# CUCKOO COMMITMENTS: REGISTRATION-BASED ENCRYPTION & KEY-VALUE MAP COMMITMENTS FOR LARGE SPACES

WORK BY DARIO FIORE [1], DIMITRIS KOLONELOS [1, 2] & PAOLA DE PERTHUIS [3,4]

FULL VERSION: HTTPS://EPRINT.IACR.ORG/2023/1389
PUBLISHED AT ASIACRYPT 2023

PRESENTATION BY PAOLA DE PERTHUIS

1: institute imdea software
2: UNIVERSIDAD POLITECNICA MADRID
3: cosmian
4: ENS ÉCOLE NORMALE SUPÉRIEURE

# REGISTRATION-BASED ENCRYPTION

MOTIVATIONS

# REGISTRATION-BASED ENCRYPTION (RBE)

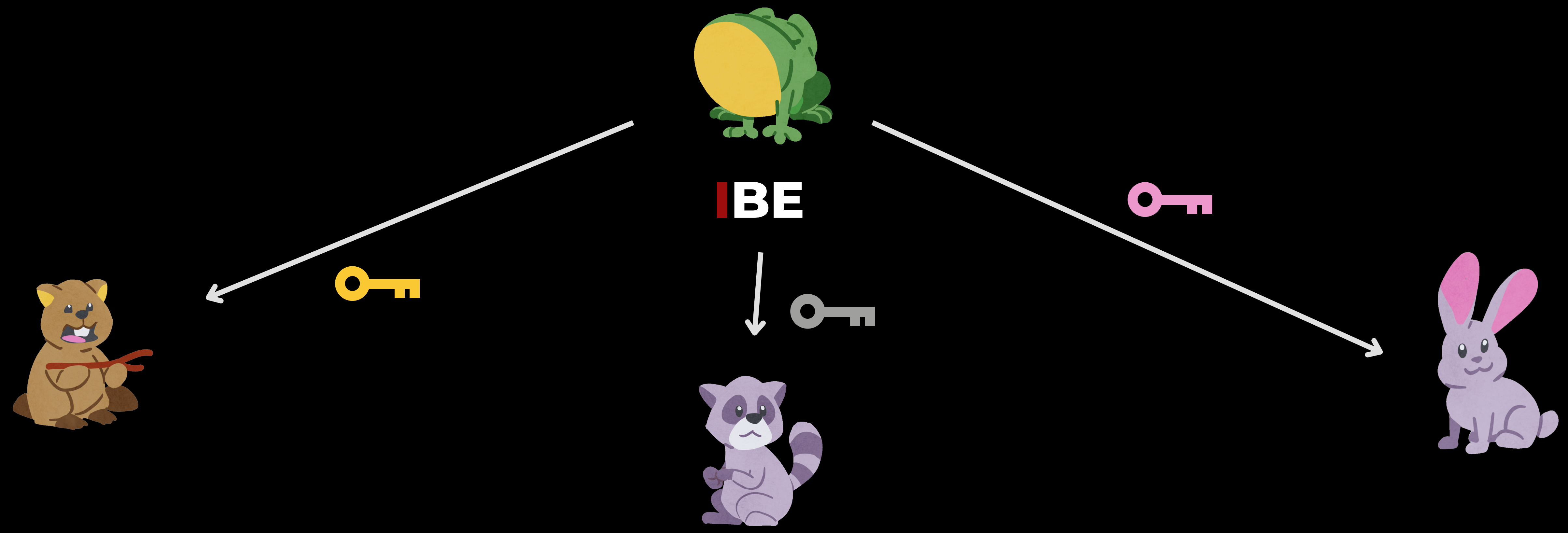
**Stemming from [TCC:Garg-Hajiabadi-Mahmoody-Rahimi-18]**

# REGISTRATION-BASED ENCRYPTION (RBE)

**Stemming from [TCC:Garg-Hajiabadi-Mahmoody-Rahimi-18]**
**Like Identity-Based Encryption (IBE),**
**but without the key curator holding secrets.**

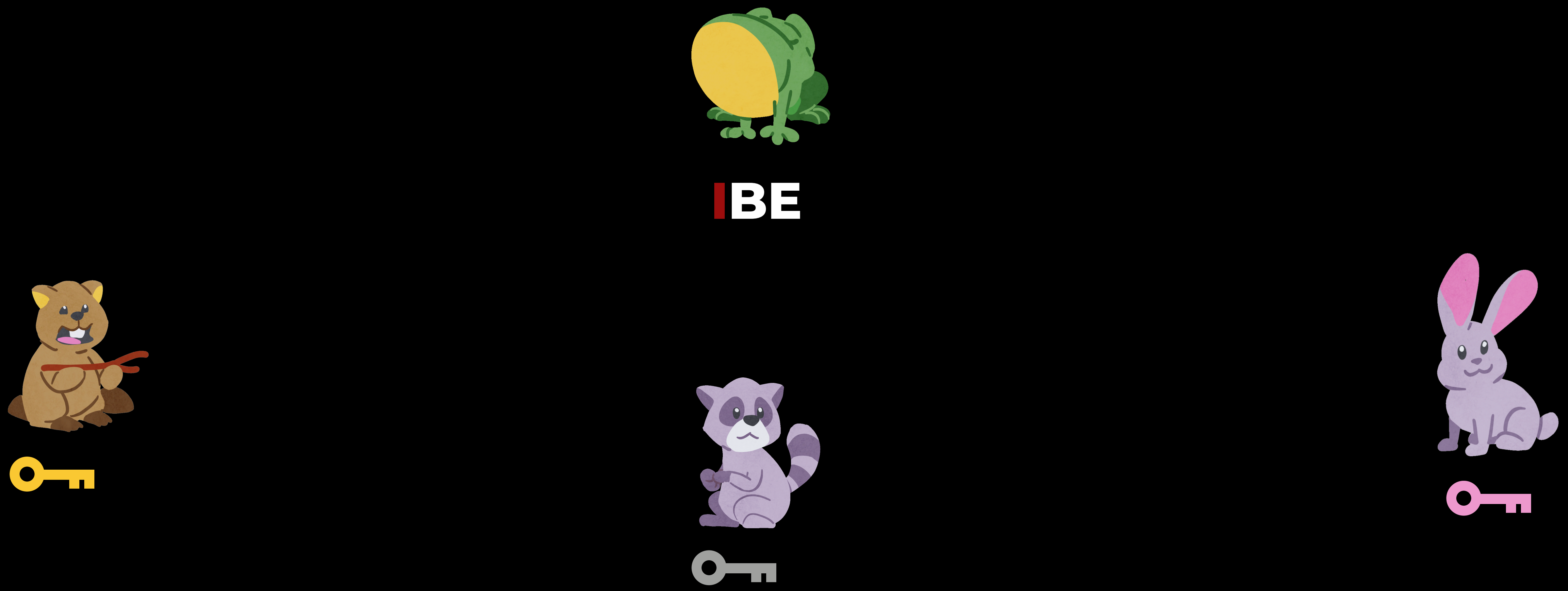
# REGISTRATION-BASED ENCRYPTION (RBE)

**Stemming from [TCC:Garg-Hajiabadi-Mahmoody-Rahimi-18]**
**Like Identity-Based Encryption (IBE),**
**but without the key curator holding secrets.**



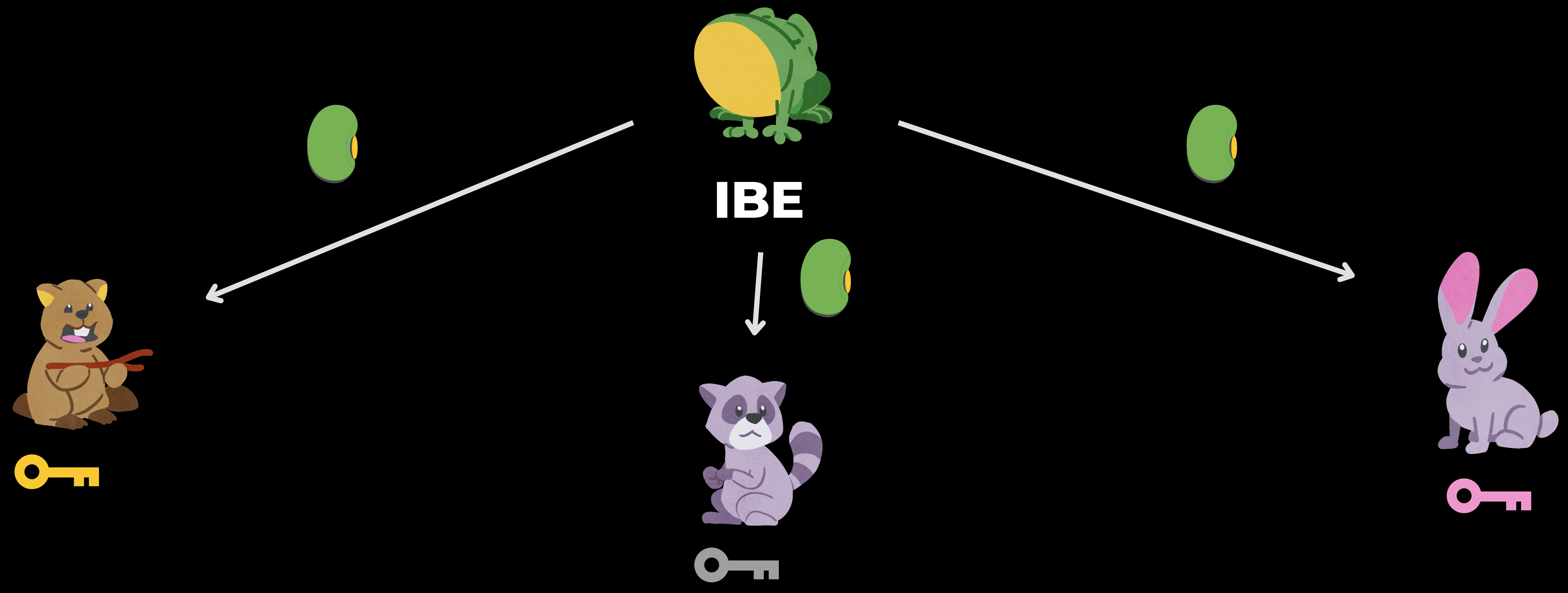
# REGISTRATION-BASED ENCRYPTION (RBE)

**Stemming from [TCC:Garg-Hajiabadi-Mahmoody-Rahimi-18]**
**Like Identity-Based Encryption (IBE),**
**but without the key curator holding secrets.**



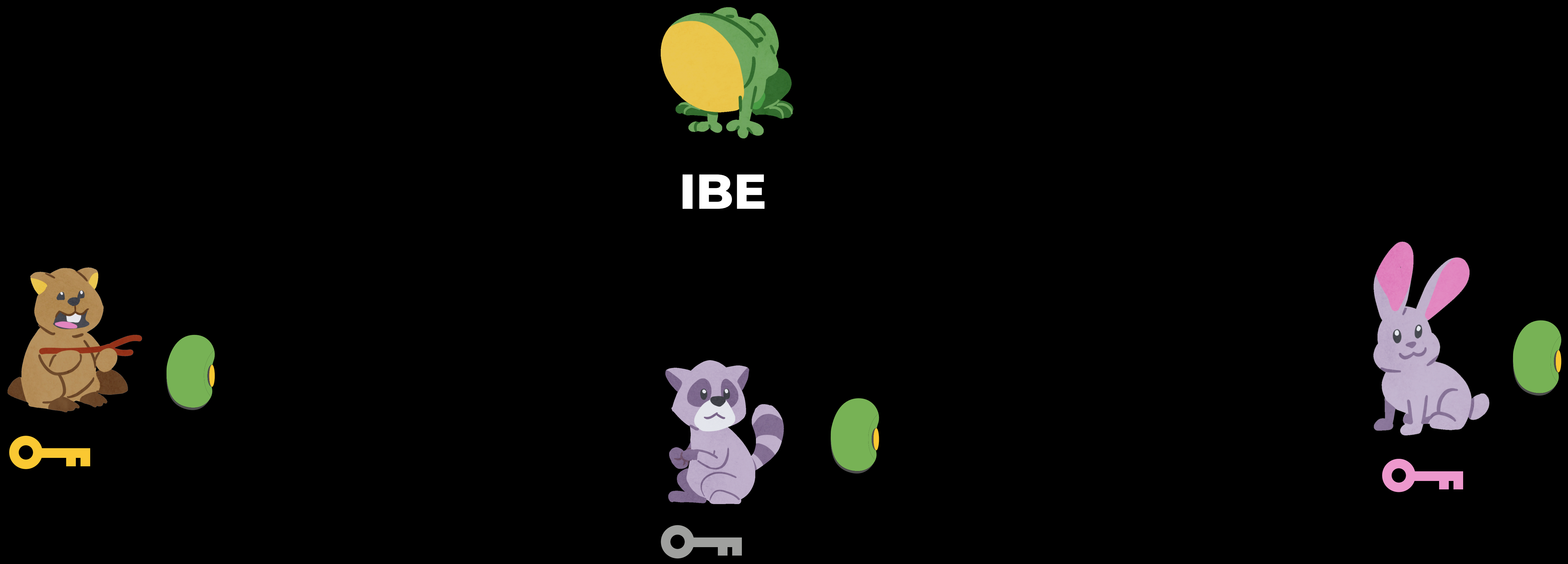
# REGISTRATION-BASED ENCRYPTION (RBE)

**Stemming from [TCC:Garg-Hajiabadi-Mahmoody-Rahimi-18]**
**Like Identity-Based Encryption (IBE),**
**but without the key curator holding secrets.**



# REGISTRATION-BASED ENCRYPTION (RBE)

**Stemming from [TCC:Garg-Hajiabadi-Mahmoody-Rahimi-18]**
**Like Identity-Based Encryption (IBE),**
**but without the key curator holding secrets.**

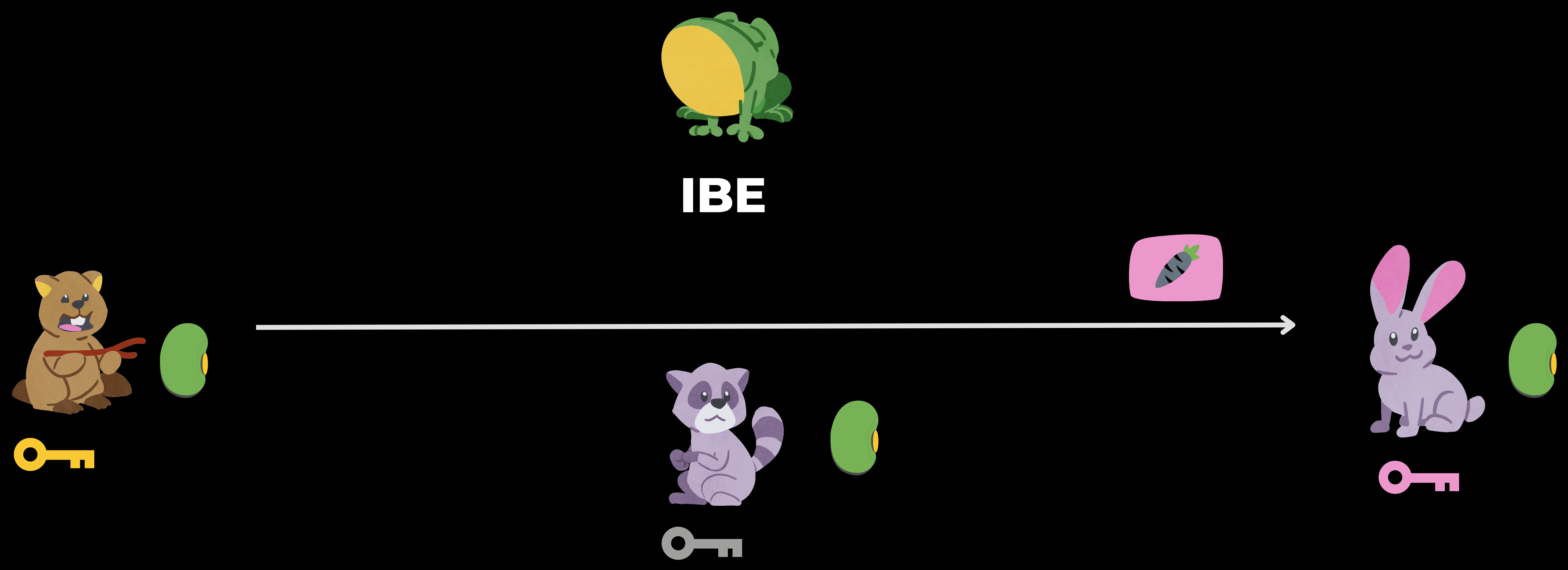


# REGISTRATION-BASED ENCRYPTION (RBE)

**Stemming from [TCC:Garg-Hajiabadi-Mahmoody-Rahimi-18]**
**Like Identity-Based Encryption (IBE),**
**but without the key curator holding secrets.**

# REGISTRATION-BASED ENCRYPTION (RBE)

**Stemming from [TCC:Garg-Hajiabadi-Mahmoody-Rahimi-18]**
**Like Identity-Based Encryption (IBE),**
**but without the key curator holding secrets.**

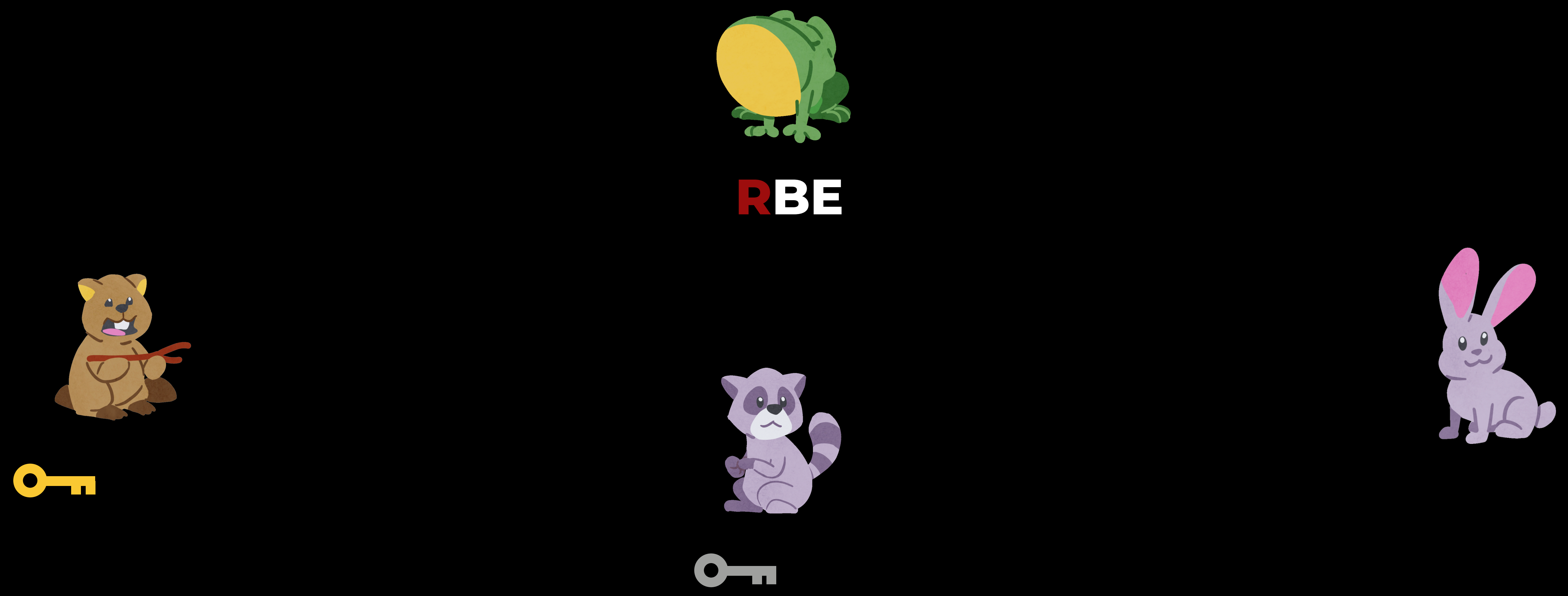


# REGISTRATION-BASED ENCRYPTION (RBE)

**Stemming from [TCC:Garg-Hajiabadi-Mahmoody-Rahimi-18]**
**Like Identity-Based Encryption (IBE),**
**but without the key curator holding secrets.**

