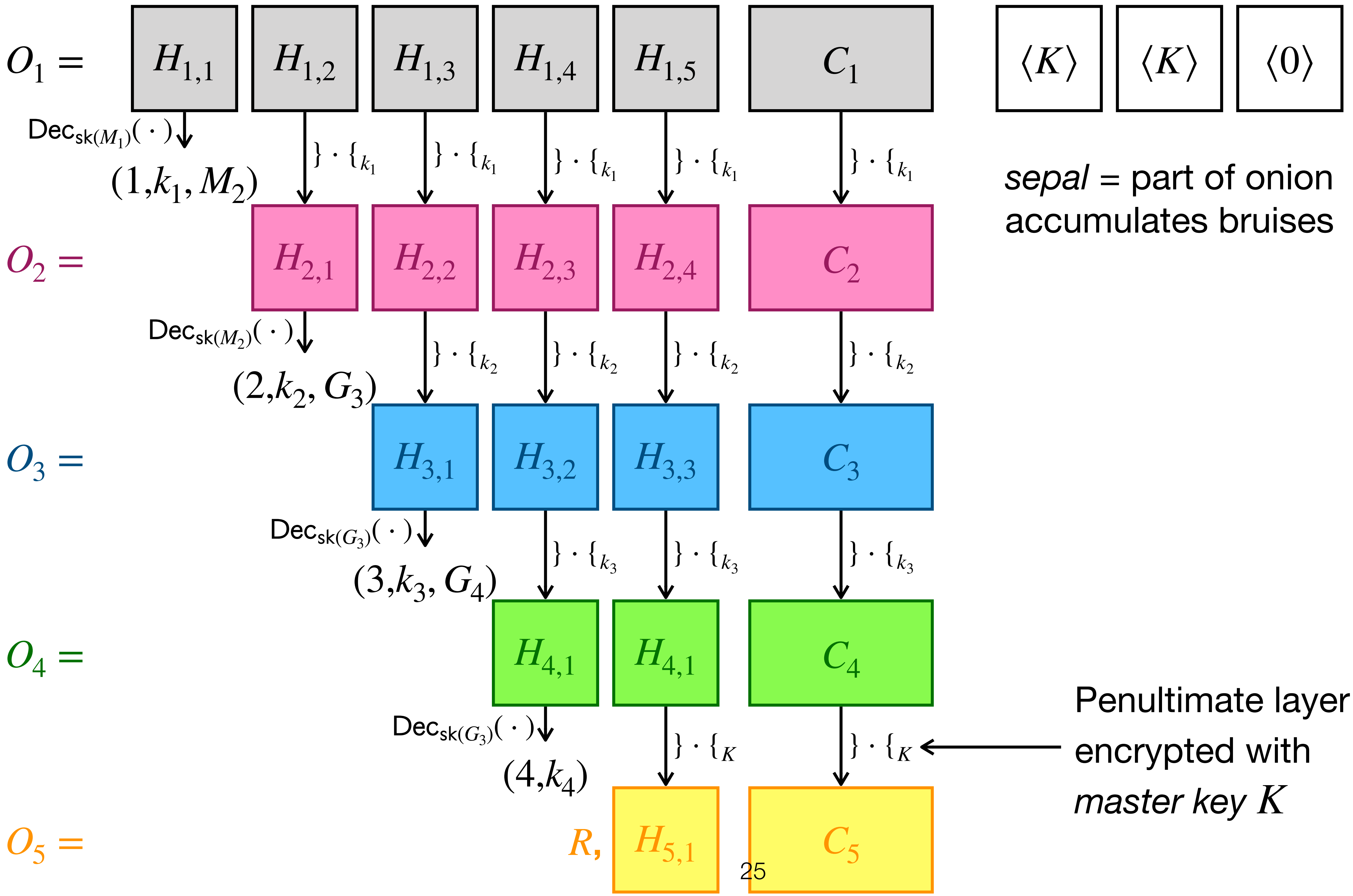
$O_4 =$ $H_{4,1}$ $H_{4,1}$ C_4

$\text{Dec}_{\text{sk}(G_3)}(\cdot)$ $(4, k_4)$ $\}\cdot\{_K$ $\}\cdot\{_K$

$O_5 =$ $R,$ $H_{5,1}$ C_5

Penultimate layer encrypted with *master key* K

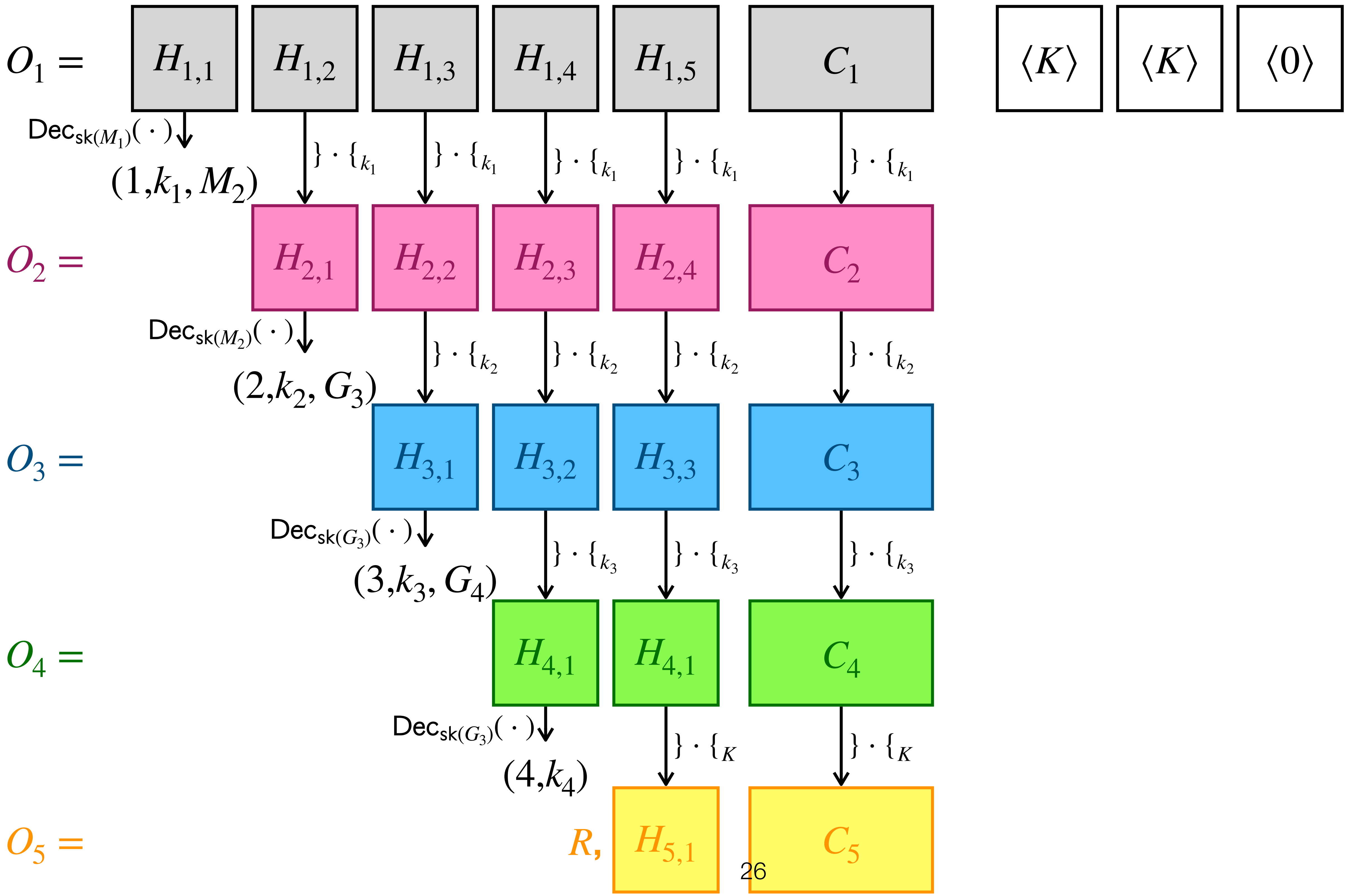
Tulip Onion Encryption



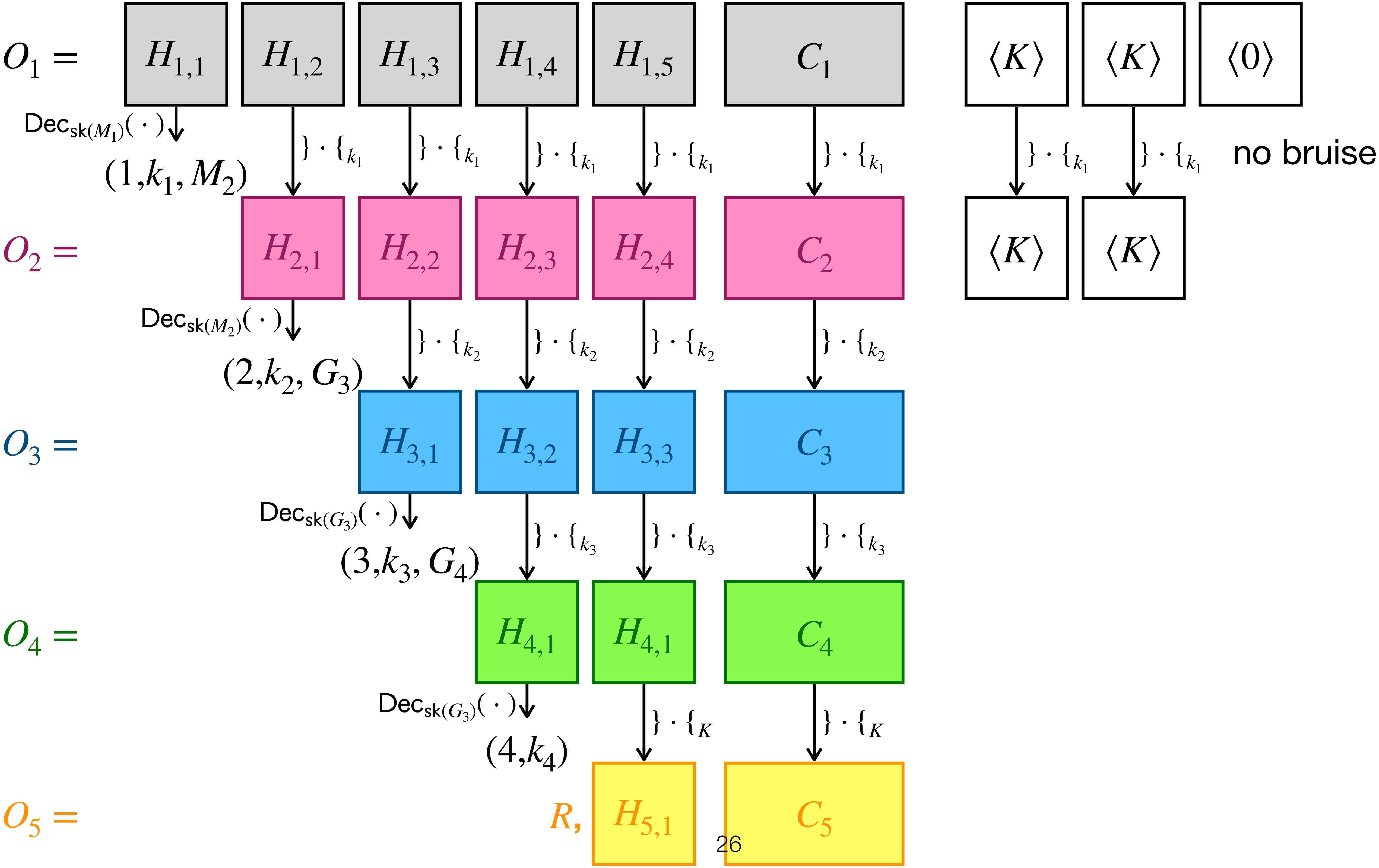
sepal = part of onion accumulates bruises

Penultimate layer encrypted with *master key* K

Tulip Onion Encryption



Tulip Onion Encryption



$O_1 =$ $H_{1,1}$ $H_{1,2}$ $H_{1,3}$ $H_{1,4}$ $H_{1,5}$ C_1 $\langle K \rangle$ $\langle K \rangle$ $\langle 0 \rangle$

$\mathsf{Dec}_{\mathsf{sk}(M_1)}(\cdot)$ $\}\cdot\{_{k_1}$ $\}\cdot\{_{k_1}$ $\}\cdot\{_{k_1}$ $\}\cdot\{_{k_1}$ $\}\cdot\{_{k_1}$ $\}\cdot\{_{k_1}$ $\}\cdot\{_{k_1}$ no bruise

$(1, k_1, M_2)$

$O_2 =$ $H_{2,1}$ $H_{2,2}$ $H_{2,3}$ $H_{2,4}$ C_2 $\langle K \rangle$ $\langle K \rangle$

$\mathsf{Dec}_{\mathsf{sk}(M_2)}(\cdot)$ $\}\cdot\{_{k_2}$ $\}\cdot\{_{k_2}$ $\}\cdot\{_{k_2}$ $\}\cdot\{_{k_2}$

$(2, k_2, G_3)$

$O_3 =$ $H_{3,1}$ $H_{3,2}$ $H_{3,3}$ C_3

$\mathsf{Dec}_{\mathsf{sk}(G_3)}(\cdot)$ $\}\cdot\{_{k_3}$ $\}\cdot\{_{k_3}$ $\}\cdot\{_{k_3}$