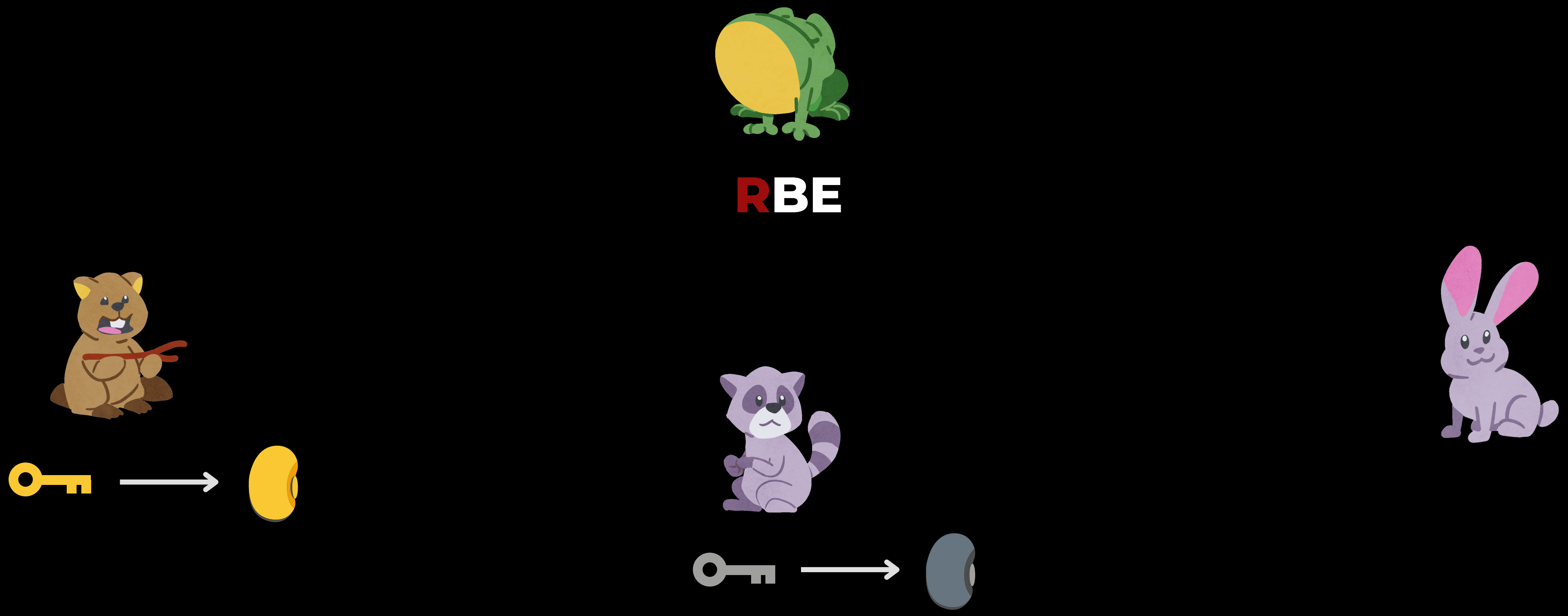
# REGISTRATION-BASED ENCRYPTION (RBE)

**Stemming from [TCC:Garg-Hajiabadi-Mahmoody-Rahimi-18]**
**Like Identity-Based Encryption (IBE),**
**but without the key curator holding secrets.**



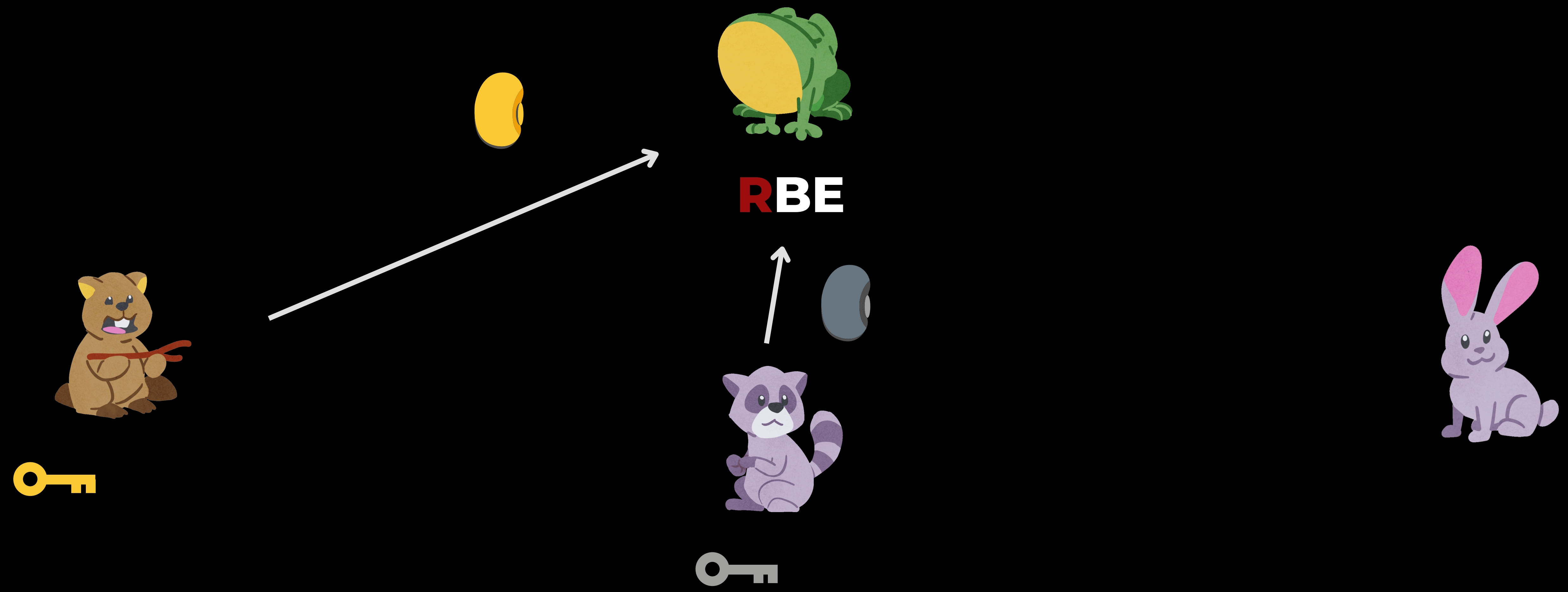
# REGISTRATION-BASED ENCRYPTION (RBE)

**Stemming from [TCC:Garg-Hajiabadi-Mahmoody-Rahimi-18]**
**Like Identity-Based Encryption (IBE),**
**but without the key curator holding secrets.**



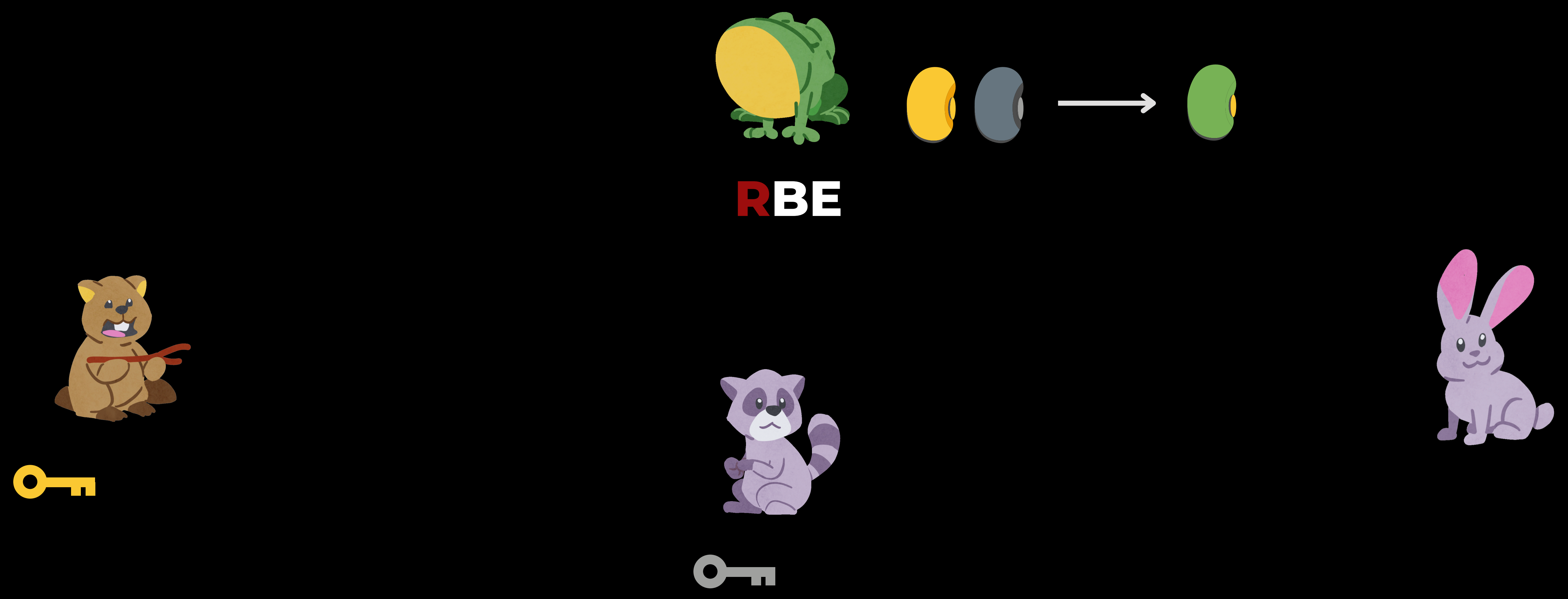
# REGISTRATION-BASED ENCRYPTION (RBE)

**Stemming from [TCC:Garg-Hajiabadi-Mahmoody-Rahimi-18]**
**Like Identity-Based Encryption (IBE),**
**but without the key curator holding secrets.**



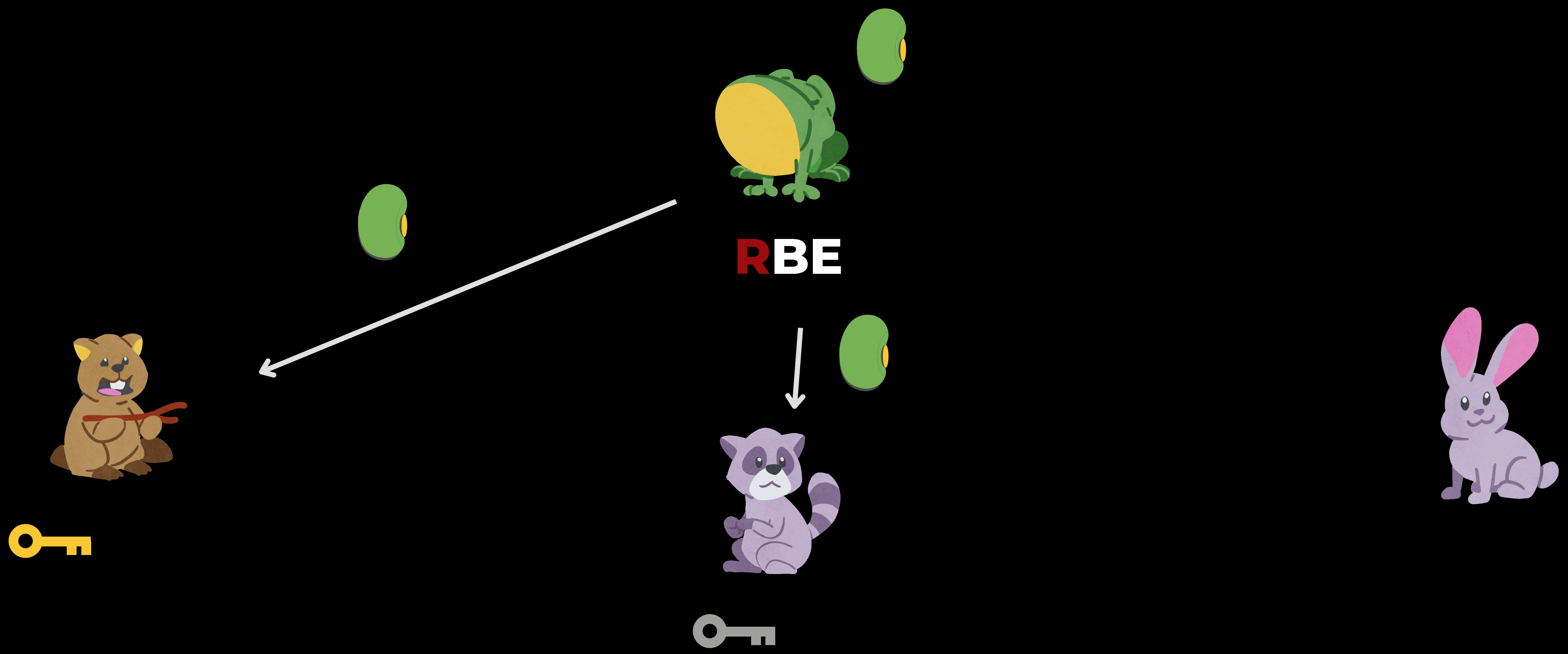
# REGISTRATION-BASED ENCRYPTION (RBE)

**Stemming from [TCC:Garg-Hajiabadi-Mahmoody-Rahimi-18]**
**Like Identity-Based Encryption (IBE),**
**but without the key curator holding secrets.**



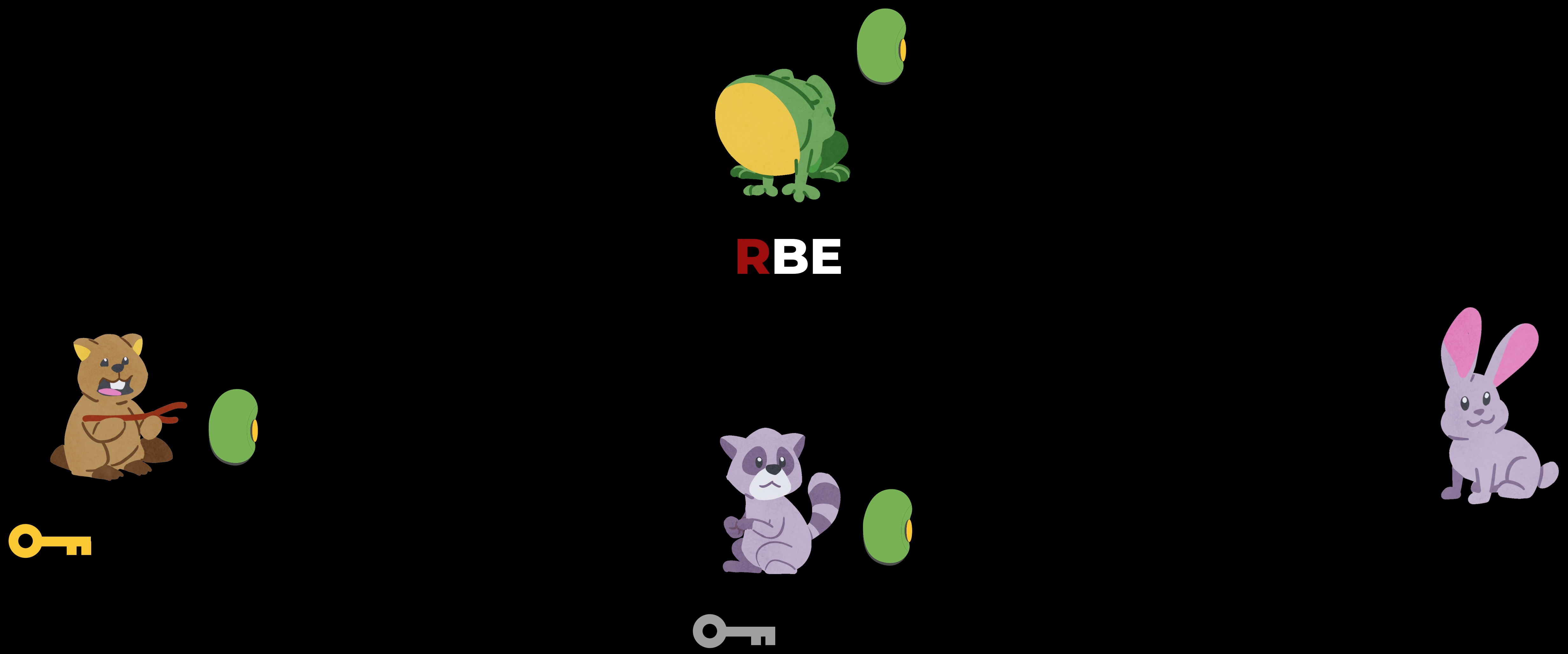
# REGISTRATION-BASED ENCRYPTION (RBE)

**Stemming from [TCC:Garg-Hajiabadi-Mahmoody-Rahimi-18]**
**Like Identity-Based Encryption (IBE),**
**but without the key curator holding secrets.**



# REGISTRATION-BASED ENCRYPTION (RBE)

**Stemming from [TCC:Garg-Hajiabadi-Mahmoody-Rahimi-18]**
**Like Identity-Based Encryption (IBE),**
**but without the key curator holding secrets.**



# REGISTRATION-BASED ENCRYPTION (RBE)

**Stemming from [TCC:Garg-Hajiabadi-Mahmoody-Rahimi-18]**
**Like Identity-Based Encryption (IBE),**
**but without the key curator holding secrets.**

# REGISTRATION-BASED ENCRYPTION (RBE)

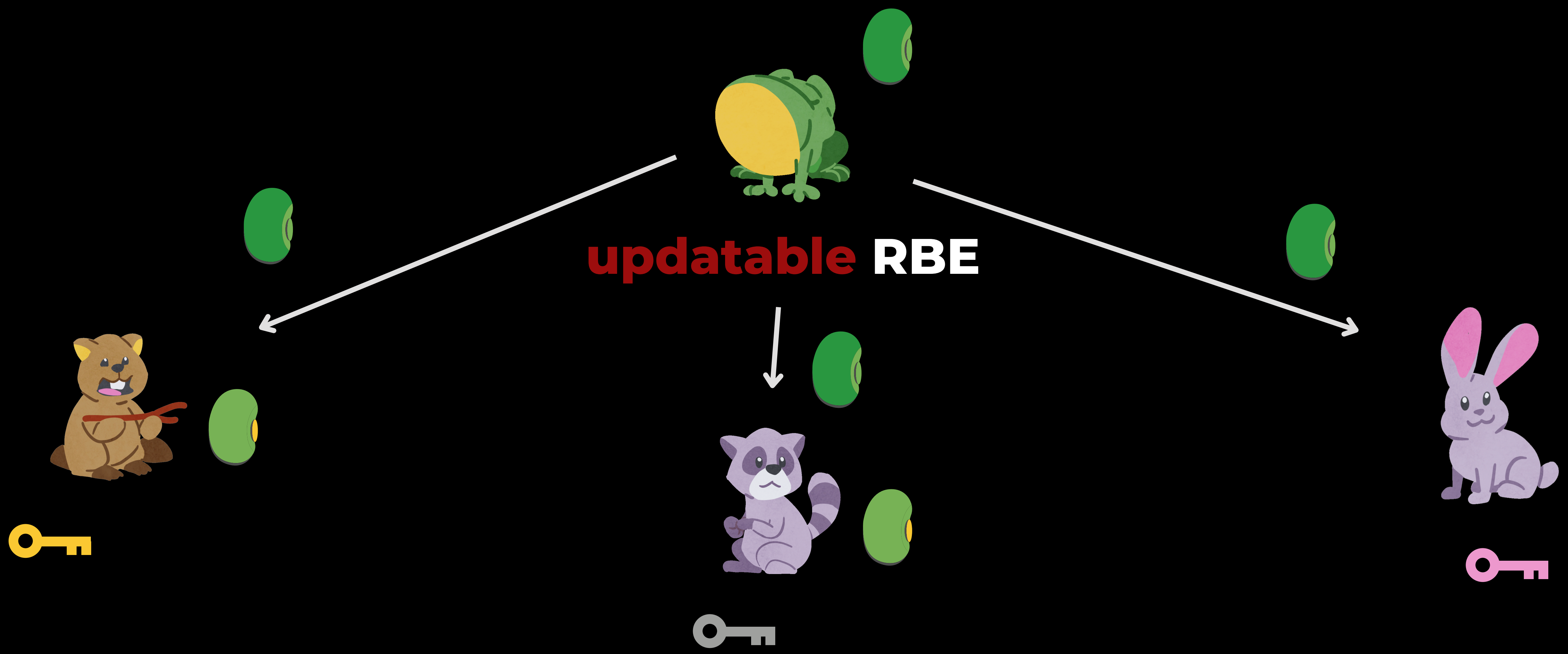
**Stemming from [TCC:Garg-Hajiabadi-Mahmoody-Rahimi-18]**
**Like Identity-Based Encryption (IBE),**
**but without the key curator holding secrets.**

# REGISTRATION-BASED ENCRYPTION (RBE)

**Stemming from [TCC:Garg-Hajiabadi-Mahmoody-Rahimi-18]**
**Like Identity-Based Encryption (IBE),**
**but without the key curator holding secrets.**

# REGISTRATION-BASED ENCRYPTION (RBE)

**Stemming from [TCC:Garg-Hajiabadi-Mahmoody-Rahimi-18]**
**Like Identity-Based Encryption (IBE),**
**but without the key curator holding secrets.**



# REGISTRATION-BASED ENCRYPTION (RBE)

**Stemming from [TCC:Garg-Hajiabadi-Mahmoody-Rahimi-18]**
**Like Identity-Based Encryption (IBE),**
**but without the key curator holding secrets.**

# REGISTRATION-BASED ENCRYPTION (RBE)

**State-of-the-art:**
**- first constructions were very inefficient;**
**- efficient black-box constructions in [Glaeser-Kolonelos-Malavolta-Rahimi-22] but identity-space of polynomial size**
**- and [EC:Döttling-Kolonelos-Lai-Lin-Malavolta-Rahimi-23] with lattices, but ciphertexts in GB**

# REGISTRATION-BASED ENCRYPTION (RBE)

**State-of-the-art:**
**- first constructions were very inefficient;**
**- efficient black-box constructions in [Glaeser-Kolonelos-Malavolta-Rahimi-22] but identity-space of polynomial size**
**- and [EC:Döttling-Kolonelos-Lai-Lin-Malavolta-Rahimi-23] with lattices, but ciphertexts in GB**

| | Setting | $\mathcal{ID}$ | Compactness | $\lvert\mathsf{ct}\rvert$ | #updates | $\lvert\mathsf{pp}\rvert + \lvert\mathsf{crs}\rvert$ |
|---|---|---|---|---|---|---|
| [HLWW23] | Pairings (C) | $\{0,1\}^*$ | Adaptive | $O(\lambda \log n)$ | $\log n$ | $O(\lambda n^{2/3} \log n)$ |
| [GKMR22] | Pairings (P) | $[1, n]$ | Adaptive | $4 \log n$ | $\log n$ | $O(\sqrt{n} \log n)$ |
| Ours P1 | Pairings (P) | $\{0,1\}^*$ | Adaptive | $6\lambda \log n$ | $\log n$ | $O(\sqrt{\lambda n} \log n)$ |
| Ours P2 | Pairings (P) | $\{0,1\}^*$ | Selective | $12 \log n$ | $\log n$ | $O(\sqrt{n} \log n)$ |
| [DKL+23] | Lattices | $\{0,1\}^*$ | Adaptive | $(2\lambda + 1) \log n$ | $\log n$ | $O(\log n)$ |
| Ours L | Lattices | $\{0,1\}^*$ | Selective | $4 \log^2 n$ | $\log n$ | $O(\log n)$ |

Table 1: Comparison of the schemes resulting from different instantiations of our compiler. $n$ is the maximum number of users to be registered. Parings (P) indicates prime order groups and Pairings (C) composite order groups respectively. $\lvert\mathsf{ct}\rvert$ in the pairing construction is measured in group elements and in the Lattice constructions LWE ciphertexts.