$(3, k_3, G_4)$

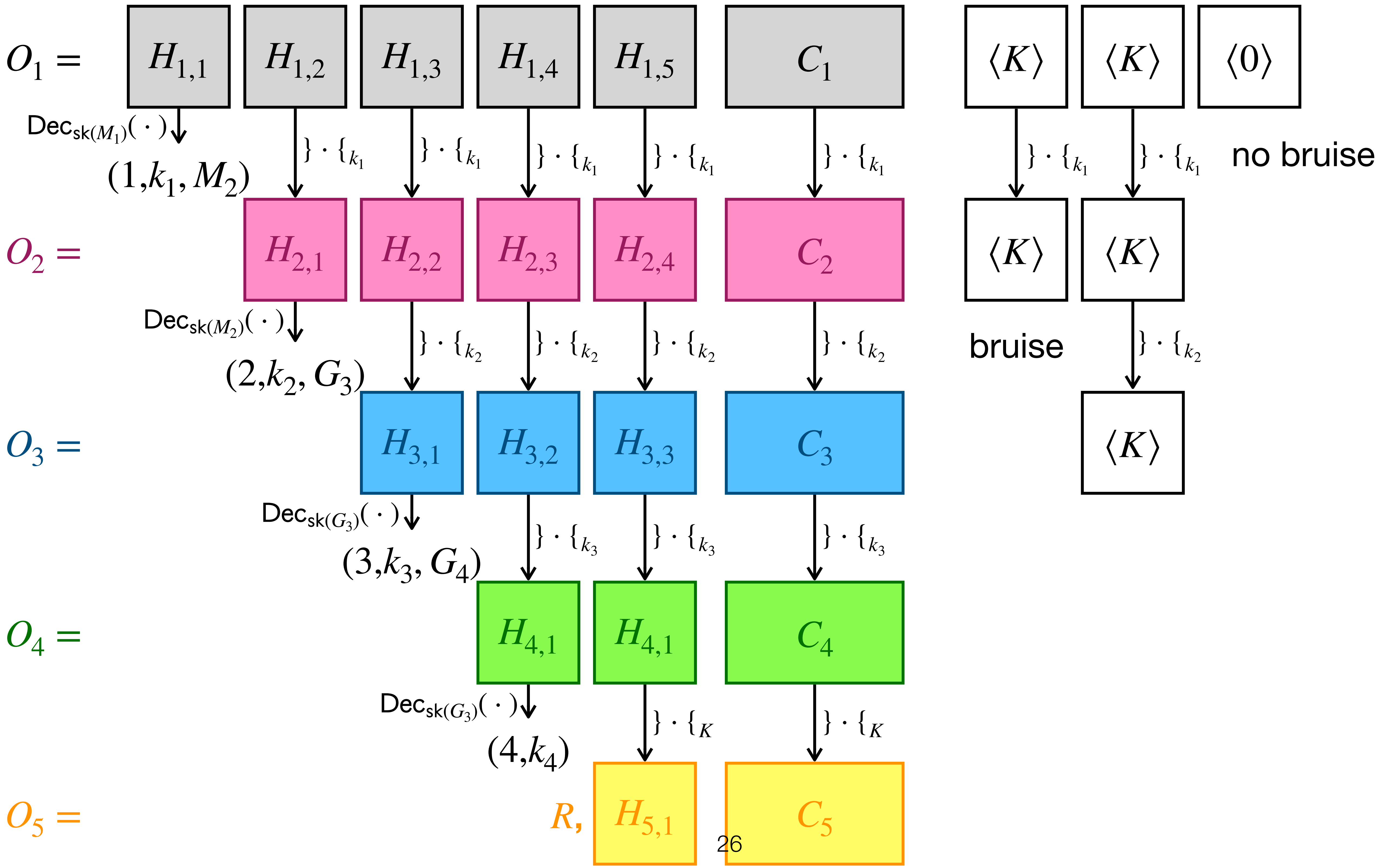
$O_4 =$ $H_{4,1}$ $H_{4,1}$ C_4

$\mathsf{Dec}_{\mathsf{sk}(G_3)}(\cdot)$ $\}\cdot\{_K$ $\}\cdot\{_K$

$(4, k_4)$

$O_5 =$ $R,$ $H_{5,1}$ C_5

Tulip Onion Encryption



$O_1 =$ $H_{1,1}$ $H_{1,2}$ $H_{1,3}$ $H_{1,4}$ $H_{1,5}$ C_1 $\langle K \rangle$ $\langle K \rangle$ $\langle 0 \rangle$

$\mathsf{Dec}_{\mathsf{sk}(M_1)}(\cdot)$ $(1, k_1, M_2)$ $\}\cdot\{_{k_1}$ $\}\cdot\{_{k_1}$ $\}\cdot\{_{k_1}$ $\}\cdot\{_{k_1}$ $\}\cdot\{_{k_1}$ $\}\cdot\{_{k_1}$ $\}\cdot\{_{k_1}$ no bruise

$O_2 =$ $H_{2,1}$ $H_{2,2}$ $H_{2,3}$ $H_{2,4}$ C_2 $\langle K \rangle$ $\langle K \rangle$

$\mathsf{Dec}_{\mathsf{sk}(M_2)}(\cdot)$ $(2, k_2, G_3)$ $\}\cdot\{_{k_2}$ $\}\cdot\{_{k_2}$ $\}\cdot\{_{k_2}$ $\}\cdot\{_{k_2}$ bruise $\}\cdot\{_{k_2}$

$O_3 =$ $H_{3,1}$ $H_{3,2}$ $H_{3,3}$ C_3 $\langle K \rangle$

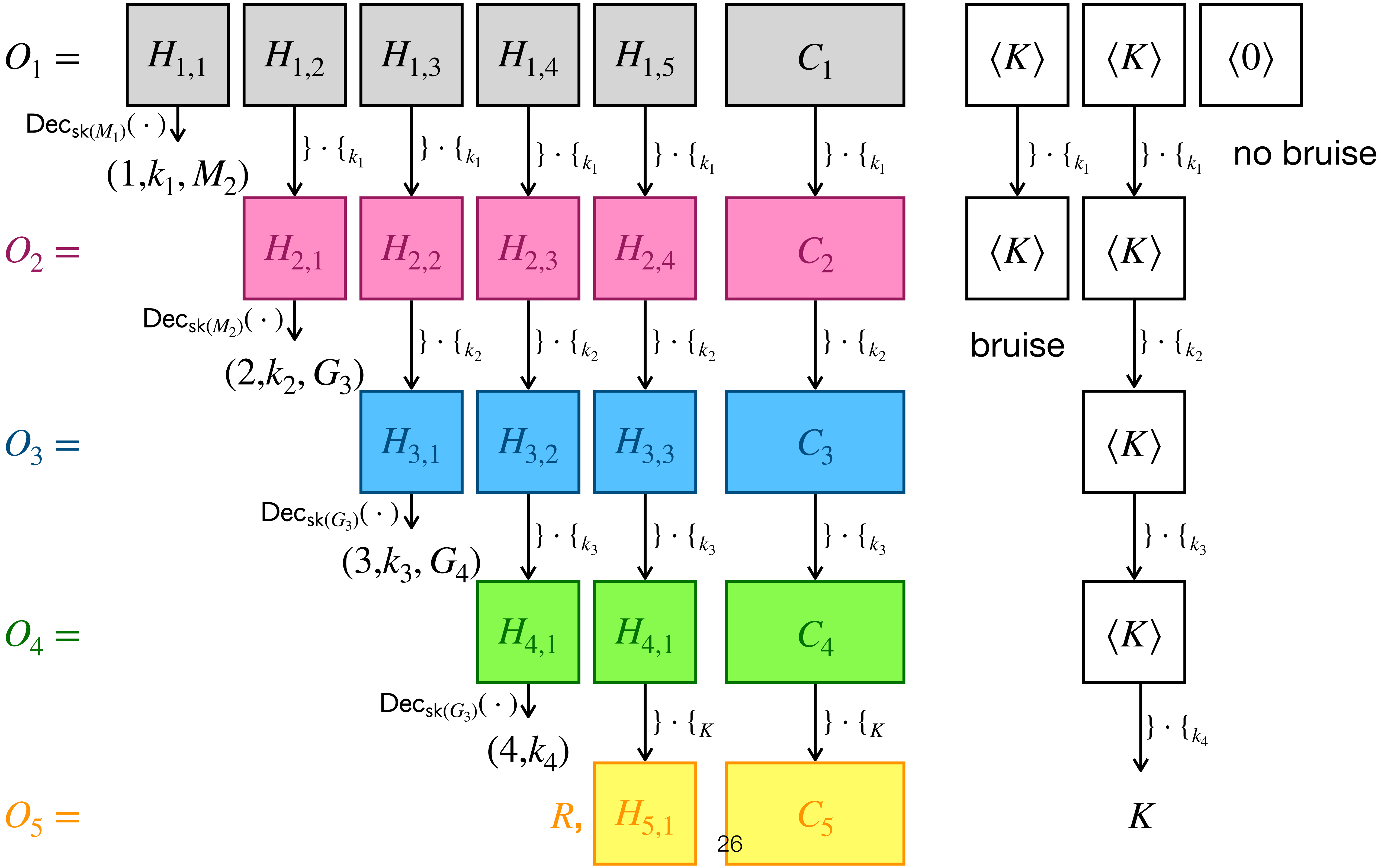
$\mathsf{Dec}_{\mathsf{sk}(G_3)}(\cdot)$ $(3, k_3, G_4)$ $\}\cdot\{_{k_3}$ $\}\cdot\{_{k_3}$ $\}\cdot\{_{k_3}$

$O_4 =$ $H_{4,1}$ $H_{4,1}$ C_4

$\mathsf{Dec}_{\mathsf{sk}(G_3)}(\cdot)$ $(4, k_4)$ $\}\cdot\{_{K}$ $\}\cdot\{_{K}$

$O_5 =$ $R,$ $H_{5,1}$ C_5

Tulip Onion Encryption



$O_1 =$ $H_{1,1}$ $H_{1,2}$ $H_{1,3}$ $H_{1,4}$ $H_{1,5}$ C_1 $\langle K \rangle$ $\langle K \rangle$ $\langle 0 \rangle$

$\mathsf{Dec}_{\mathsf{sk}(M_1)}(\cdot)$ $(1, k_1, M_2)$ $\} \cdot \{_{k_1}$ $\} \cdot \{_{k_1}$ $\} \cdot \{_{k_1}$ $\} \cdot \{_{k_1}$ $\} \cdot \{_{k_1}$ $\} \cdot \{_{k_1}$ $\} \cdot \{_{k_1}$ no bruise

$O_2 =$ $H_{2,1}$ $H_{2,2}$ $H_{2,3}$ $H_{2,4}$ C_2 $\langle K \rangle$ $\langle K \rangle$

$\mathsf{Dec}_{\mathsf{sk}(M_2)}(\cdot)$ $(2, k_2, G_3)$ $\} \cdot \{_{k_2}$ $\} \cdot \{_{k_2}$ $\} \cdot \{_{k_2}$ $\} \cdot \{_{k_2}$ bruise $\} \cdot \{_{k_2}$