# A TECHNIQUE BASED ON CUCKOO HASHING

A NEW SETTING FOR CUCKOO HASHING

# CUCKOO HASHING

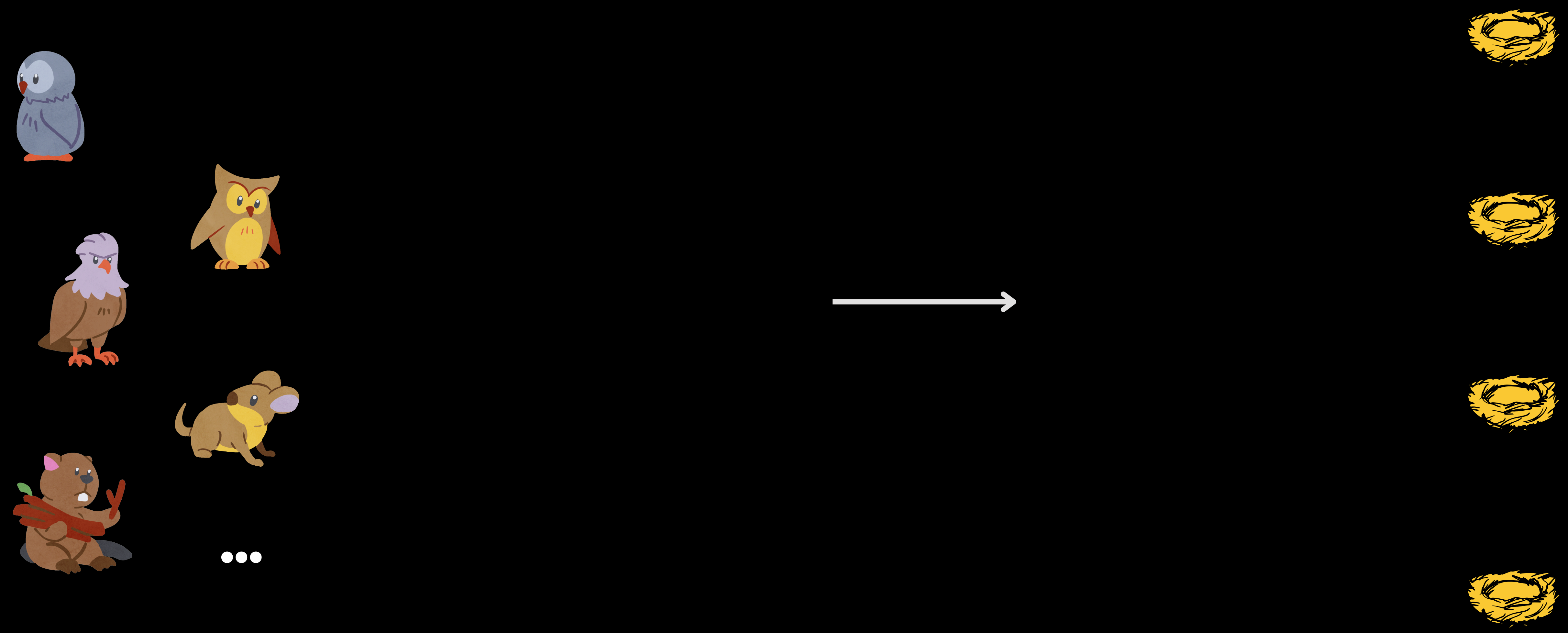
**A powerful technique**

# CUCKOO HASHING

## A powerful technique

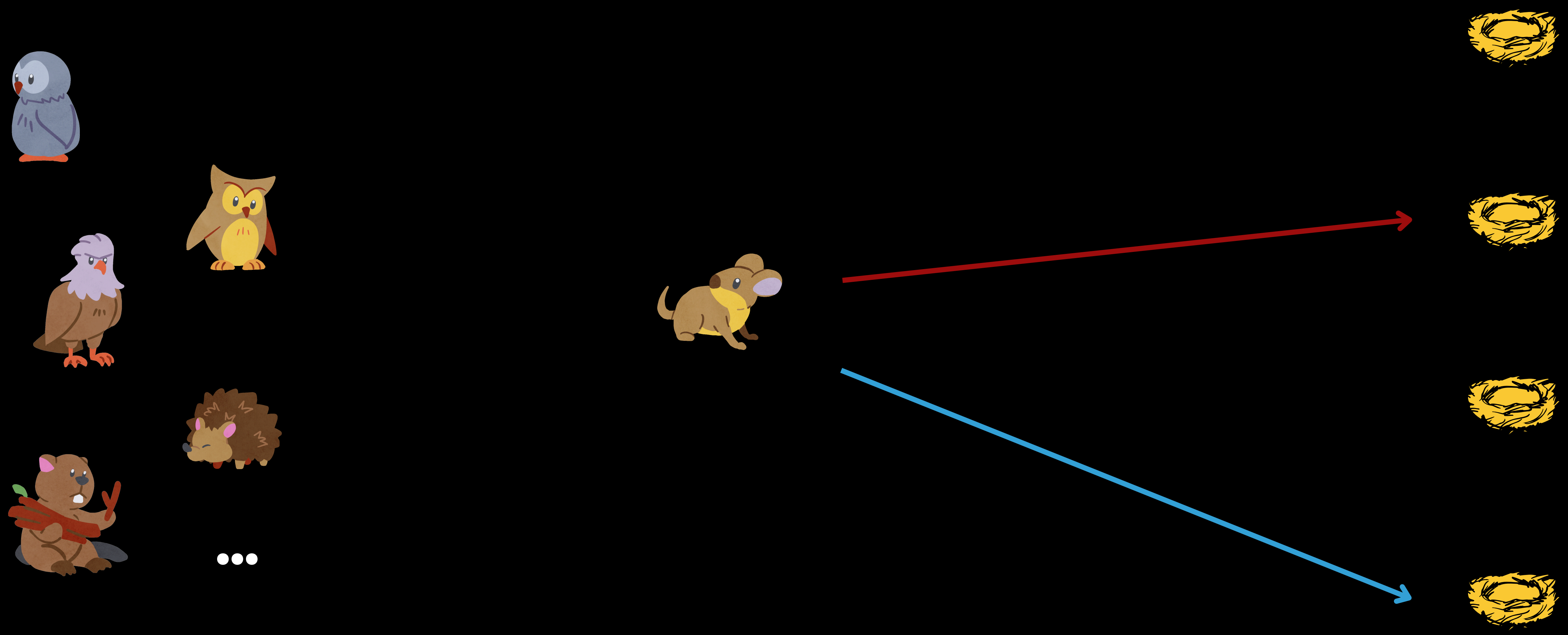
# CUCKOO HASHING

## A powerful technique



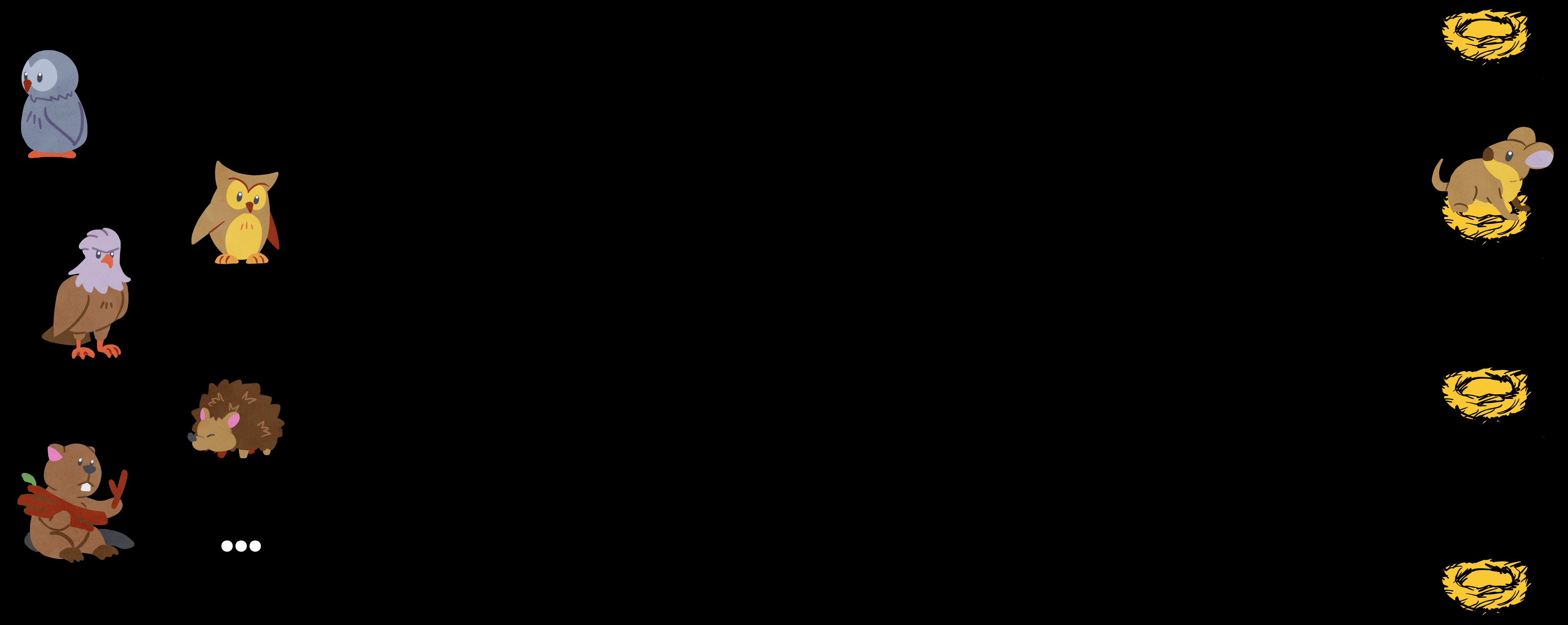
# CUCKOO HASHING

**A powerful technique; an example with two hash functions, and one animal per nest**



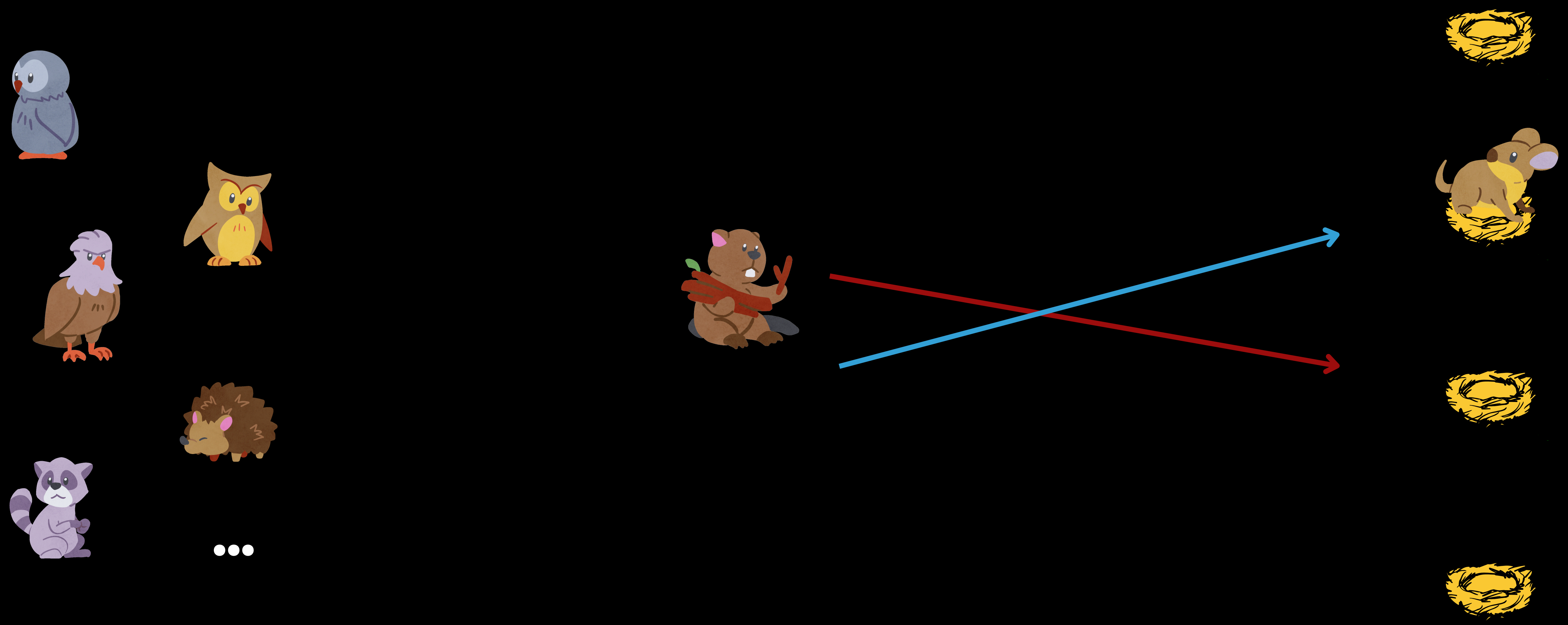
# CUCKOO HASHING

**A powerful technique; an example with two hash functions, and one animal per nest**



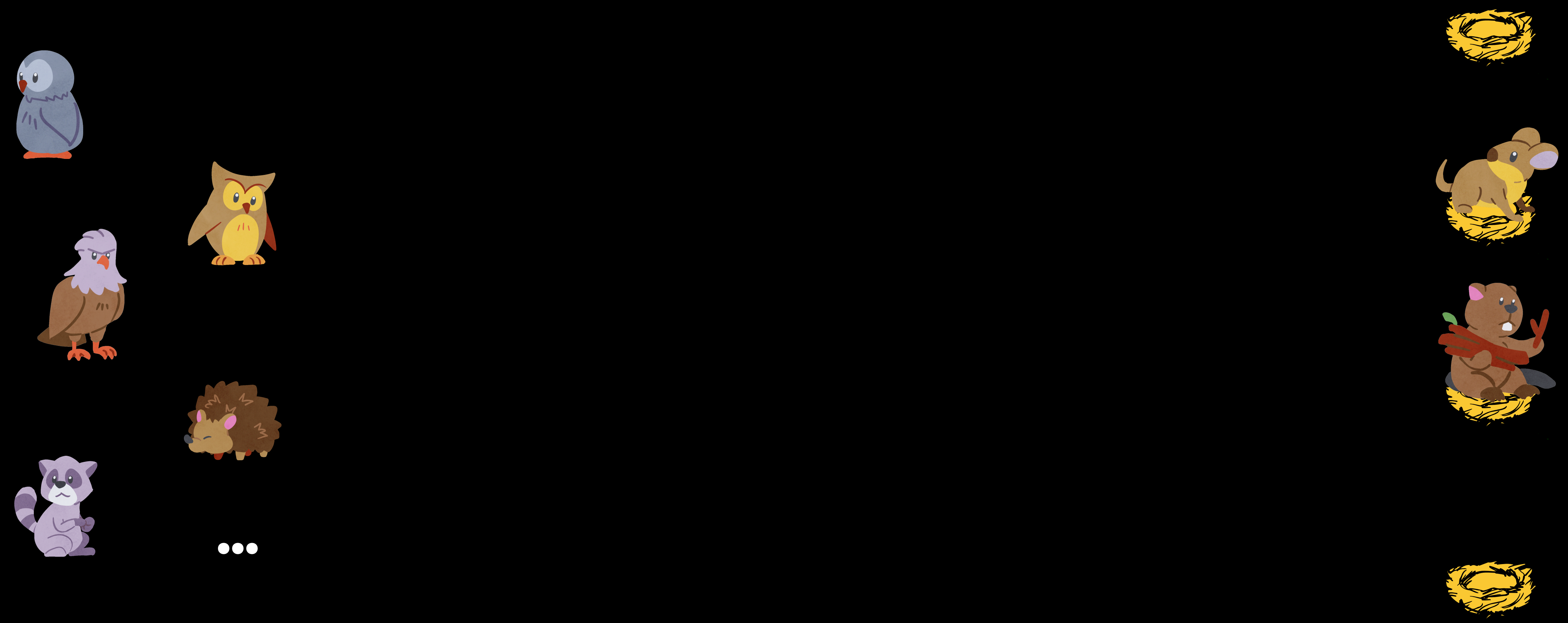
# CUCKOO HASHING

**A powerful technique; an example with two hash functions, and one animal per nest**



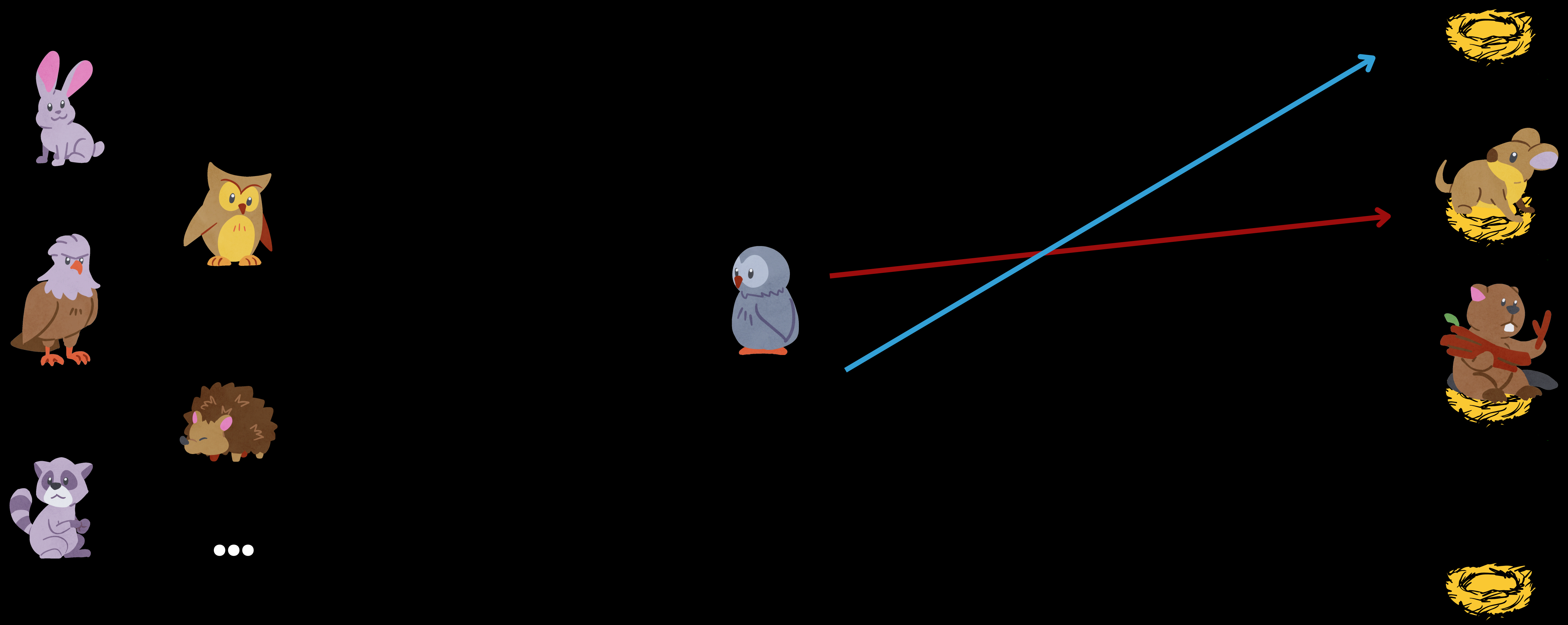
# CUCKOO HASHING

**A powerful technique; an example with two hash functions, and one animal per nest**



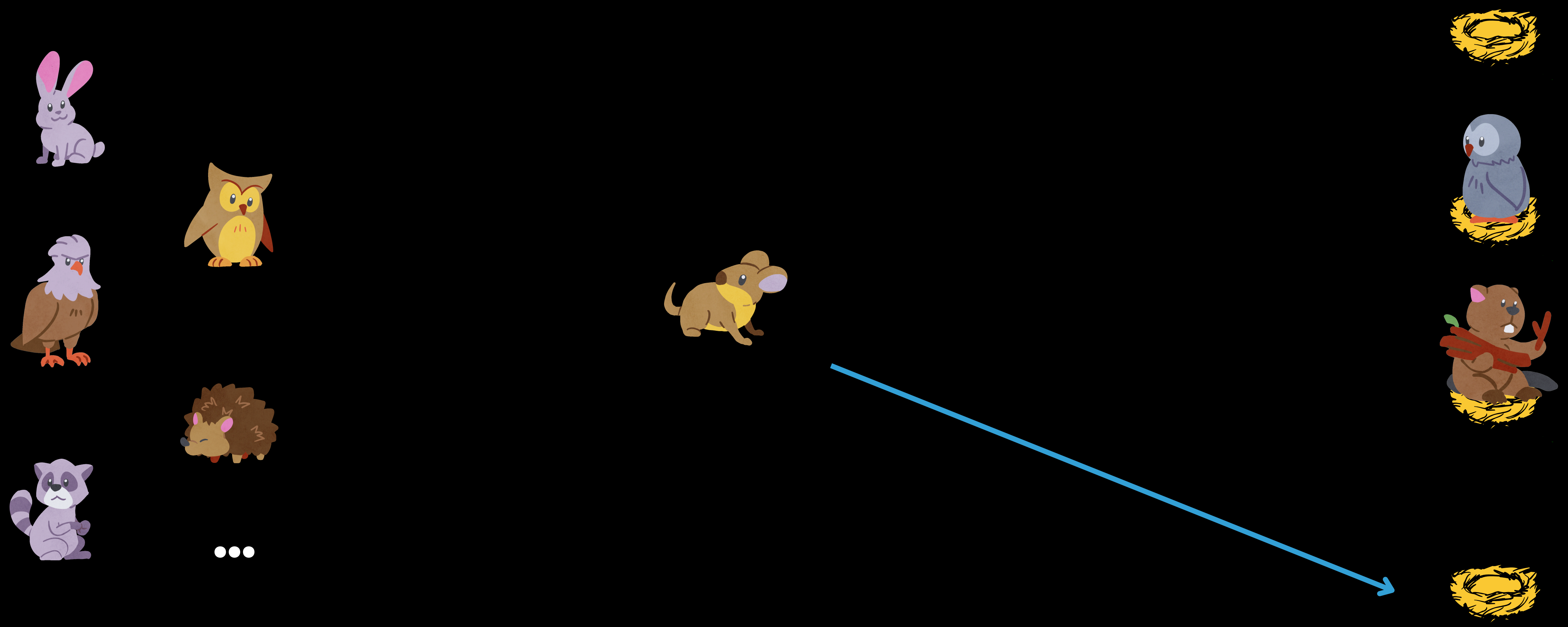
# CUCKOO HASHING

**A powerful technique; an example with two hash functions, and one animal per nest**



# CUCKOO HASHING

**A powerful technique; an example with two hash functions, and one animal per nest**

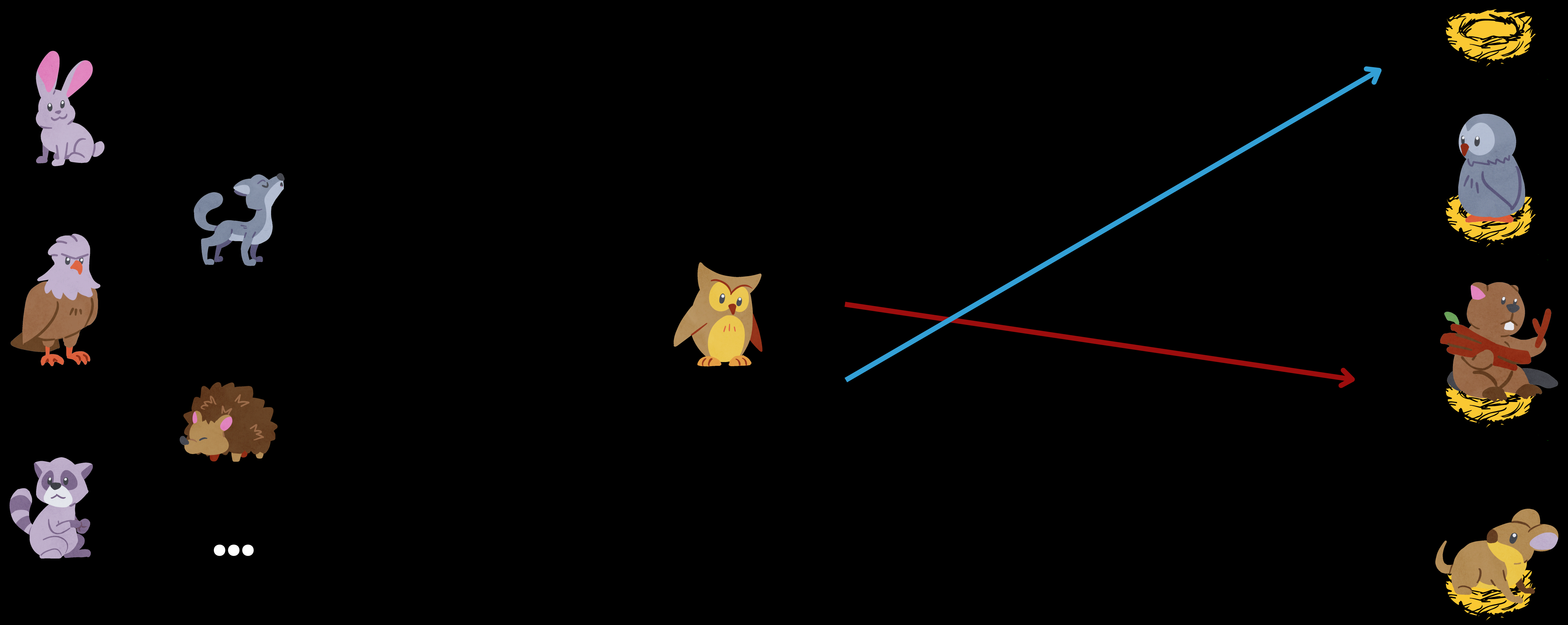


# CUCKOO HASHING

**A powerful technique; an example with two hash functions, and one animal per nest**

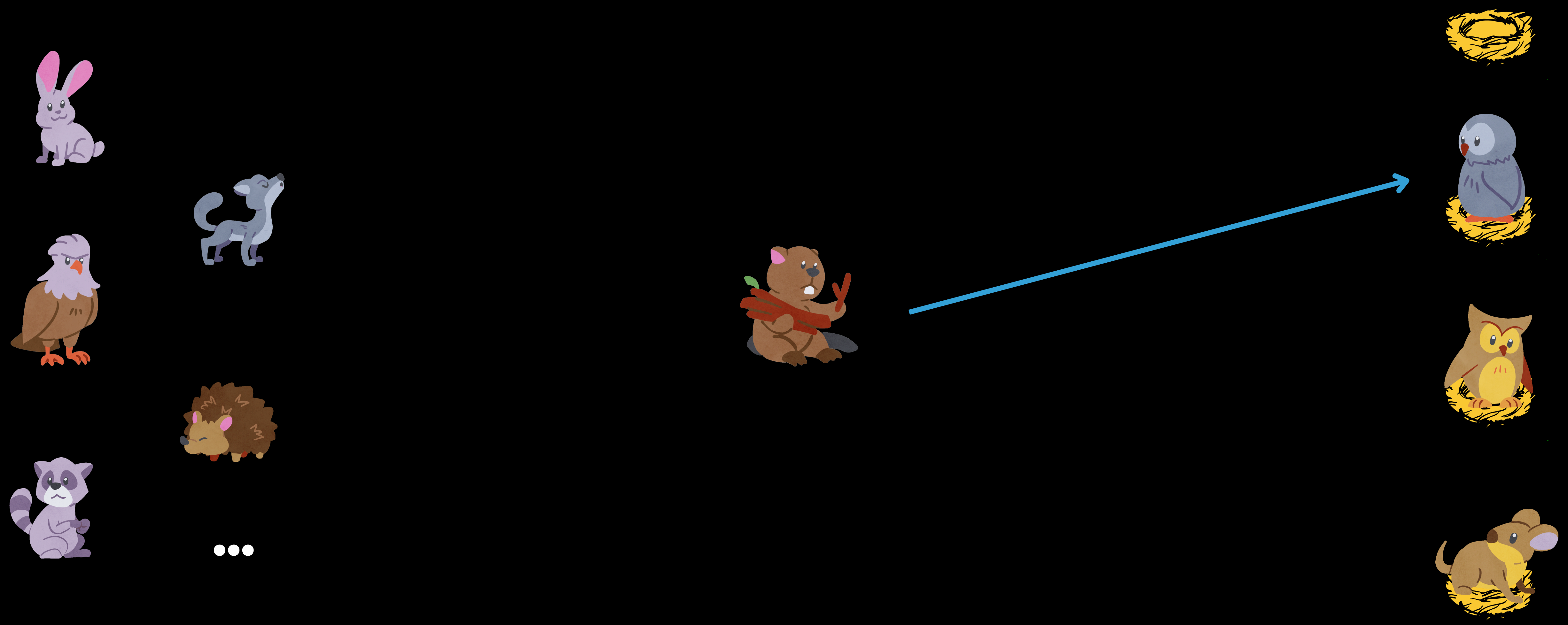
# CUCKOO HASHING

**A powerful technique; an example with two hash functions, and one animal per nest**



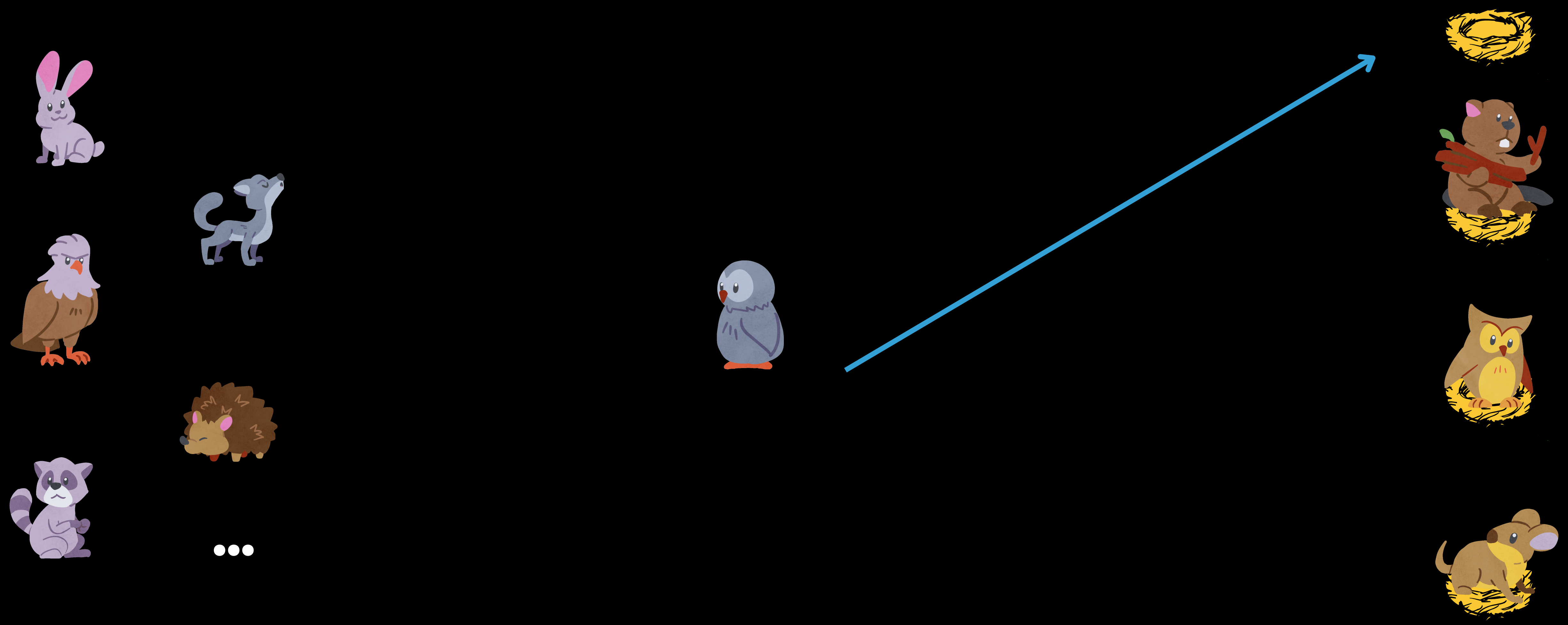
# CUCKOO HASHING

**A powerful technique; an example with two hash functions, and one animal per nest**



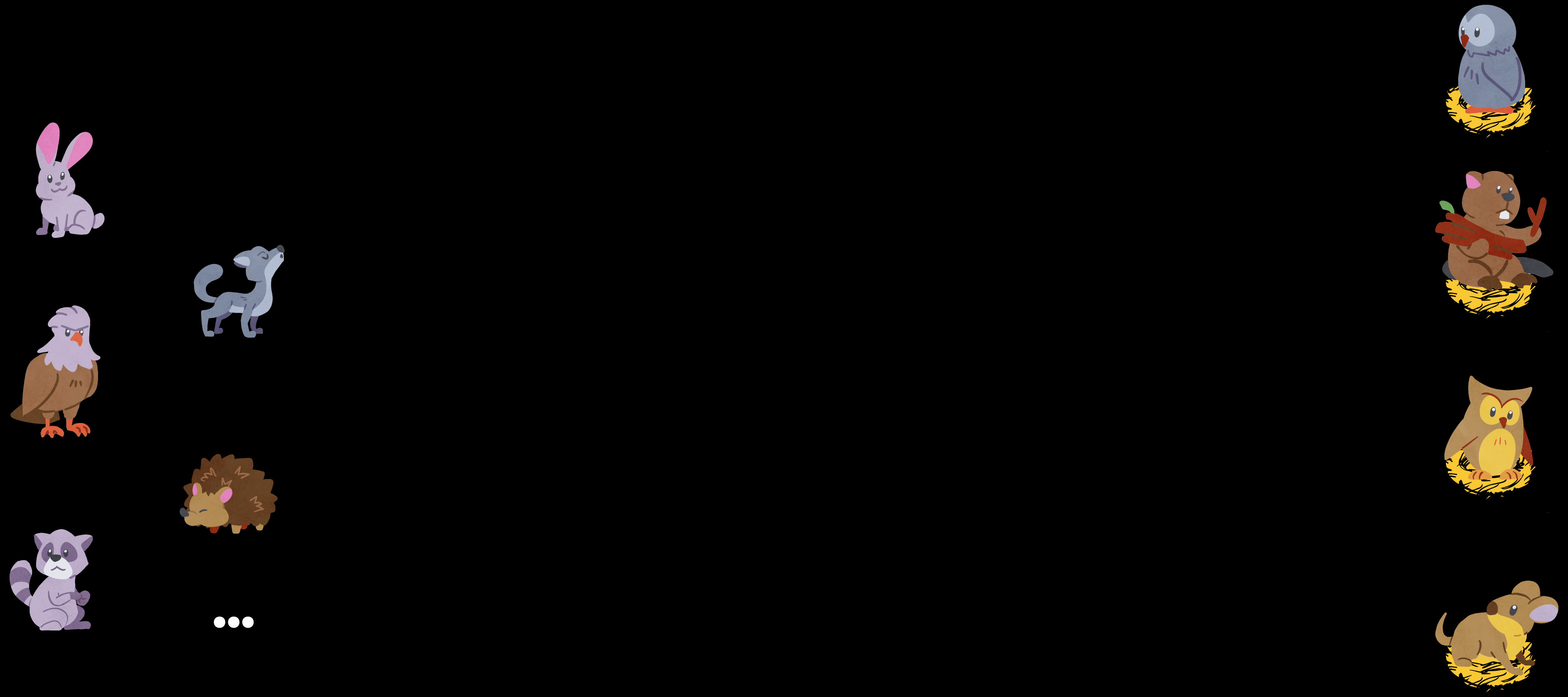