$O_3 =$ $H_{3,1}$ $H_{3,2}$ $H_{3,3}$ C_3 $\langle K \rangle$

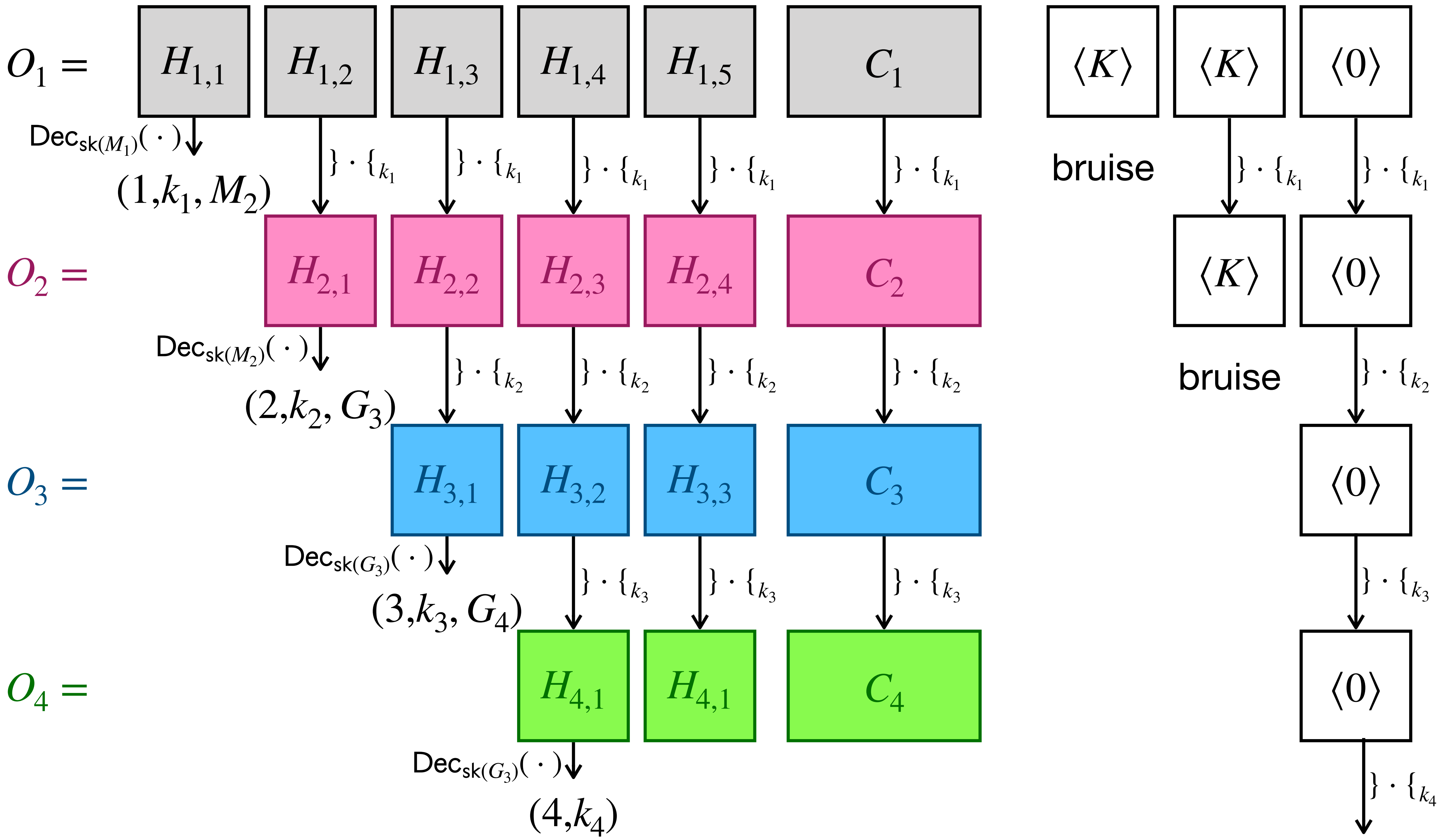
$\mathsf{Dec}_{\mathsf{sk}(G_3)}(\cdot)$ $(3, k_3, G_4)$ $\} \cdot \{_{k_3}$ $\} \cdot \{_{k_3}$ $\} \cdot \{_{k_3}$ $\} \cdot \{_{k_3}$

$O_4 =$ $H_{4,1}$ $H_{4,1}$ C_4 $\langle K \rangle$

$\mathsf{Dec}_{\mathsf{sk}(G_3)}(\cdot)$ $(4, k_4)$ $\} \cdot \{_{K}$ $\} \cdot \{_{K}$ $\} \cdot \{_{k_4}$

$O_5 =$ $R,$ $H_{5,1}$ C_5 K

Tulip Onion Encryption



$O_1 =$ $H_{1,1}$ $H_{1,2}$ $H_{1,3}$ $H_{1,4}$ $H_{1,5}$ C_1 $\langle K \rangle$ $\langle K \rangle$ $\langle 0 \rangle$

$\mathsf{Dec}_{\mathsf{sk}(M_1)}(\cdot)$ $(1, k_1, M_2)$ $\}\cdot\{_{k_1}$ $\}\cdot\{_{k_1}$ $\}\cdot\{_{k_1}$ $\}\cdot\{_{k_1}$ $\}\cdot\{_{k_1}$ bruise $\}\cdot\{_{k_1}$ $\}\cdot\{_{k_1}$

$O_2 =$ $H_{2,1}$ $H_{2,2}$ $H_{2,3}$ $H_{2,4}$ C_2 $\langle K \rangle$ $\langle 0 \rangle$

$\mathsf{Dec}_{\mathsf{sk}(M_2)}(\cdot)$ $(2, k_2, G_3)$ $\}\cdot\{_{k_2}$ $\}\cdot\{_{k_2}$ $\}\cdot\{_{k_2}$ $\}\cdot\{_{k_2}$ bruise $\}\cdot\{_{k_2}$

$O_3 =$ $H_{3,1}$ $H_{3,2}$ $H_{3,3}$ C_3 $\langle 0 \rangle$

$\mathsf{Dec}_{\mathsf{sk}(G_3)}(\cdot)$ $(3, k_3, G_4)$ $\}\cdot\{_{k_3}$ $\}\cdot\{_{k_3}$ $\}\cdot\{_{k_3}$ $\}\cdot\{_{k_3}$

$O_4 =$ $H_{4,1}$ $H_{4,1}$ C_4 $\langle 0 \rangle$

$\mathsf{Dec}_{\mathsf{sk}(G_3)}(\cdot)$ $(4, k_4)$ $\}\cdot\{_{k_4}$

Too bruised; can't recover R, O_5

Contribution 3.

Our construction: Π_t

Onion Routing Protocol Π_t

Onion forming phase. Every P_i forms many *bruisable onions:*

- An onion to recipient
- A random number of checkpoint onions to random locations:
    - Expected number is polylog in security parameter

Onion routing (execution) phase. Every P_i does:

```
1. Initialize time to $t = 1$.
2. Peels onions as they arrive:
    a. If onion is “on time” or “early”, place in outbox.
       Update onion counts, accordingly.
    b. If onion is “late,” bruise and send immediately.
3. If onion count for time $t \geq \text{threshold}$:
    a. Send onions for time $t$ in random order.
    b. Update $t = t + 1$.
```

Thank you!

Backup Slides

Standard Onion Security [Ando-Lysyanskaya]

Standard Onion Security [Ando-Lysyanskaya]

Intuition. Information behind an honest party remains hidden.

Standard Onion Security [Ando-Lysyanskaya]

Intuition. Information behind an honest party remains hidden.

Informal Definition. Adversary cannot distinguish between:

1. $O_1^{b=0}$ formed on all onion parameters:

    - Message m challenge
    - Complete path (Q_1, Q_2, Q_3, Q_4, R) and associated keys and metadata

Standard Onion Security [Ando-Lysyanskaya]

Intuition. Information behind an honest party remains hidden.

Informal Definition. Adversary cannot distinguish between:

1. $O_1^{b=0}$ formed on all onion parameters:
   - Message m challenge
   - Complete path (Q_1, Q_2, Q_3, Q_4, R) and associated keys and metadata
2. $O_1^{b=1}$ formed without any information after honest intermediary, e.g., Q_3:
   - Dummy message $\perp$
   - Partial path, e.g, (Q_1, Q_2, Q_3), and associated keys and metadata

Standard Onion Security [Ando-Lysyanskaya]

Intuition. Information behind an honest party remains hidden.