# CUCKOO HASHING

**A powerful technique; an example with two hash functions, and one animal per nest**



# CUCKOO HASHING

**A powerful technique; an example with two hash functions, and one animal per nest**

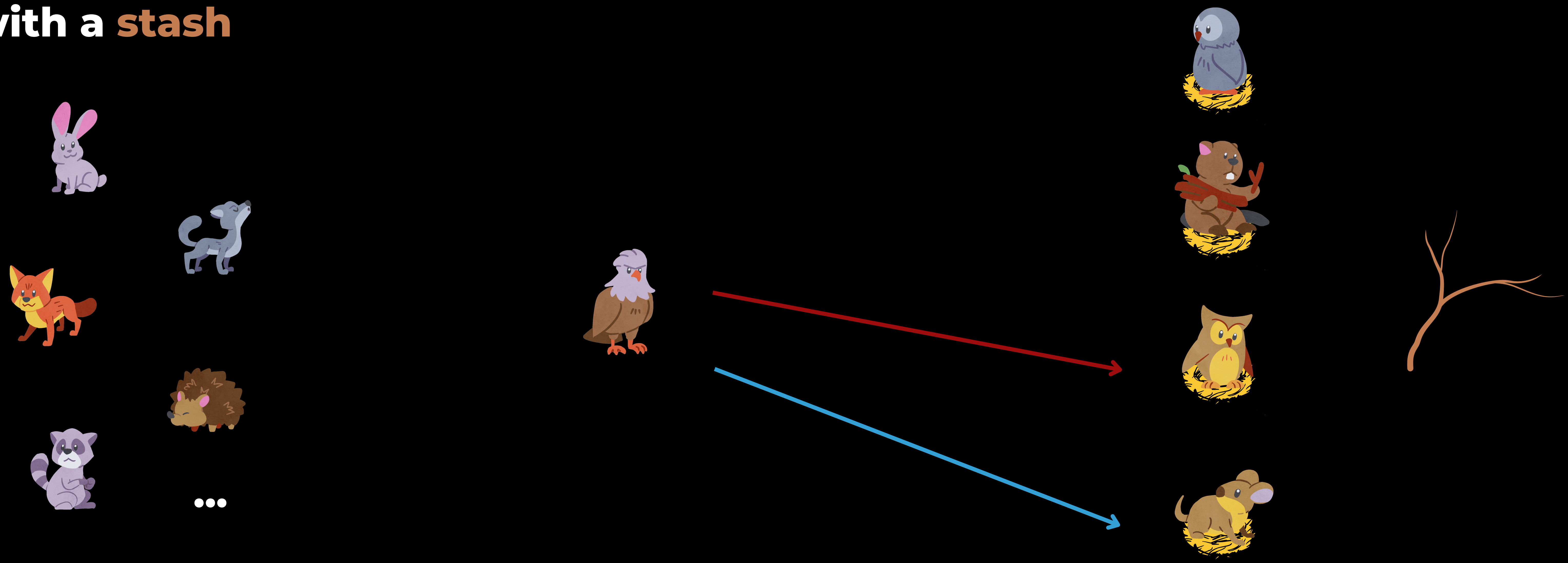


# CUCKOO HASHING

**A powerful technique; an example with two hash functions, and one animal per nest and with a stash**

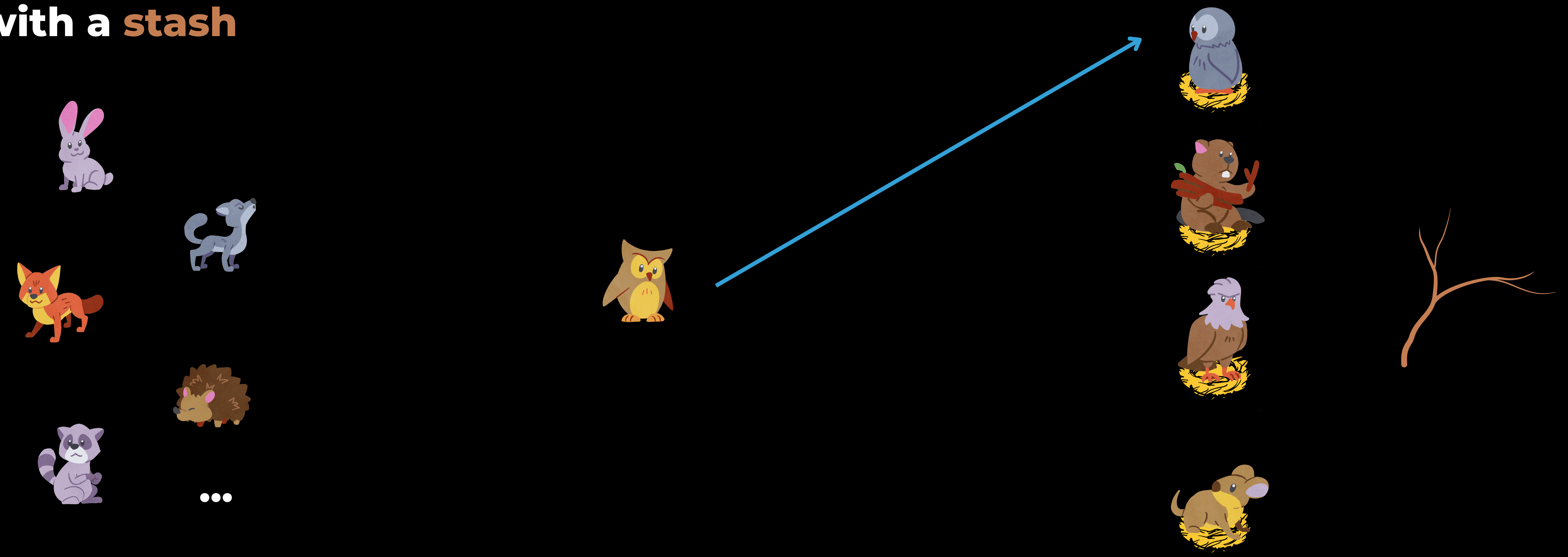
# CUCKOO HASHING

**A powerful technique; an example with two hash functions, and one animal per nest**
**and with a stash**



# CUCKOO HASHING

**A powerful technique; an example with two hash functions, and one animal per nest and with a stash**

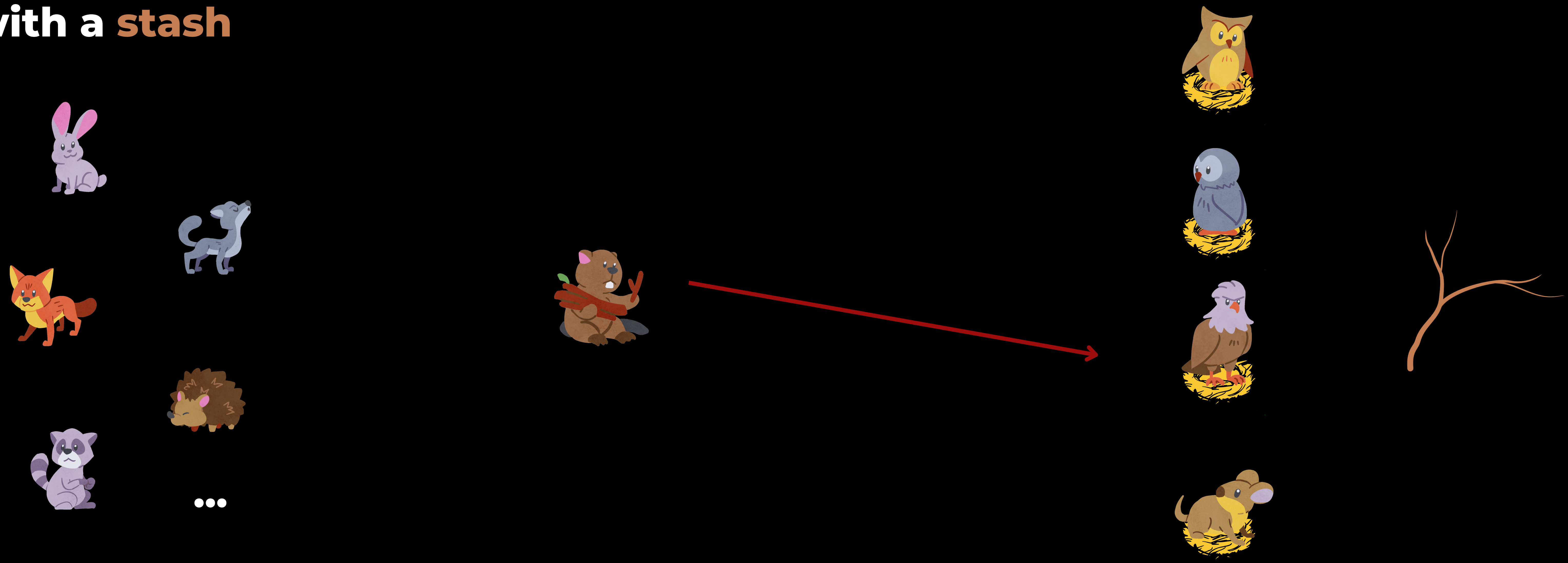


# CUCKOO HASHING

**A powerful technique; an example with two hash functions, and one animal per nest and with a stash**

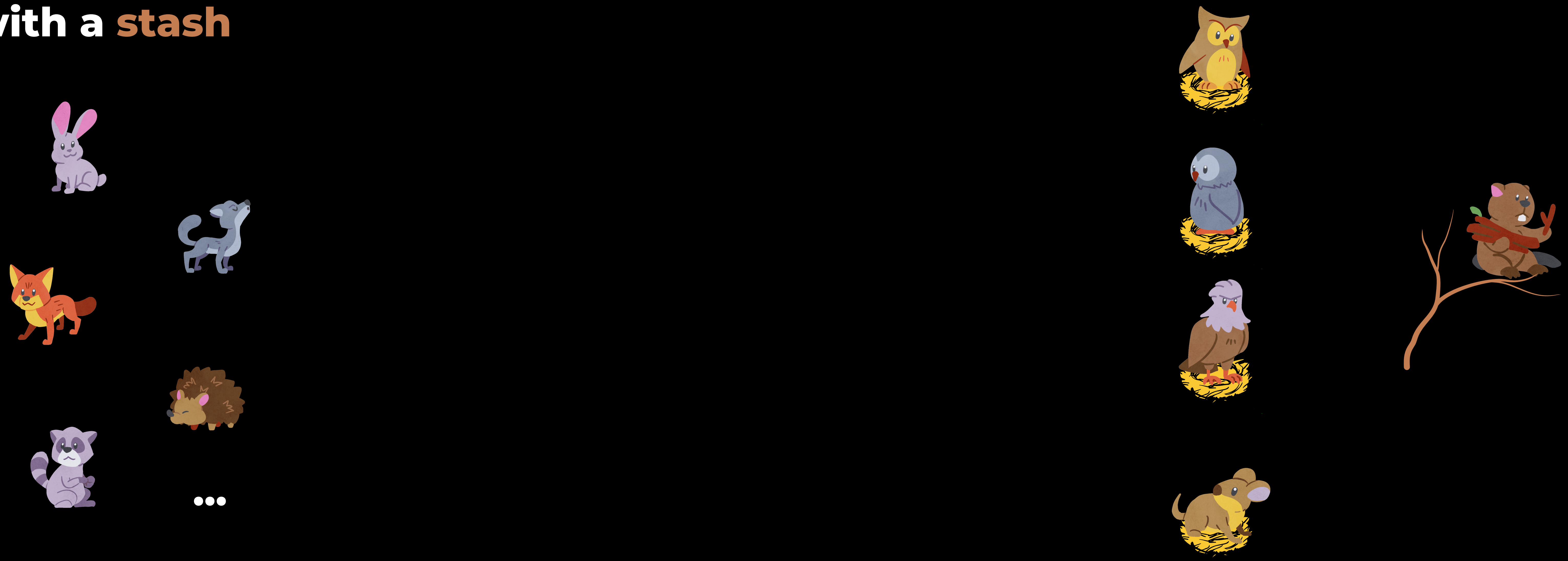

# CUCKOO HASHING

**A powerful technique; an example with two hash functions, and one animal per nest and with a stash**



# CUCKOO HASHING

A powerful technique; an example with two hash functions,
and one animal per nest
and with a stash

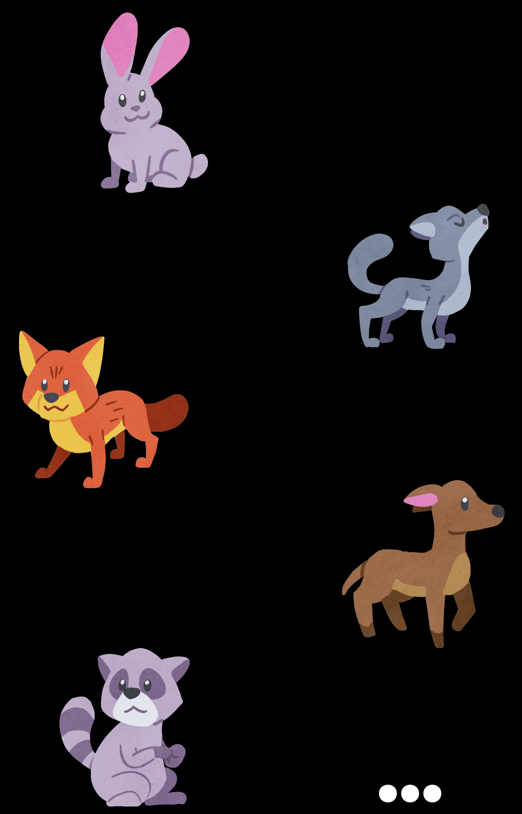


# CUCKOO HASHING

**A powerful technique; an example with two hash functions, and one animal per nest and with a stash**

# CUCKOO HASHING

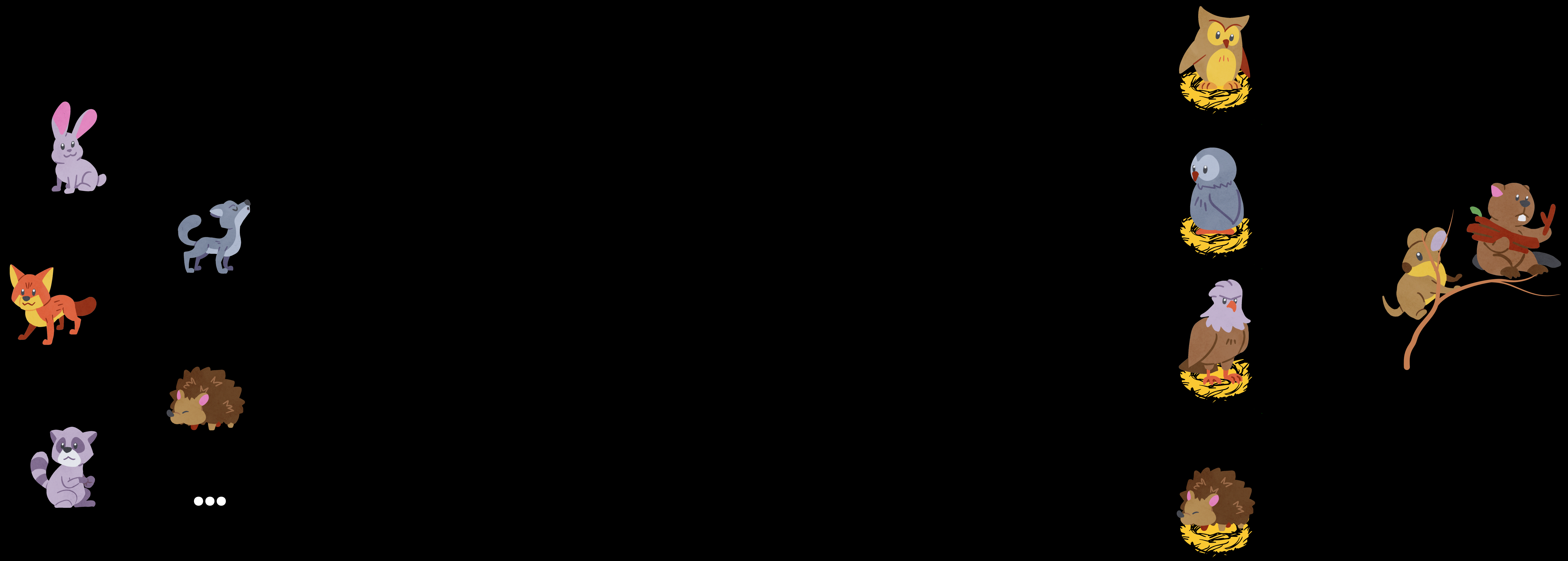
**A powerful technique; an example with two hash functions, and one animal per nest**
**and with a stash**



...

# CUCKOO HASHING

## Performance



# CUCKOO HASHING

**Performance, for negligible failure (in λ):**
**- h = 2 hash functions, N = 2hn nests with capacity one, to store n animals:**
**average constant insertion time, worst-case log(n) stash.**

# CUCKOO HASHING

**Performance, for negligible failure (in λ):**
**- h = 2 hash functions, N = 2hn nests with capacity one, to store n animals:**
**average constant insertion time, worst-case log(n) stash.**
**Security against adversaries wanting to fill the stash?**
(choosing animals maliciously)