Informal Definition. Adversary cannot distinguish between:

challenge bit

1. $O_1^{b=0}$ formed on all onion parameters:
   - Message m challenge
   - Complete path (Q_1, Q_2, Q_3, Q_4, R) and associated keys and metadata
2. $O_1^{b=1}$ formed without any information after honest intermediary, e.g., Q_3:
   - Dummy message $\perp$
   - Partial path, e.g, (Q_1, Q_2, Q_3), and associated keys and metadata

Standard Onion Security [Ando-Lysyanskaya]

Intuition. Information behind an honest party remains hidden.

Informal Definition. Adversary cannot distinguish between:

challenge bit

1. $O_1^{b=0}$ formed on all onion parameters:

   onion layer

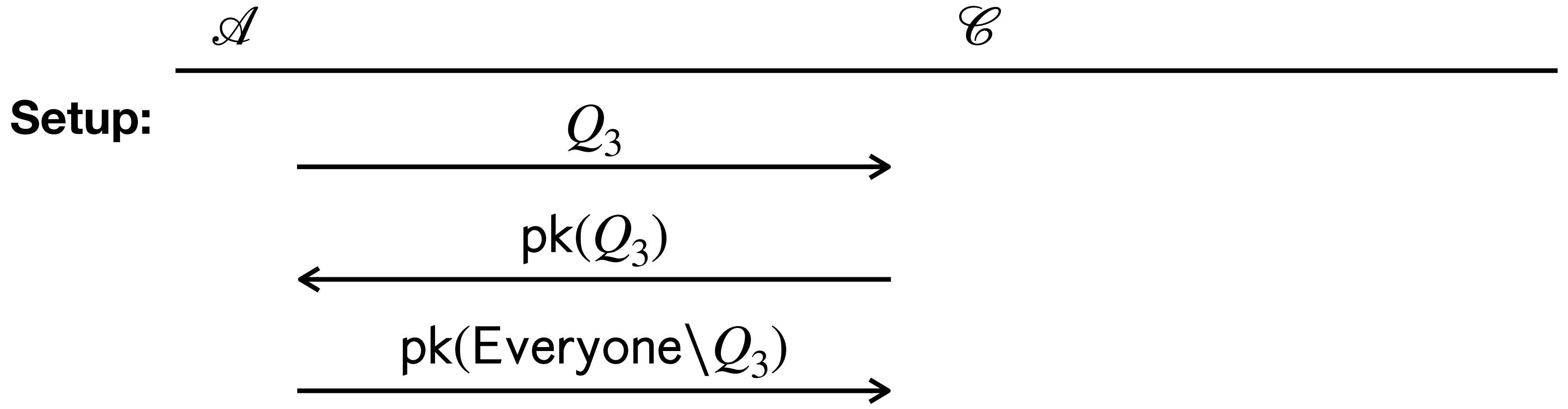
   - Message m challenge
   - Complete path (Q_1, Q_2, Q_3, Q_4, R) and associated keys and metadata

2. $O_1^{b=1}$ formed without any information after honest intermediary, e.g., Q_3:

   - Dummy message $\perp$
   - Partial path, e.g, (Q_1, Q_2, Q_3), and associated keys and metadata

Standard Onion Encryption: Security

Formal Definition. CCA2-like, defined w.r.t. Onion Security Game:



$\mathscr{A}$ $\mathscr{C}$

Setup:

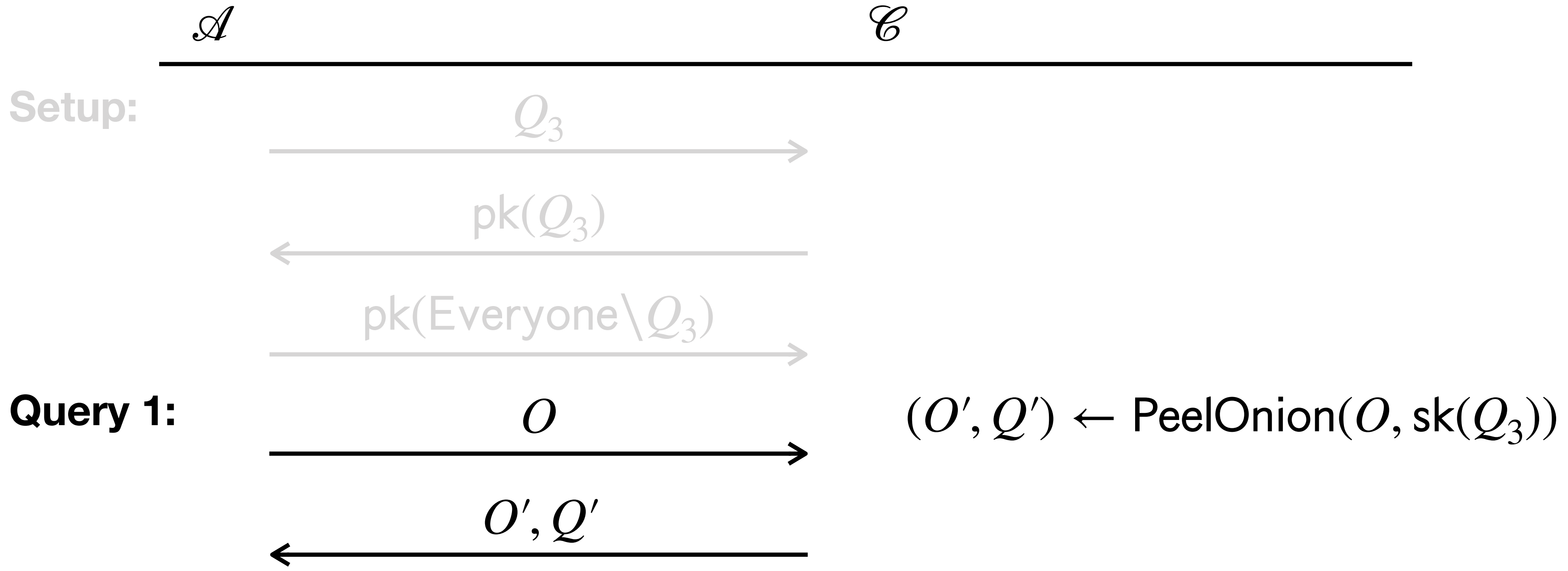
Q_3

pk(Q_3)

pk(Everyone$\backslash Q_3$)

Standard Onion Encryption: Security

Formal Definition. CCA2-like, defined w.r.t. Onion Security Game:



$\mathscr{A}$ $\mathscr{C}$

Setup:

$\xrightarrow{Q_3}$

$\xleftarrow{\mathsf{pk}(Q_3)}$

$\xrightarrow{\mathsf{pk}(\mathsf{Everyone} \backslash Q_3)}$

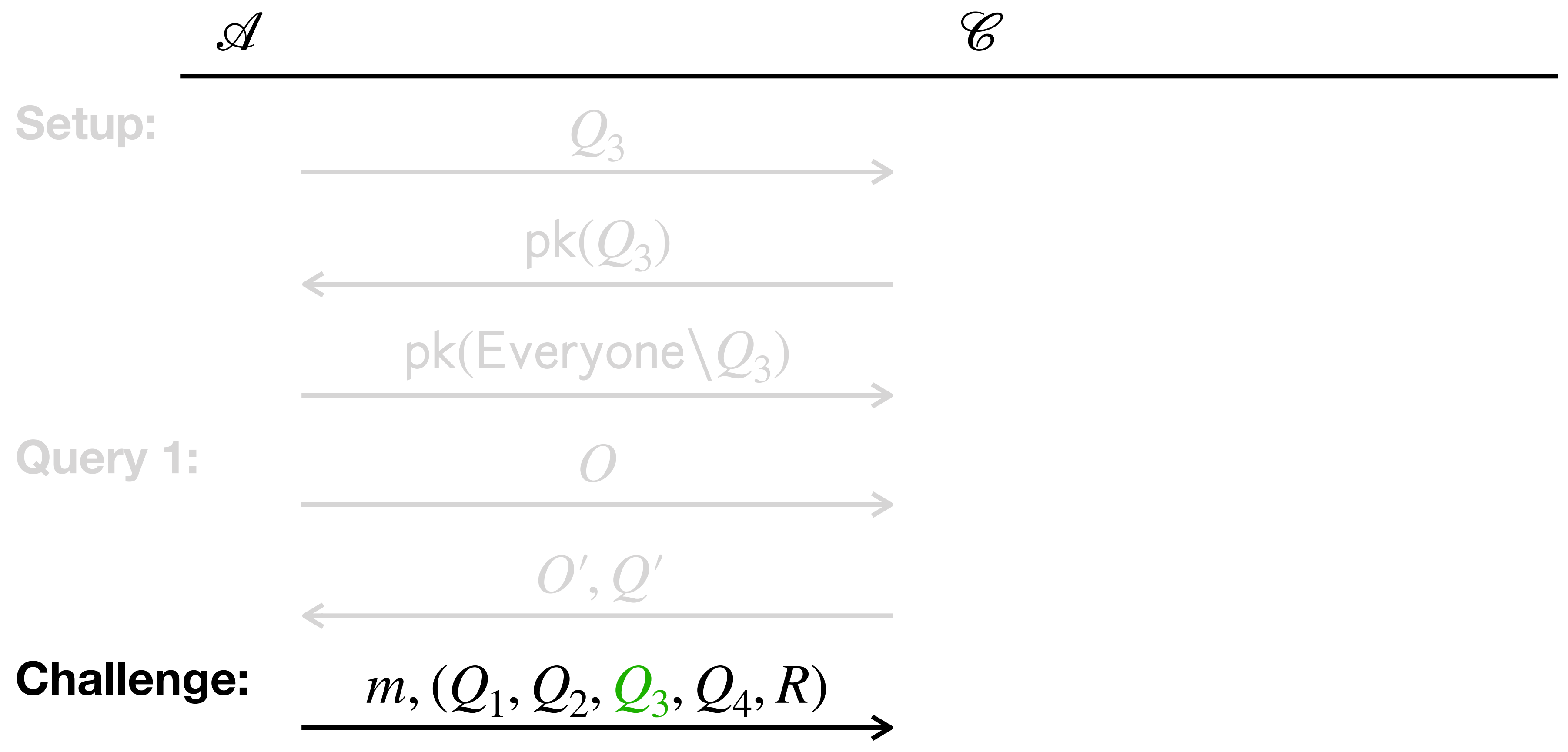
Query 1:

$\xrightarrow{O}$ $(O', Q') \leftarrow \mathsf{PeelOnion}(O, \mathsf{sk}(Q_3))$

$\xleftarrow{O', Q'}$

Standard Onion Encryption: Security

Formal Definition. CCA2-like, defined w.r.t. Onion Security Game:



$\mathscr{A}$ $\mathscr{C}$

Setup: $\xrightarrow{Q_3}$

$\xleftarrow{\mathsf{pk}(Q_3)}$

$\xrightarrow{\mathsf{pk}(\text{Everyone} \setminus Q_3)}$

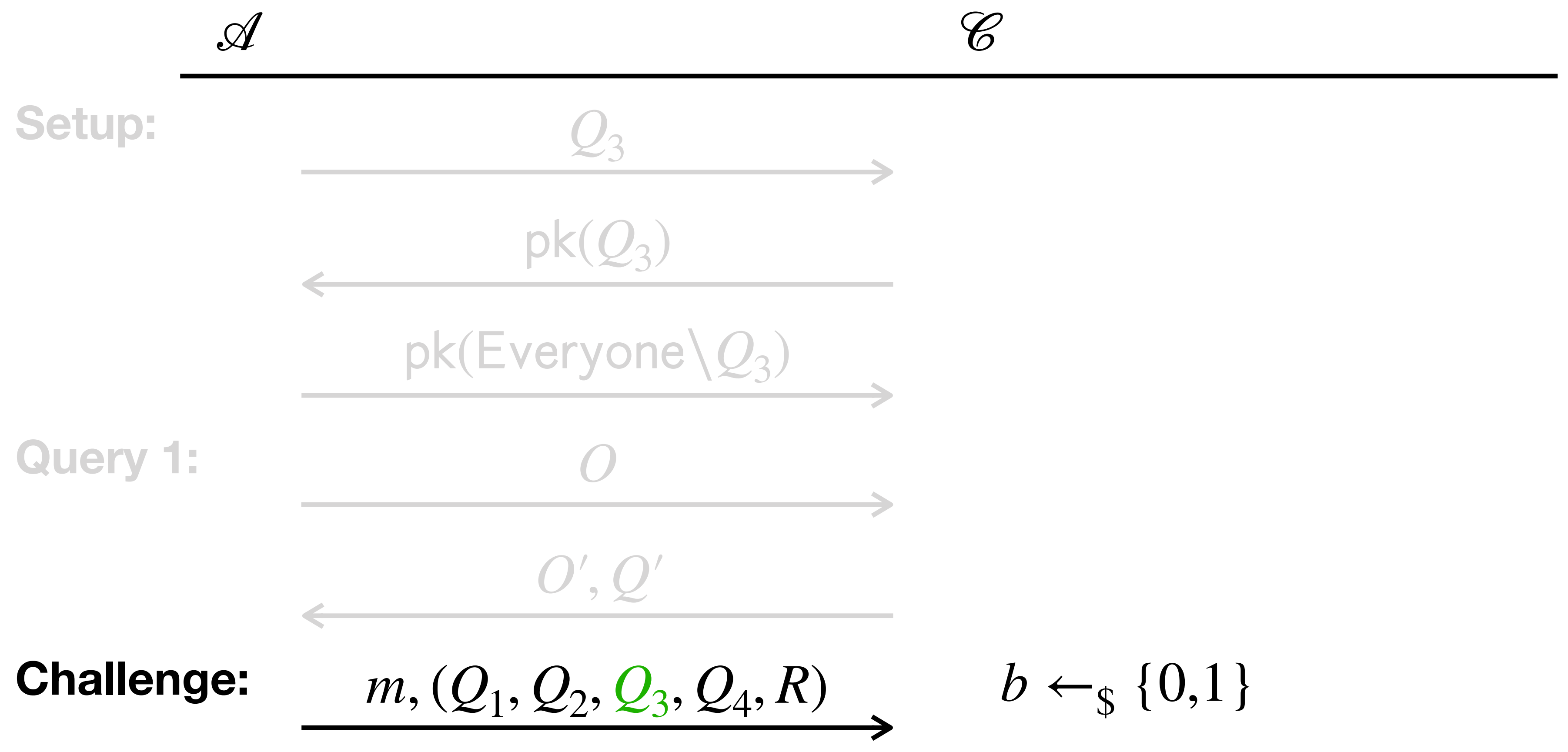
Query 1: $\xrightarrow{O}$

$\xleftarrow{O', Q'}$

Challenge: $\xrightarrow{m, (Q_1, Q_2, Q_3, Q_4, R)}$

Standard Onion Encryption: Security

Formal Definition. CCA2-like, defined w.r.t. Onion Security Game:



$\mathscr{A}$ $\mathscr{C}$

Setup: $\xrightarrow{Q_3}$

$\xleftarrow{\text{pk}(Q_3)}$

$\xrightarrow{\text{pk}(\text{Everyone} \backslash Q_3)}$

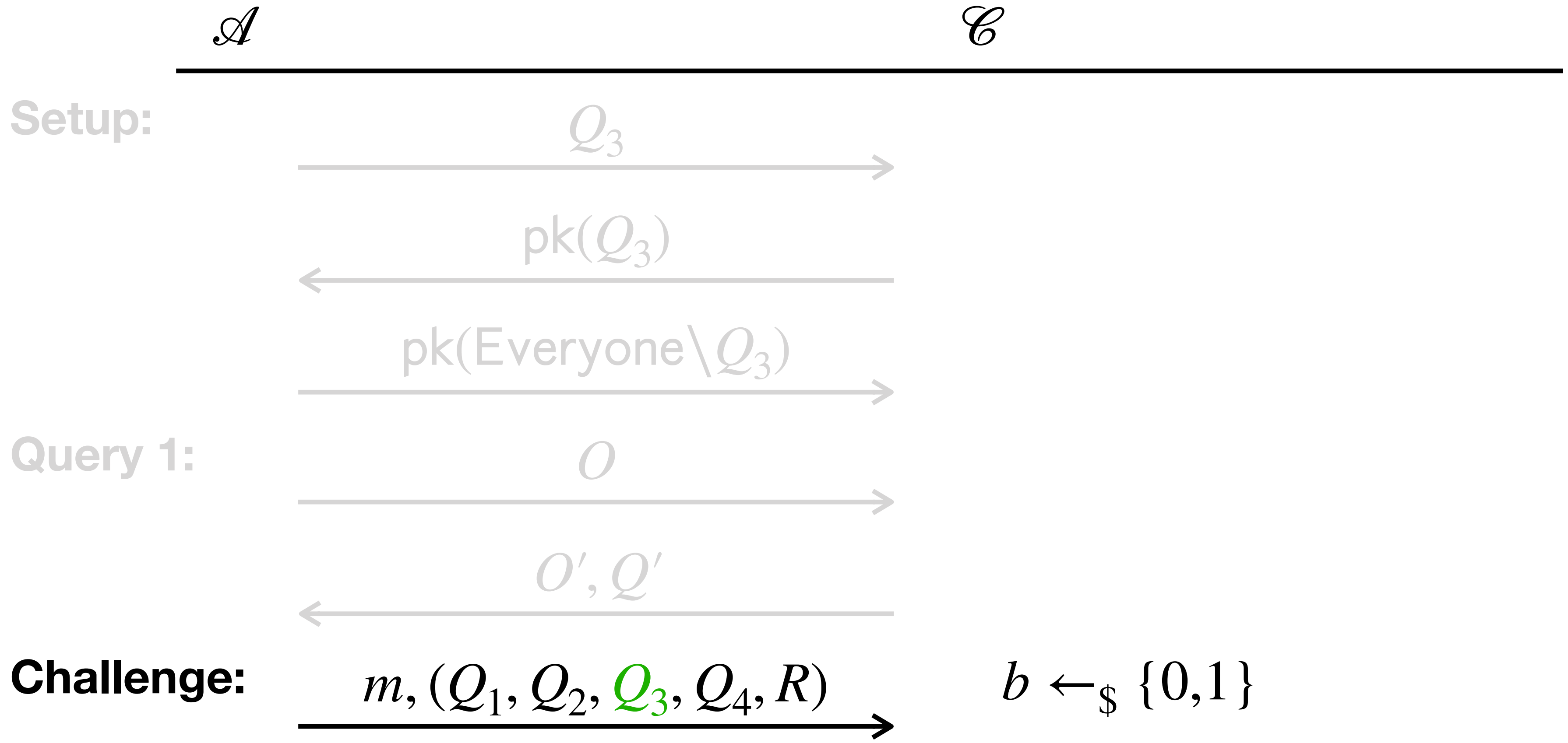
Query 1: $\xrightarrow{O}$

$\xleftarrow{O', Q'}$

Challenge: $\xrightarrow{m, (Q_1, Q_2, Q_3, Q_4, R)}$ $b \leftarrow_{\$} \{0,1\}$

Standard Onion Encryption: Security

Formal Definition. CCA2-like, defined w.r.t. Onion Security Game:



$\mathscr{A}$ $\mathscr{C}$

Setup:

$Q_3 \rightarrow$

$\leftarrow$ pk(Q_3)

pk(Everyone\\Q_3) $\rightarrow$

Query 1:

$O \rightarrow$

$\leftarrow O', Q'$

Challenge:

$m, (Q_1, Q_2, Q_3, Q_4, R) \rightarrow$

$b \leftarrow_{\$} \{0,1\}$

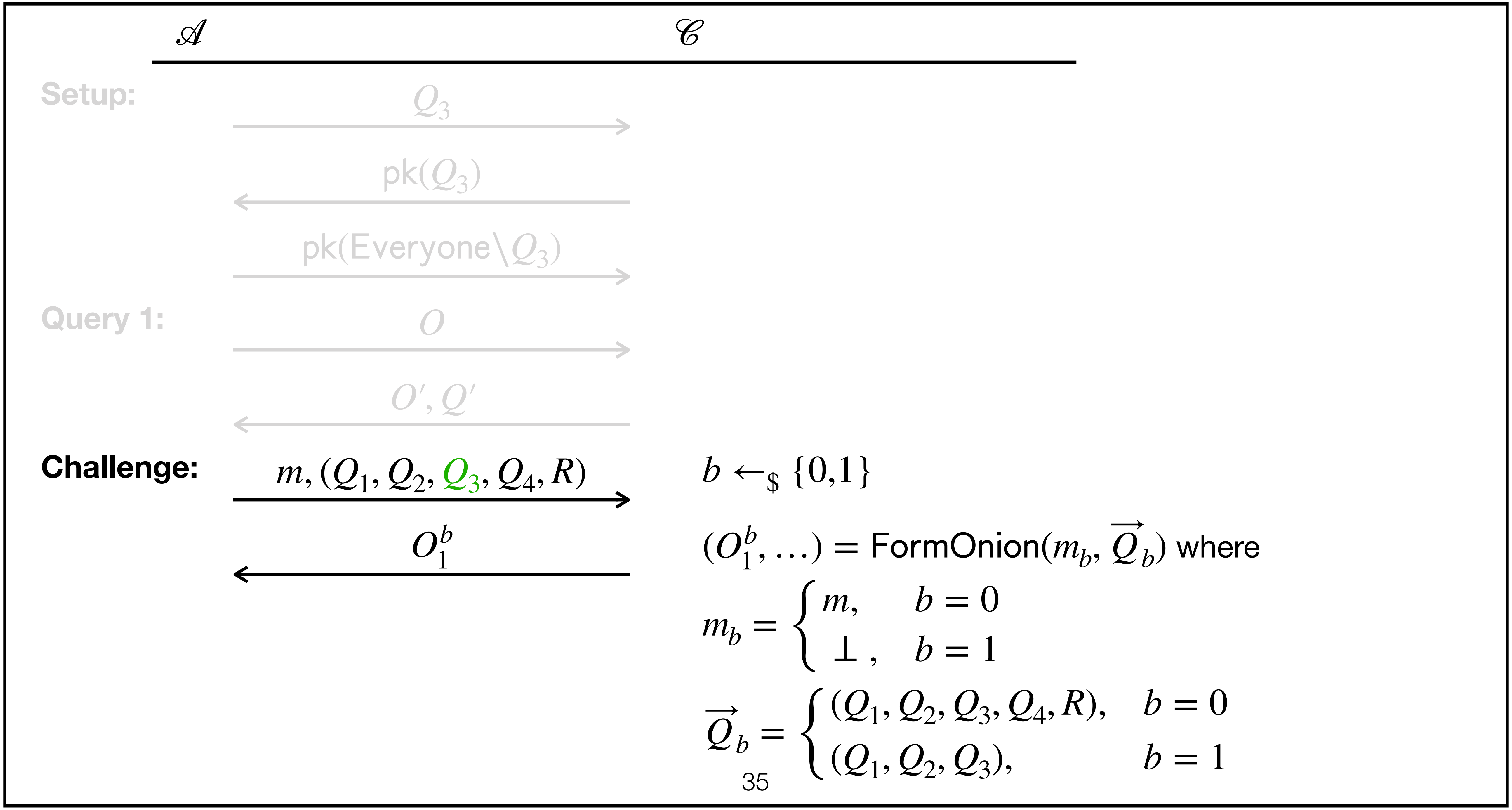
$(O_1^b, \ldots)$ = FormOnion($m_b, \overrightarrow{Q}_b$) where

$$m_b = \begin{cases} m, & b = 0 \\ \perp, & b = 1 \end{cases}$$

$$\overrightarrow{Q}_b = \begin{cases} (Q_1, Q_2, Q_3, Q_4, R), & b = 0 \\ (Q_1, Q_2, Q_3), & b = 1 \end{cases}$$

Standard Onion Encryption: Security

Formal Definition. CCA2-like, defined w.r.t. Onion Security Game:



$\mathscr{A}$ $\mathscr{C}$

Setup: Q_3 → ; ← $\text{pk}(Q_3)$; $\text{pk}(\text{Everyone}\backslash Q_3)$ →

Query 1: O → ; ← O', Q'