# CUCKOO HASHING

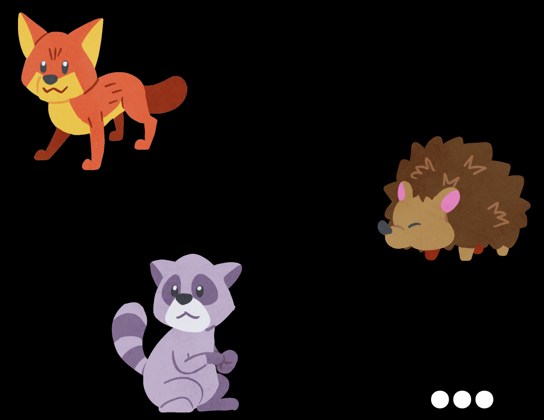
**Performance, for negligible failure (in λ):**

**- h = 2 hash functions, N = 2hn nests with capacity one, to store n animals: average constant insertion time, worst-case log(n) stash.**

**Security against adversaries wanting to fill the stash?**

(choosing animals maliciously)

**- h = 2, N = 2kn nests: average constant insertion, worst-case n stash**



...

# CUCKOO HASHING

**Performance, for negligible failure (in $\lambda$):**

**- $h = 2$ hash functions, $N = 2hn$ nests with capacity one, to store $n$ animals: average constant insertion time, worst-case $\log(n)$ stash.**

**Security against adversaries wanting to fill the stash?**

(choosing animals maliciously)

**- $h = 2$, $N = 2kn$ nests: average constant insertion, worst-case $n$ stash**

**- $h = \lambda$, $N = 2\lambda n$ nests: average $\lambda$ time insertion, worst-case empty stash**

# CUCKOO HASHING

**Performance, for negligible failure (in λ):**

**- h = 2 hash functions, N = 2hn nests with capacity one, to store n animals: average constant insertion time, worst-case log(n) stash.**

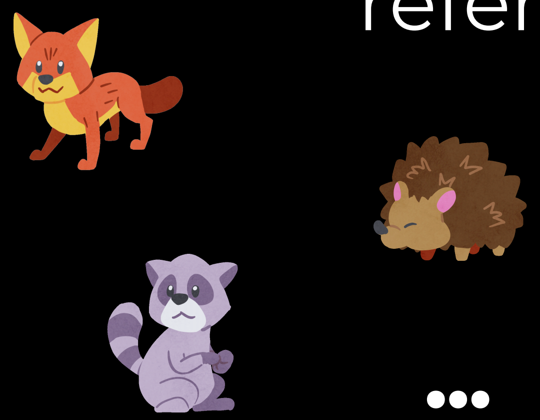
**Security against adversaries wanting to fill the stash?**

(choosing animals maliciously)

**- h = 2, N = 2kn nests: average constant insertion, worst-case n stash**

**- h = λ, N = 2λn nests: average λ time insertion, worst-case empty stash**

reference for parameters in cryptography: [C:Yeo23]



...

# OUR CONSTRUCTION

**USING CUCKOO HASHING WITH VECTOR COMMITMENTS (VC), AND WITNESS ENCRYPTION FOR VC (VCWE)**

# THE GLAESER-KOLONELOS-MALAVOLTA-RAHIMI (GKMR) RBE, USING LIBERT-YUNG VECTOR COMMITMENTS [TCC:LY10]

$$\mathsf{crs} = g_1, g_2, g_3, g_5, g_6$$

 $= 1$

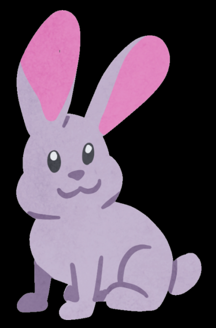
$\longrightarrow \; = g_1$

$g_2, g_3$

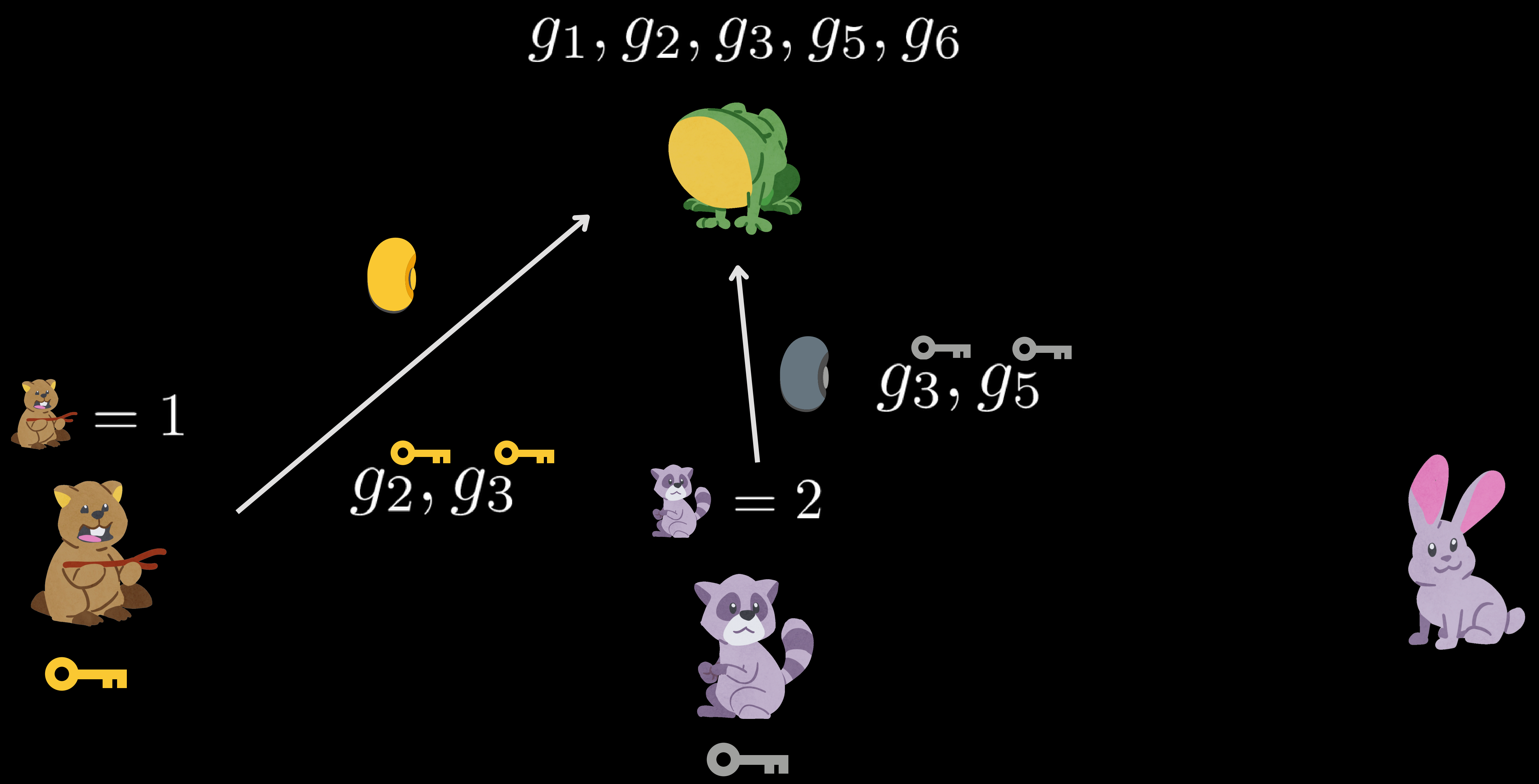
 $= 2$

$g_3, g_5$

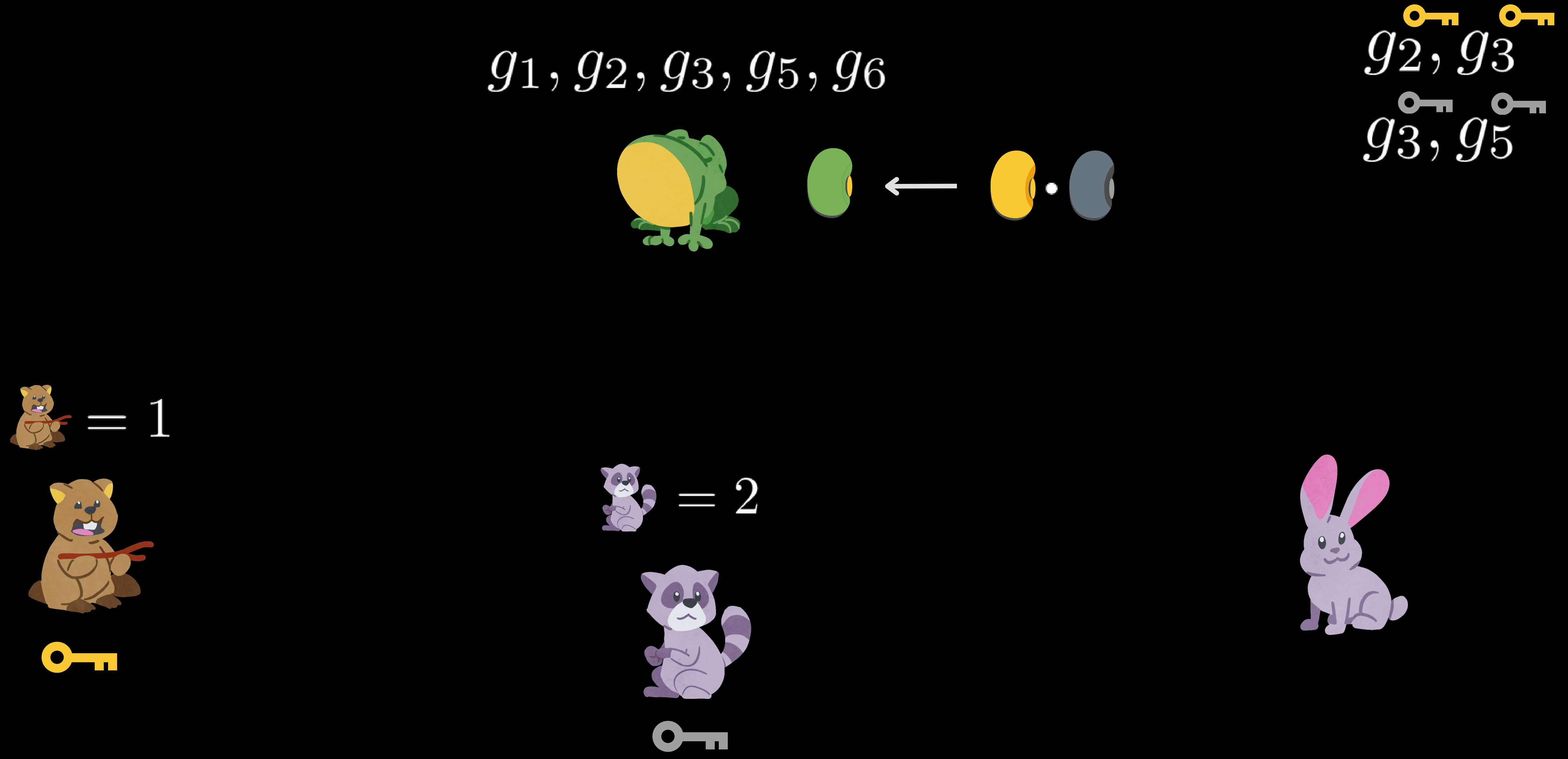
$\longrightarrow \; = g_2$



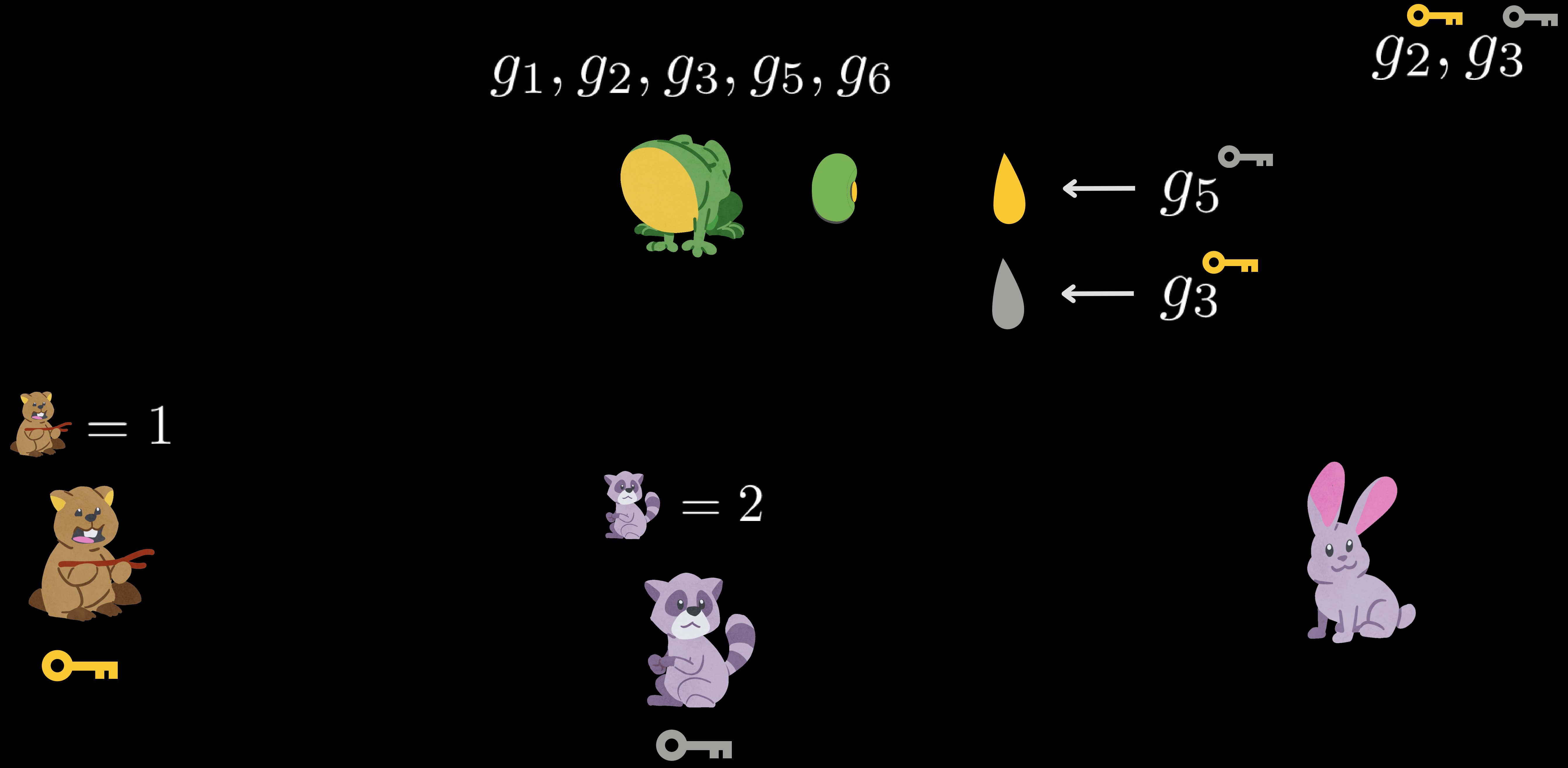
# THE GLAESER-KOLONELOS-MALAVOLTA-RAHIMI (GKMR) RBE, USING LIBERT-YUNG VECTOR COMMITMENTS [TCC:LY10]



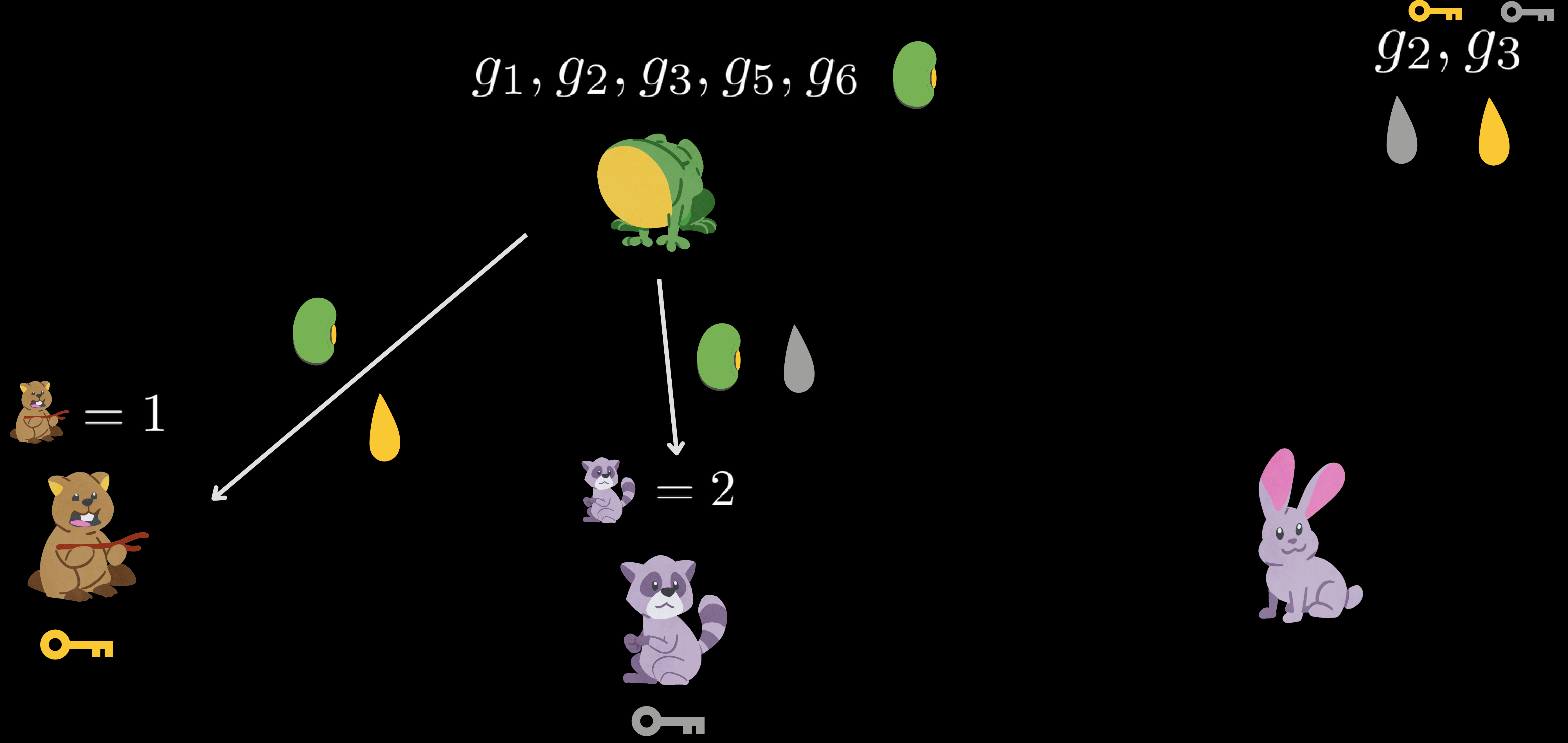
# THE GLAESER-KOLONELOS-MALAVOLTA-RAHIMI (GKMR) RBE, USING LIBERT-YUNG VECTOR COMMITMENTS [TCC:LY10]



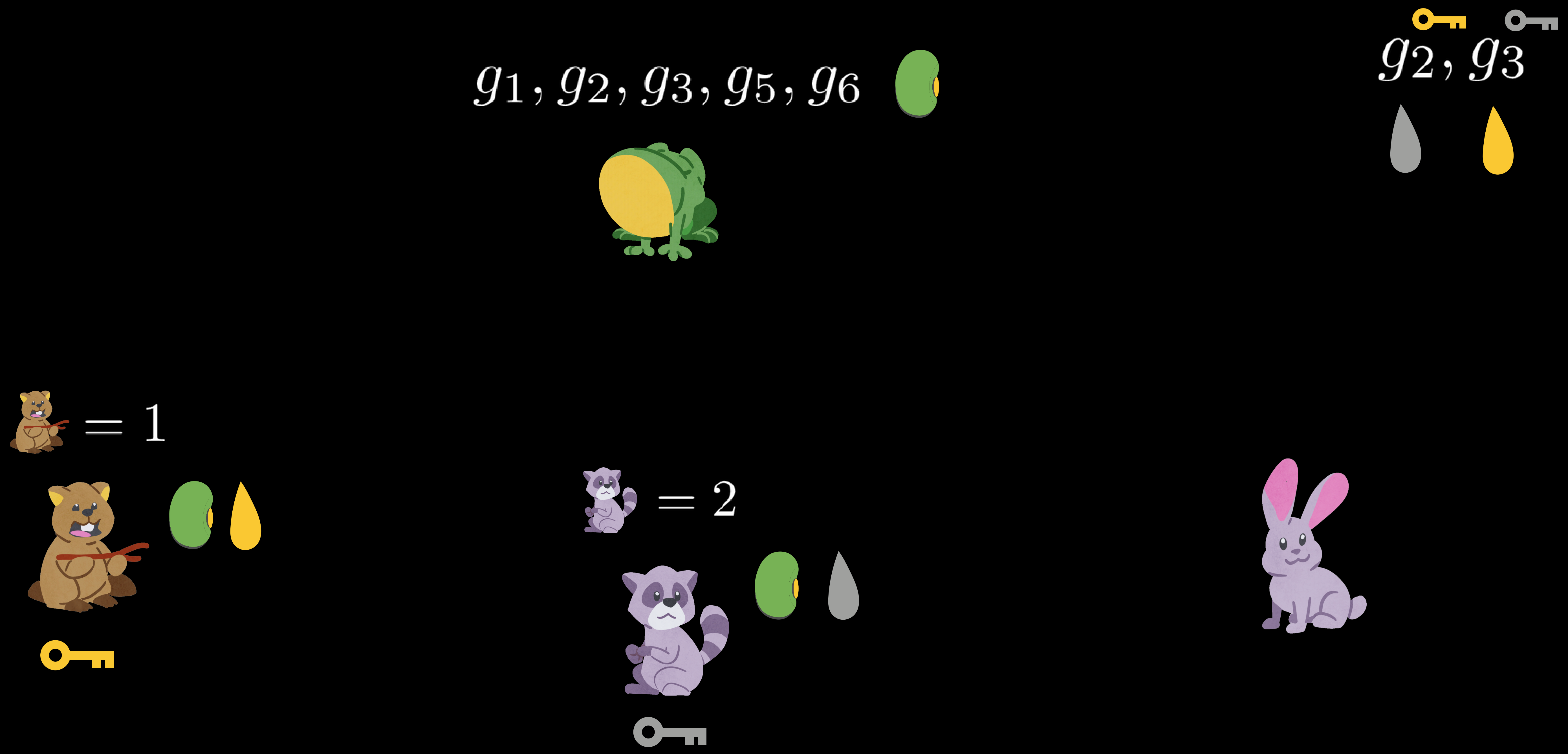
# THE GLAESER-KOLONELOS-MALAVOLTA-RAHIMI (GKMR) RBE, USING LIBERT-YUNG VECTOR COMMITMENTS [TCC:LY10]



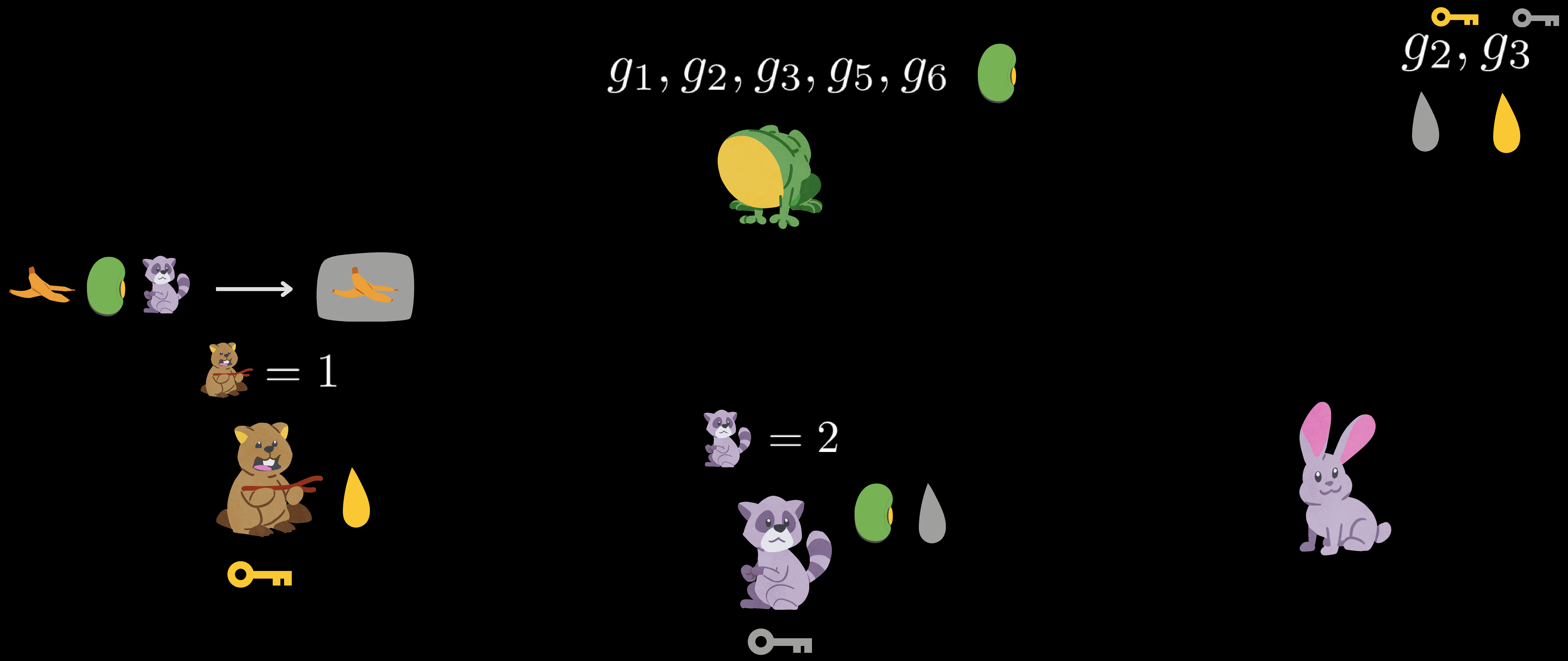
# THE GLAESER-KOLONELOS-MALAVOLTA-RAHIMI (GKMR) RBE, USING LIBERT-YUNG VECTOR COMMITMENTS [TCC:LY10]



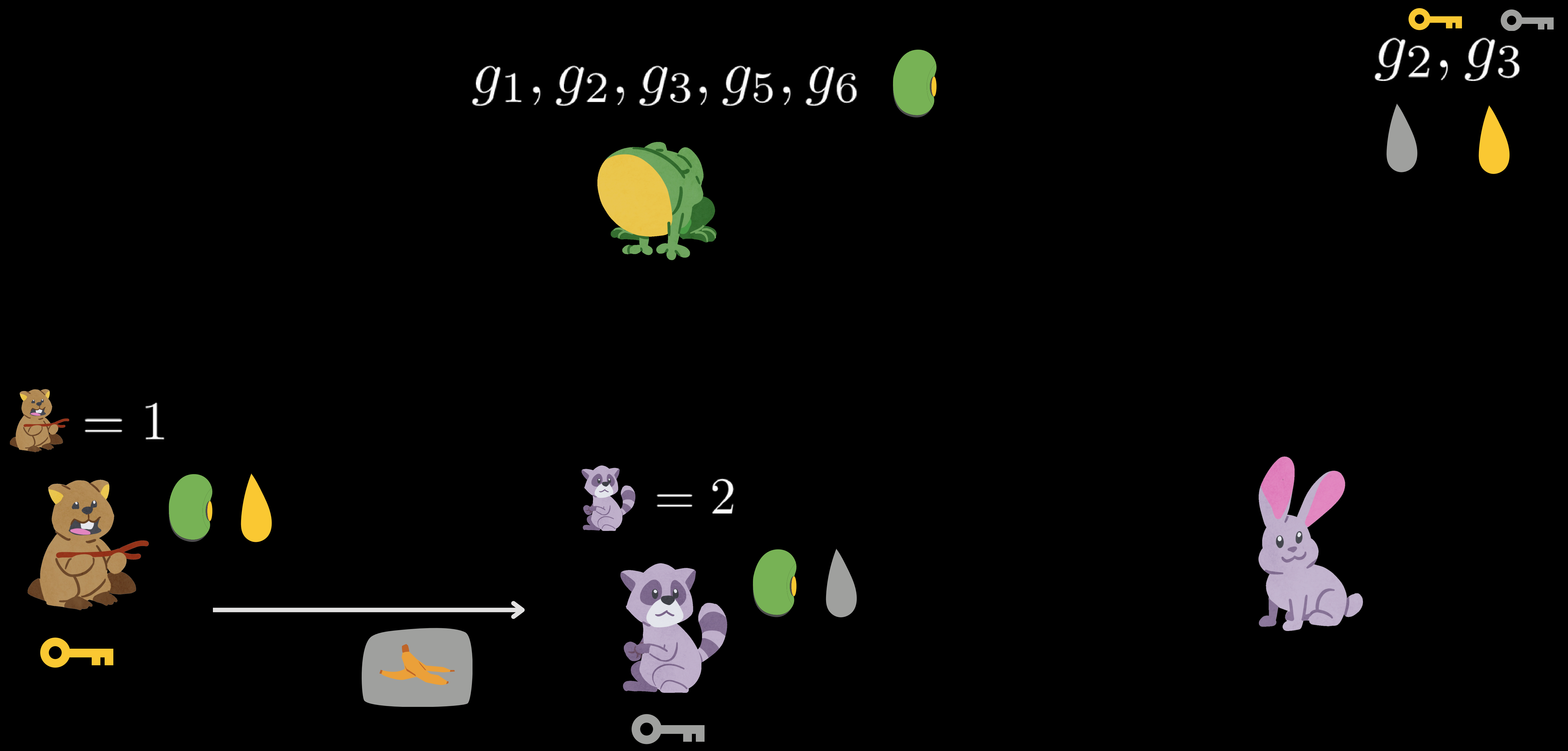
# THE GLAESER-KOLONELOS-MALAVOLTA-RAHIMI (GKMR) RBE, USING LIBERT-YUNG VECTOR COMMITMENTS [TCC:LY10]