Challenge: $m, (Q_1, Q_2, Q_3, Q_4, R)$ → ; ← O_1^b

$b \leftarrow_{\$} \{0,1\}$

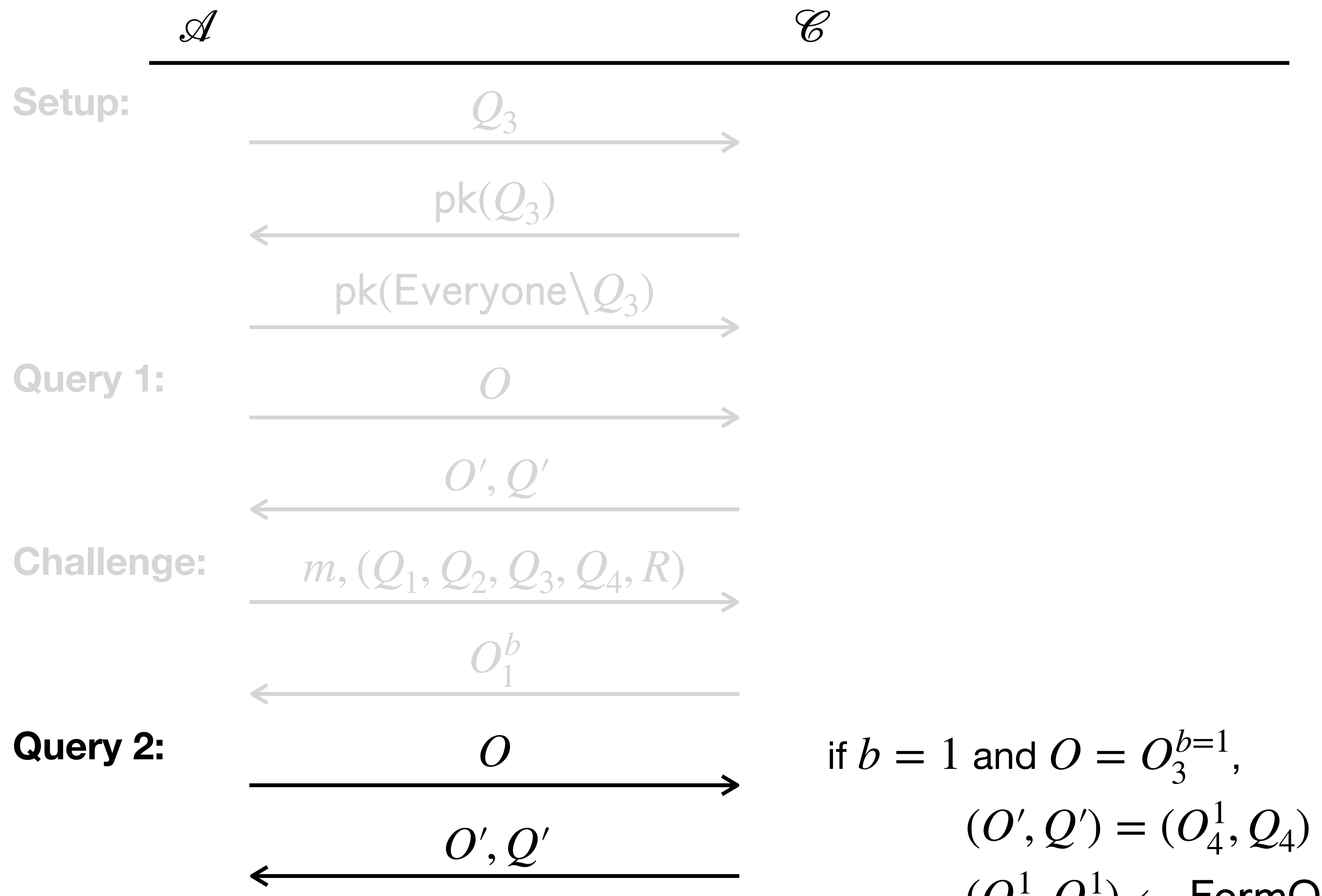
$(O_1^b, \ldots) = \text{FormOnion}(m_b, \overrightarrow{Q}_b)$ where

$$m_b = \begin{cases} m, & b = 0 \\ \perp, & b = 1 \end{cases}$$

$$\overrightarrow{Q}_b = \begin{cases} (Q_1, Q_2, Q_3, Q_4, R), & b = 0 \\ (Q_1, Q_2, Q_3), & b = 1 \end{cases}$$

Standard Onion Encryption: Security

Formal Definition. CCA2-like, defined w.r.t. Onion Security Game:



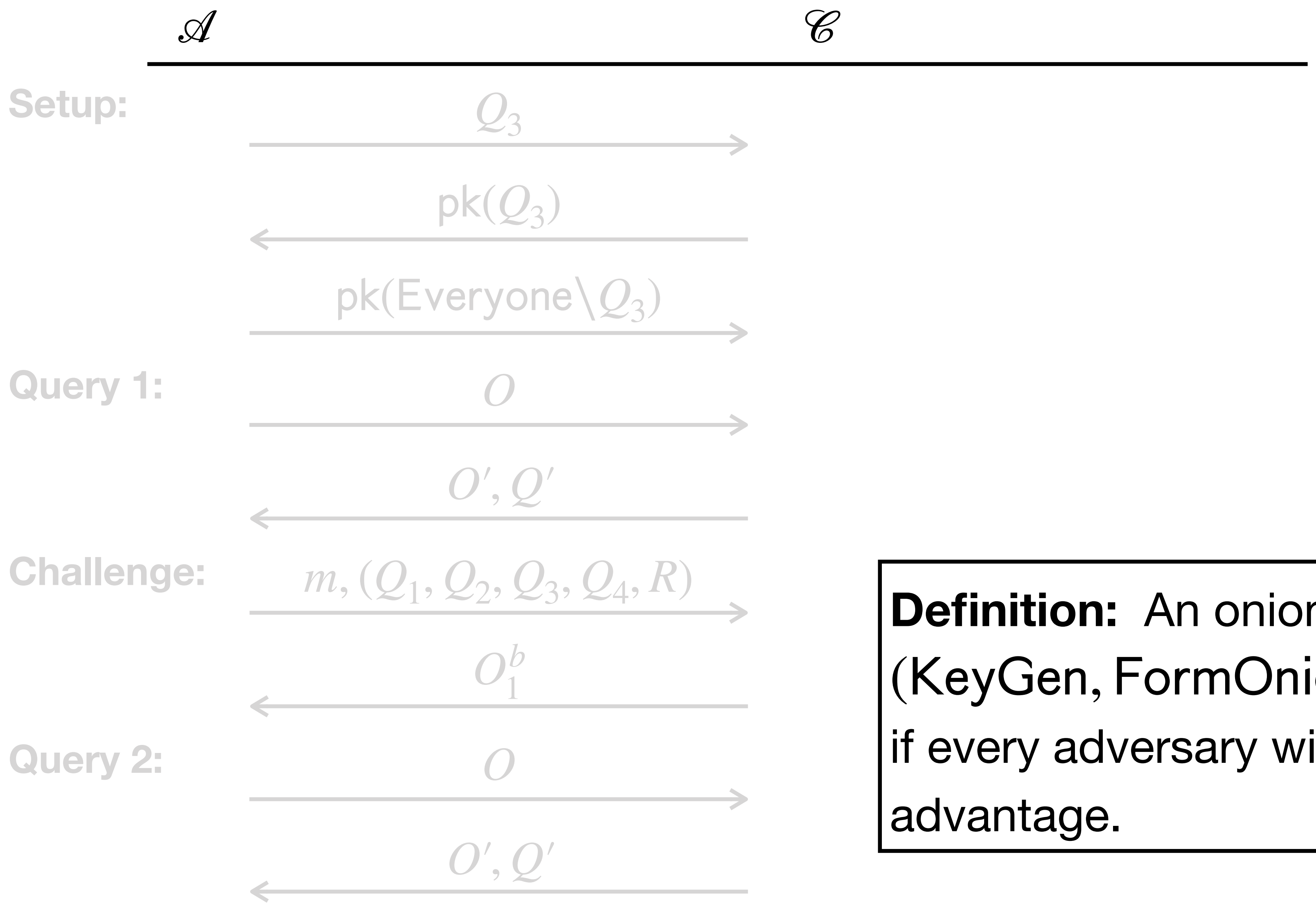
if $b = 1$ and $O = O_3^{b=1}$,

$(O', Q') = (O_4^1, Q_4)$ where

$(O_4^1, O_5^1) \leftarrow$ FormOnion$(m, (Q_4, R))$

Standard Onion Encryption: Security

Formal Definition. CCA2-like, defined w.r.t. Onion Security Game:



$\mathscr{A}$ $\mathscr{C}$

Setup:

Q_3

pk(Q_3)

pk(Everyone$\backslash Q_3$)

Query 1:

O

O', Q'

Challenge:

$m, (Q_1, Q_2, Q_3, Q_4, R)$

O_1^b

Query 2:

O

O', Q'

Definition: An onion encryption scheme (KeyGen, FormOnion, PeelOnion) is *secure* if every adversary wins with negligible advantage.

Output guess b' and wins if $b' = b$