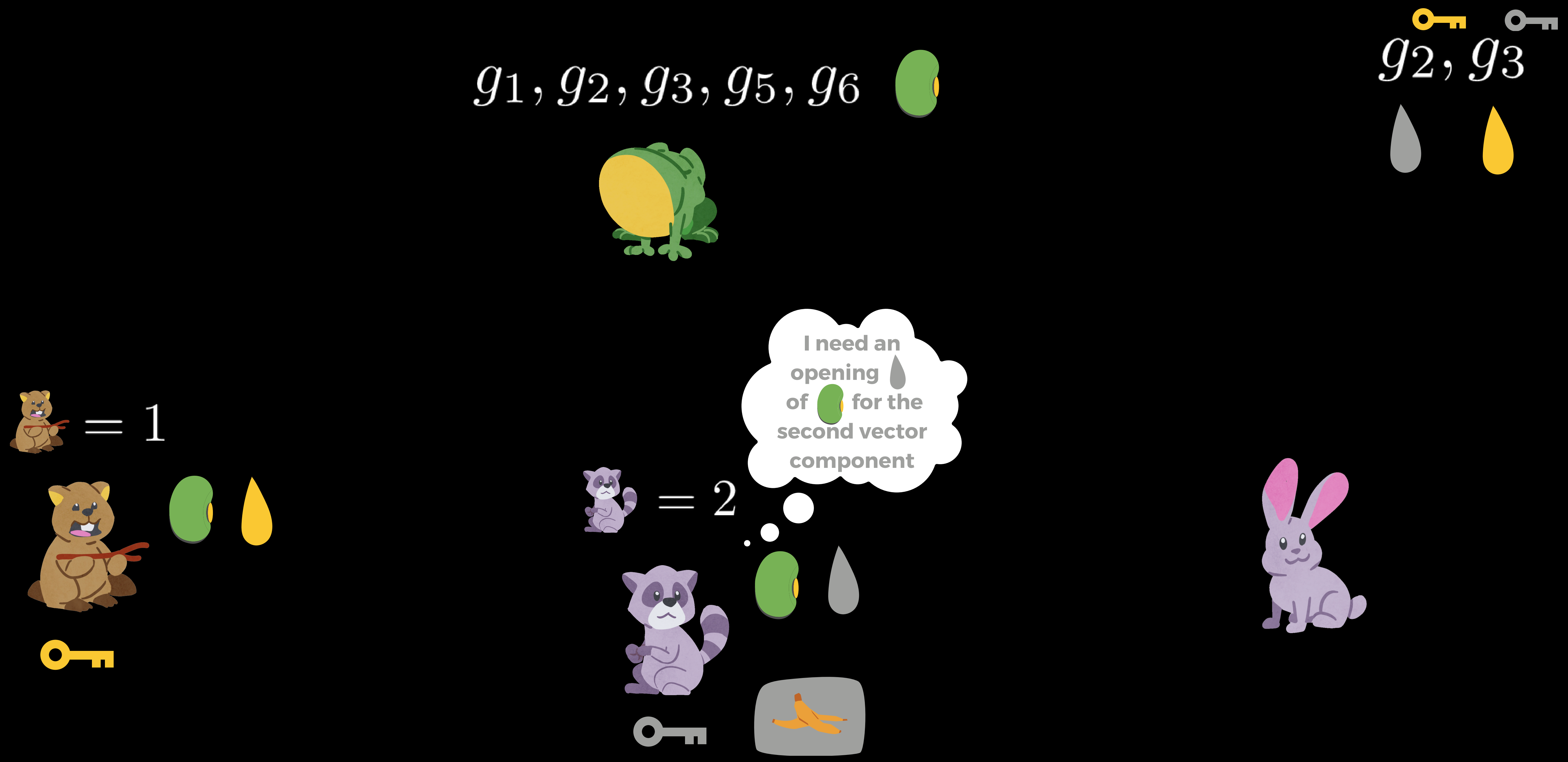
# THE GLAESER-KOLONELOS-MALAVOLTA-RAHIMI (GKMR) RBE, USING LIBERT-YUNG VECTOR COMMITMENTS [TCC:LY10]



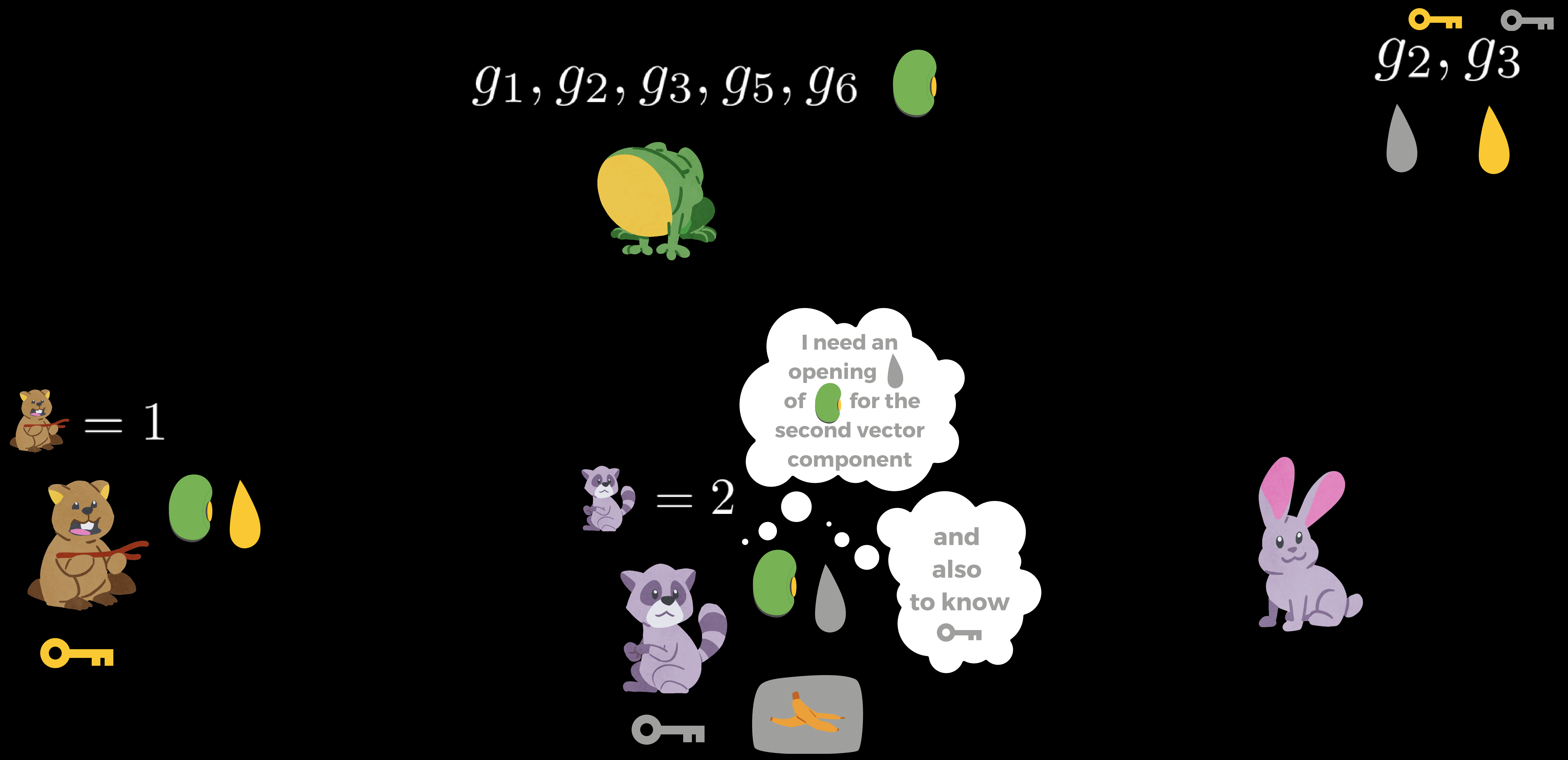
# THE GLAESER-KOLONELOS-MALAVOLTA-RAHIMI (GKMR) RBE, USING LIBERT-YUNG VECTOR COMMITMENTS [TCC:LY10]



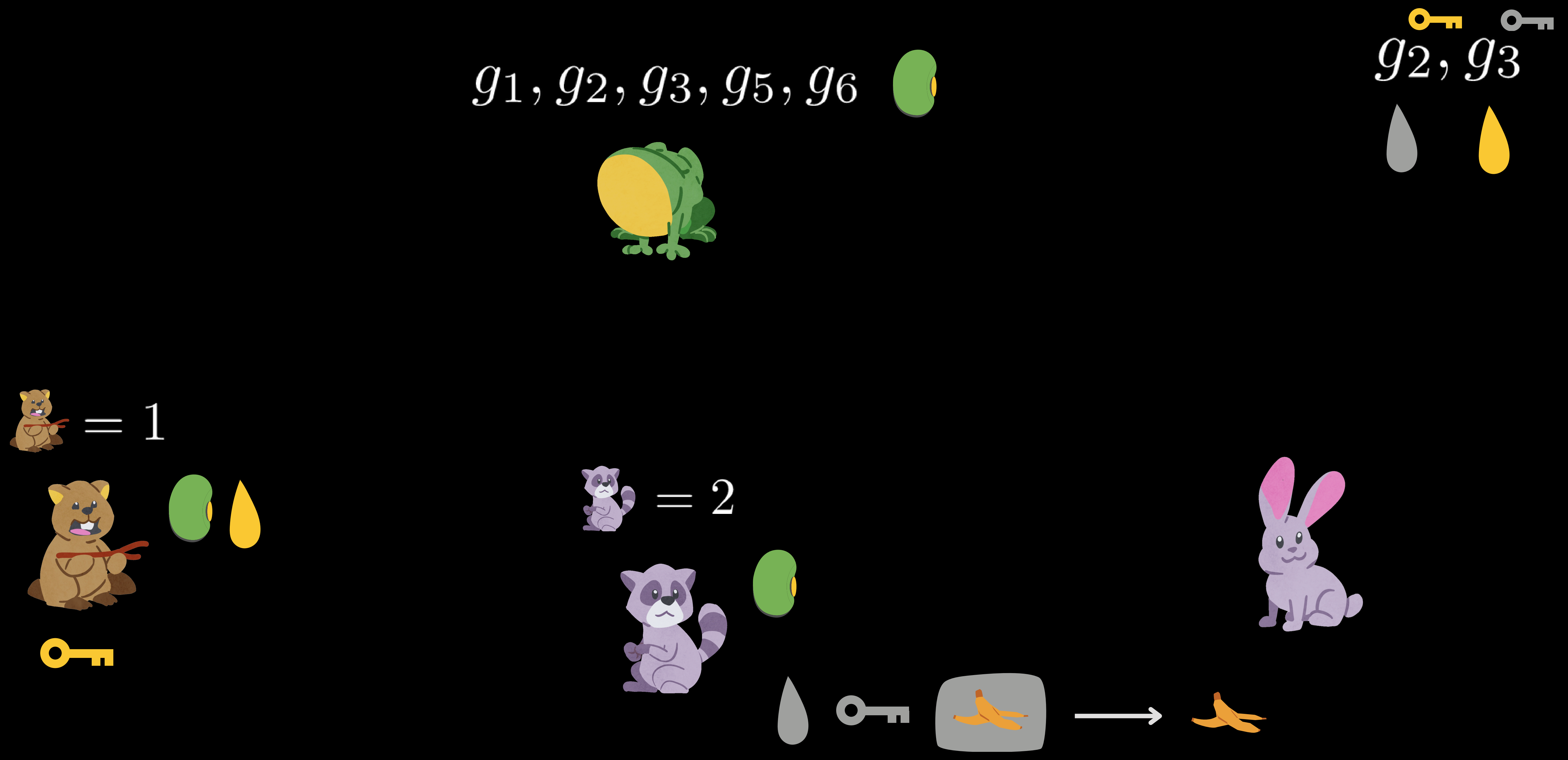
# THE GLAESER-KOLONELOS-MALAVOLTA-RAHIMI (GKMR) RBE, USING LIBERT-YUNG VECTOR COMMITMENTS [TCC:LY10]



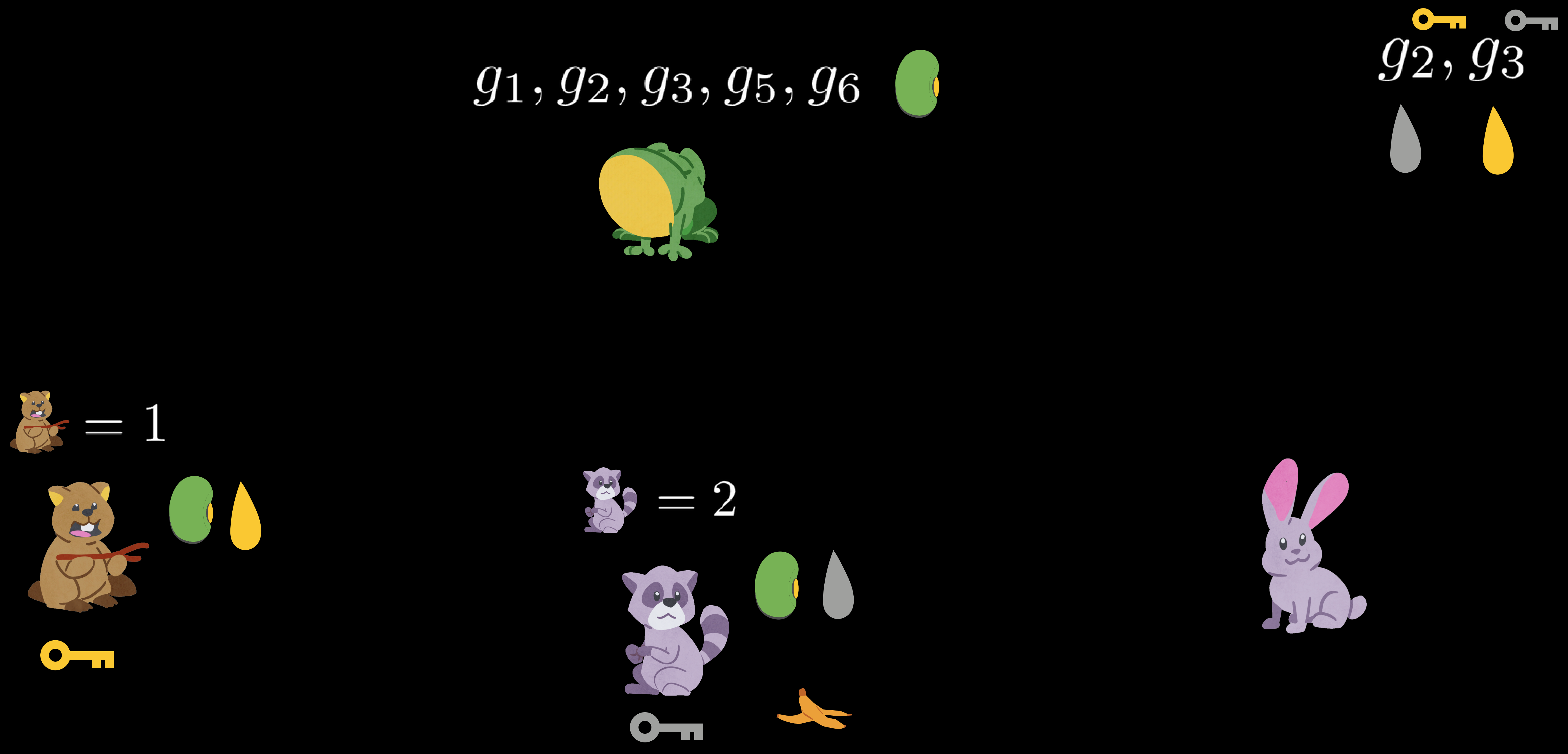
# THE GLAESER-KOLONELOS-MALAVOLTA-RAHIMI (GKMR) RBE, USING LIBERT-YUNG VECTOR COMMITMENTS [TCC:LY10]



# THE GLAESER-KOLONELOS-MALAVOLTA-RAHIMI (GKMR) RBE, USING LIBERT-YUNG VECTOR COMMITMENTS [TCC:LY10]

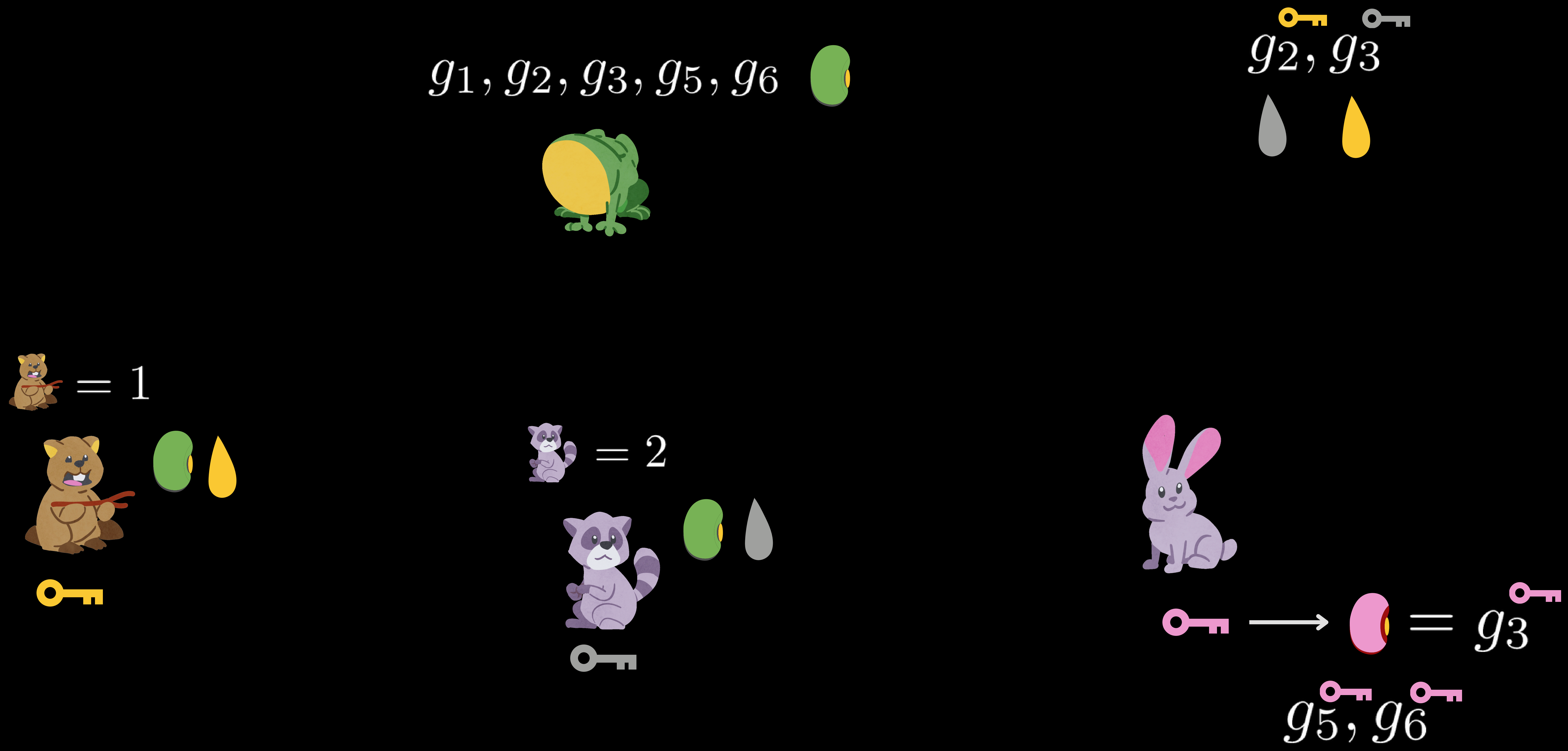


# THE GLAESER-KOLONELOS-MALAVOLTA-RAHIMI (GKMR) RBE, USING LIBERT-YUNG VECTOR COMMITMENTS [TCC:LY10]



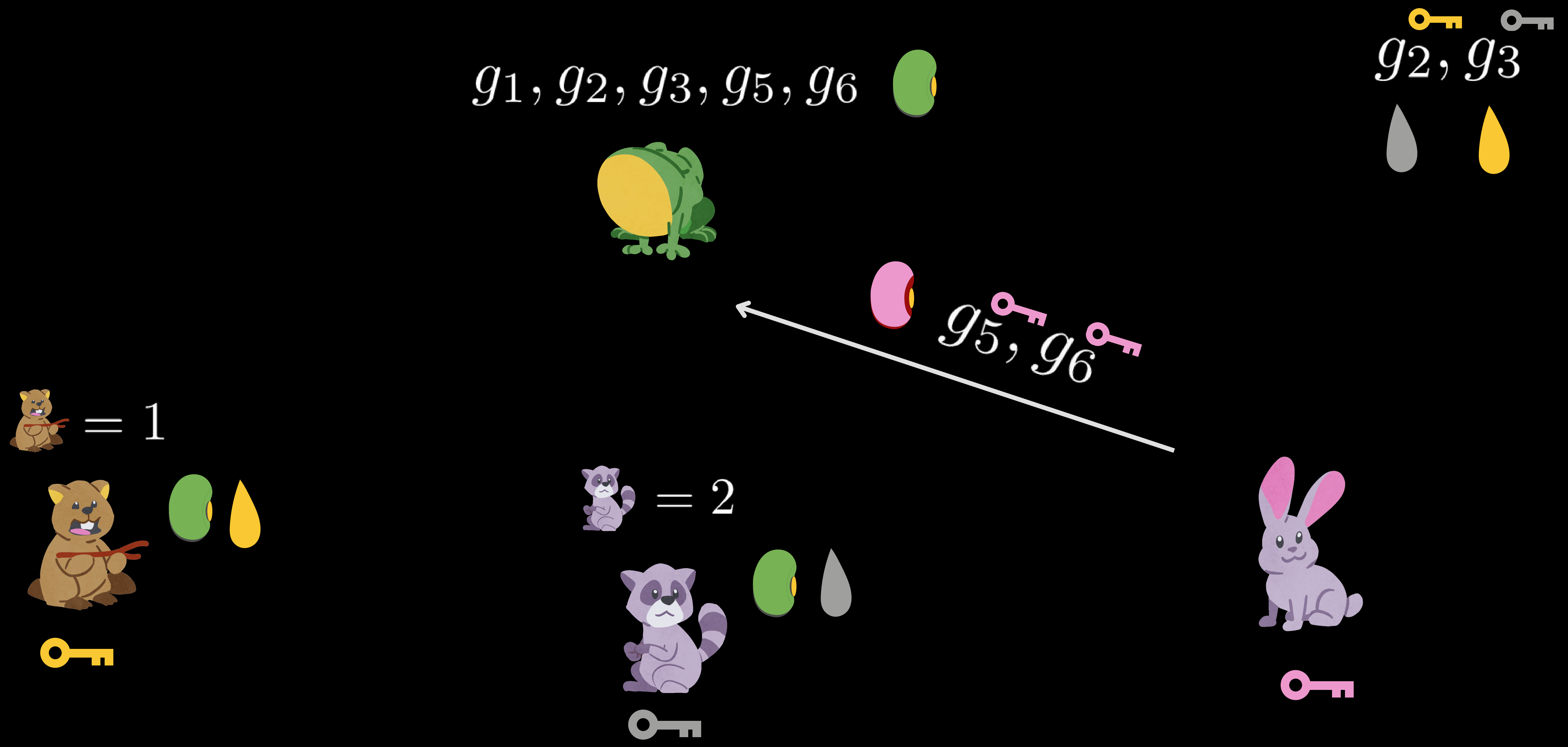
# THE GLAESER-KOLONELOS-MALAVOLTA-RAHIMI (GKMR) RBE, USING LIBERT-YUNG VECTOR COMMITMENTS [TCC:LY10]

## Updates



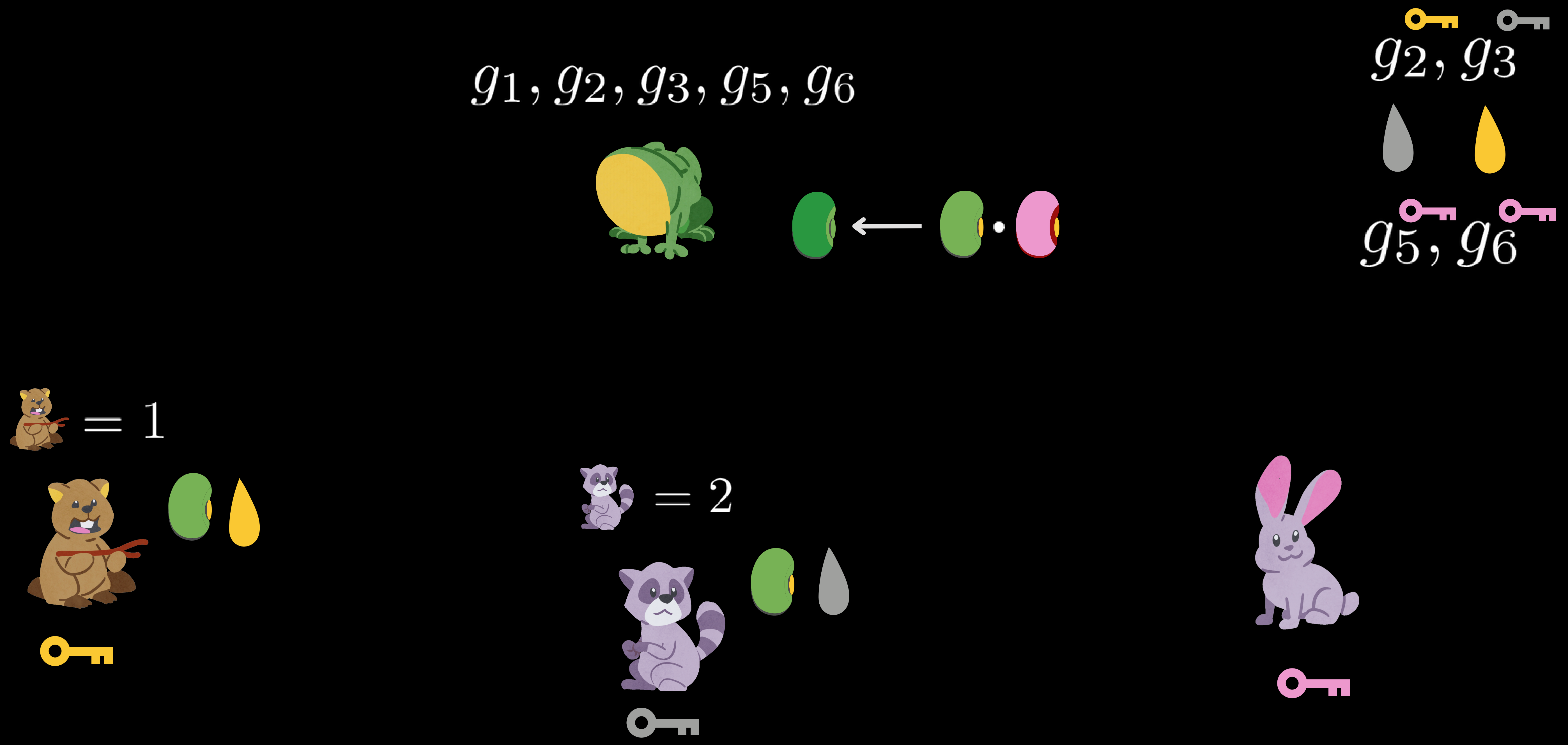
# THE GLAESER-KOLONELOS-MALAVOLTA-RAHIMI (GKMR) RBE, USING LIBERT-YUNG VECTOR COMMITMENTS [TCC:LY10]

**Updates**



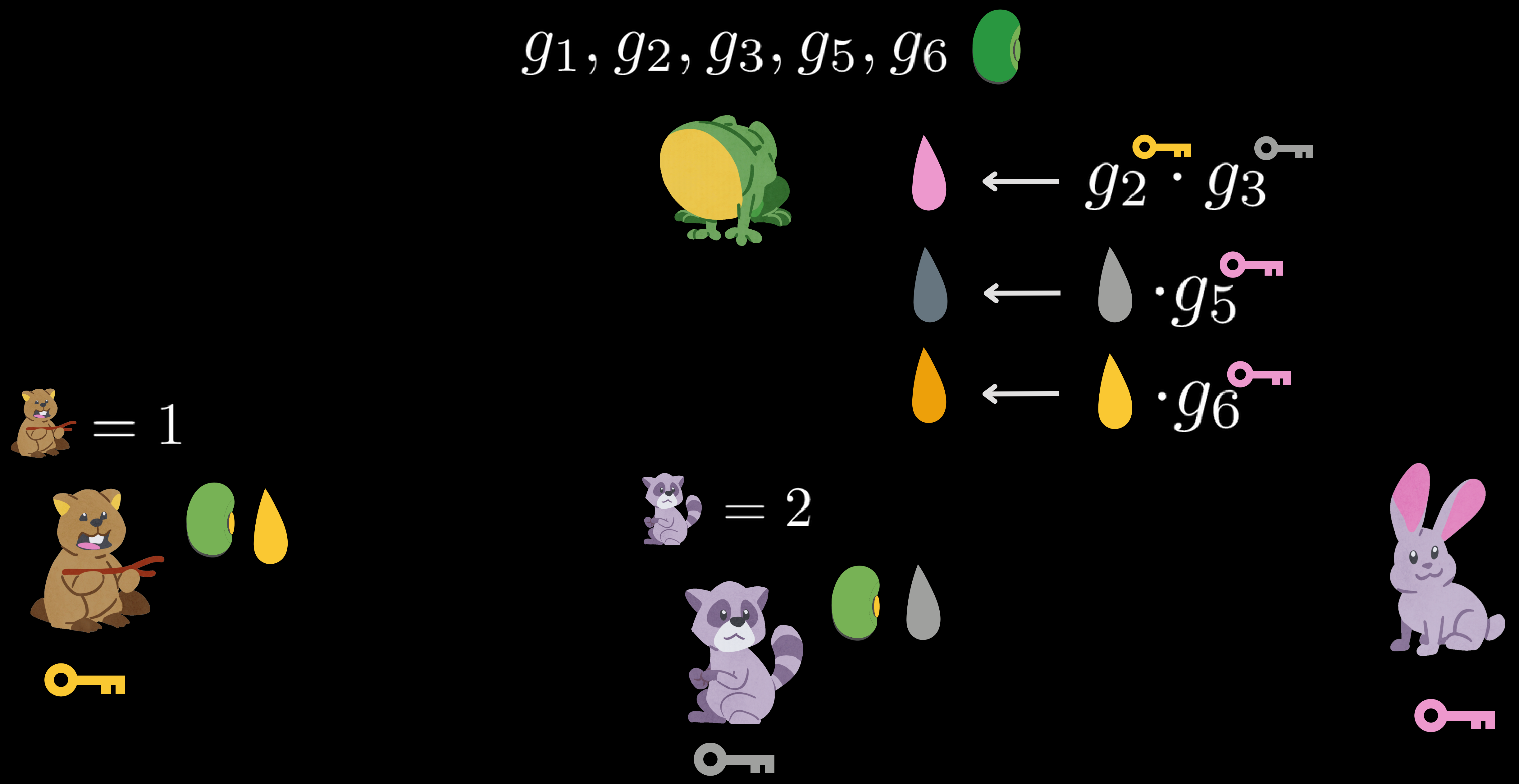
# THE GLAESER-KOLONELOS-MALAVOLTA-RAHIMI (GKMR) RBE, USING LIBERT-YUNG VECTOR COMMITMENTS [TCC:LY10]

**Updates**



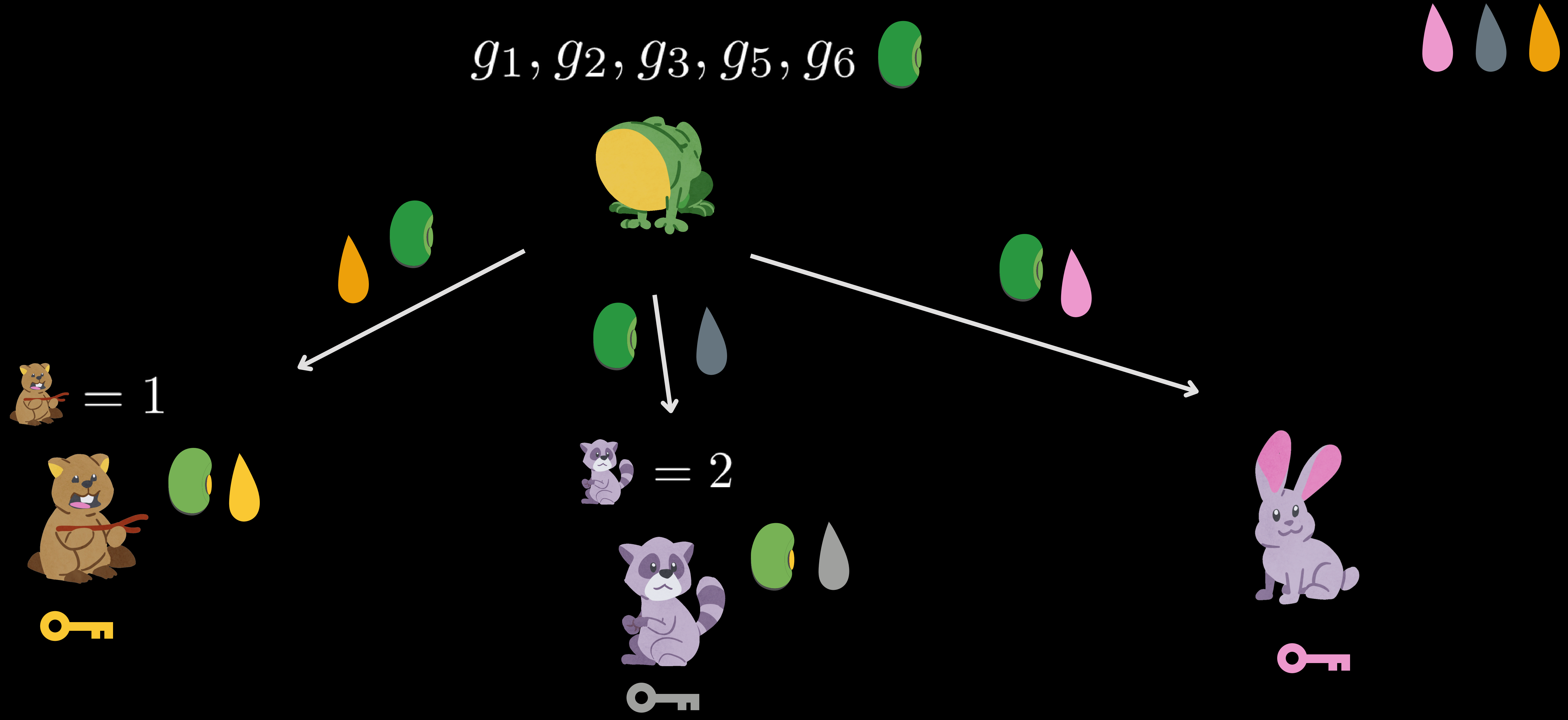
# THE GLAESER-KOLONELOS-MALAVOLTA-RAHIMI (GKMR) RBE, USING LIBERT-YUNG VECTOR COMMITMENTS [TCC:LY10]

**Updates**



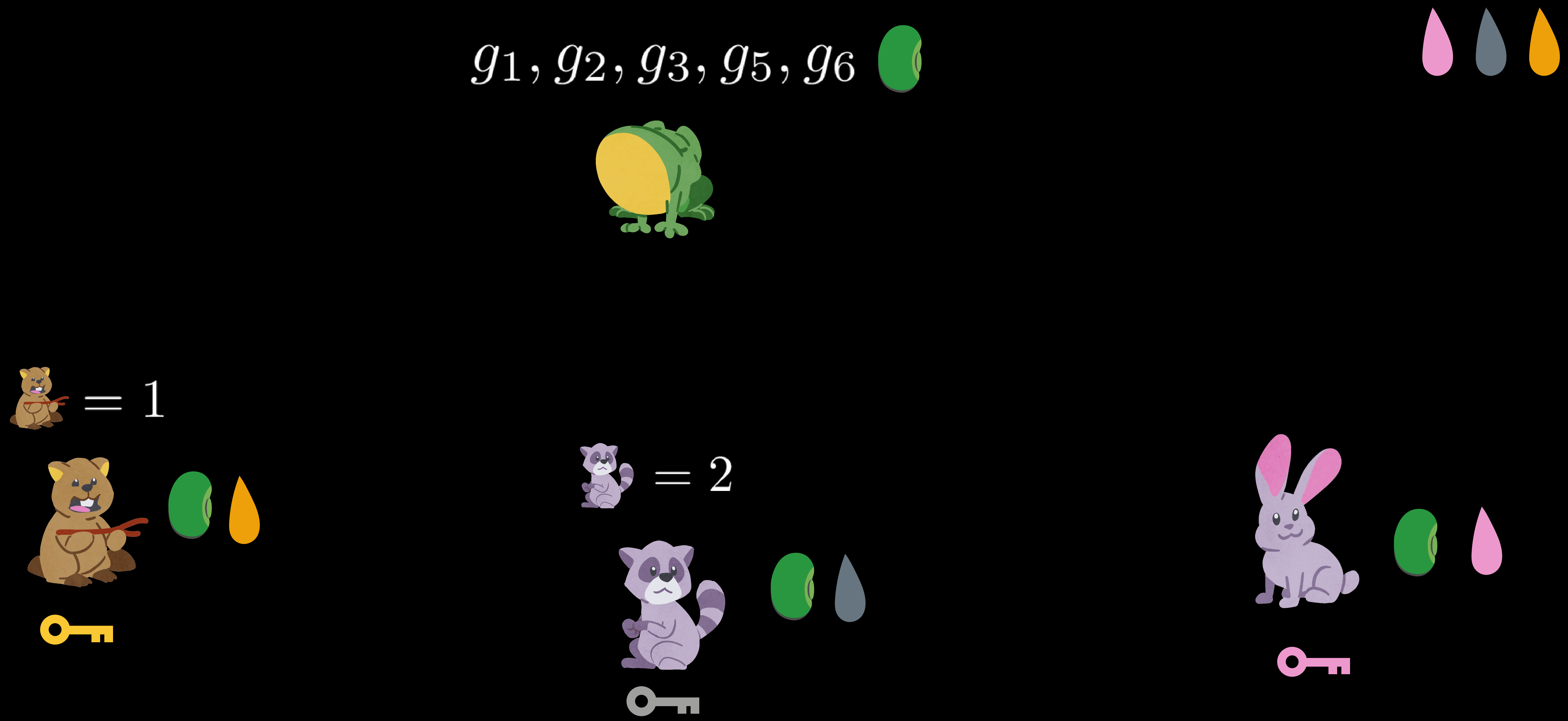
# THE GLAESER-KOLONELOS-MALAVOLTA-RAHIMI (GKMR) RBE, USING LIBERT-YUNG VECTOR COMMITMENTS [TCC:LY10]

**Updates**

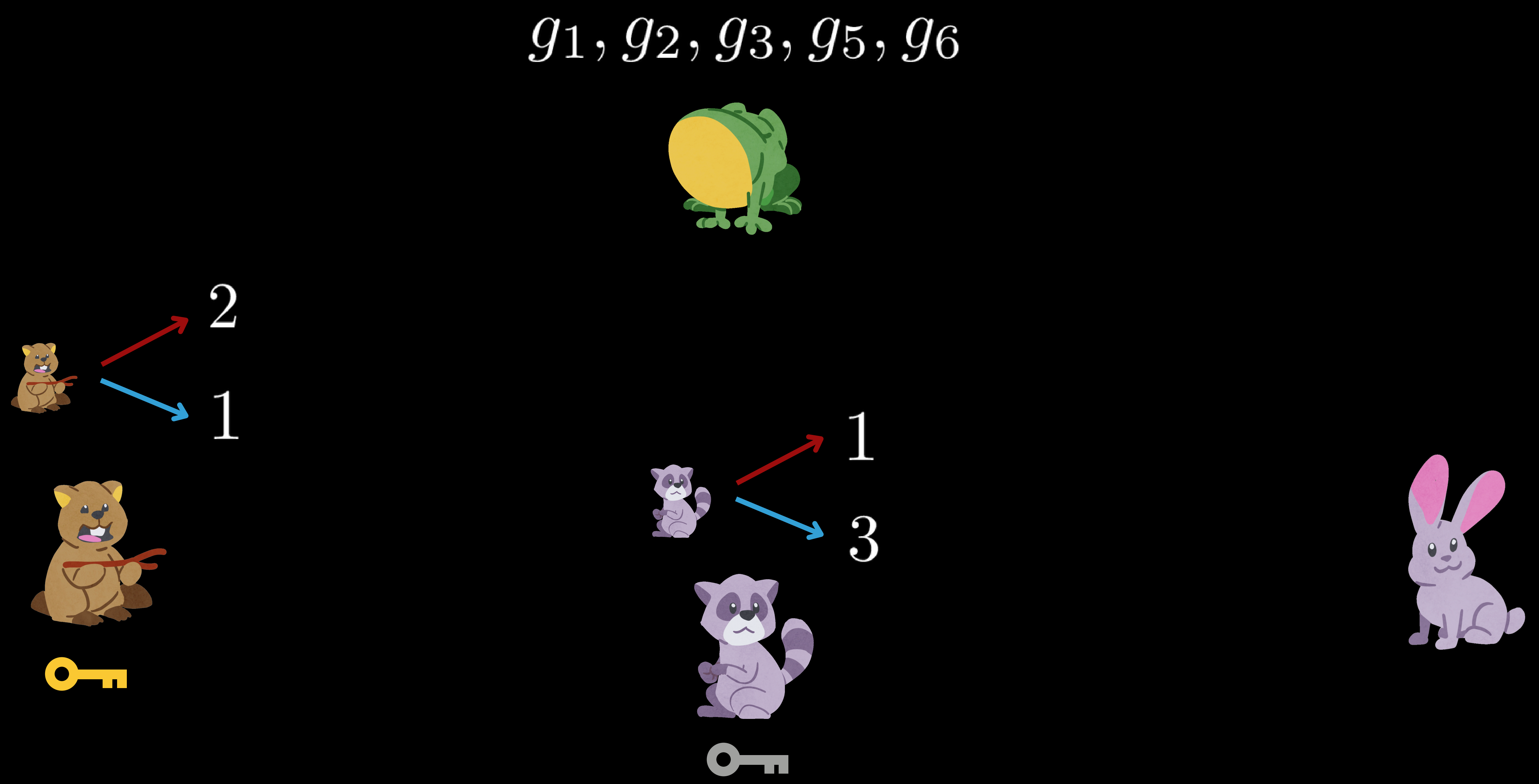


# THE GLAESER-KOLONELOS-MALAVOLTA-RAHIMI (GKMR) RBE, USING LIBERT-YUNG VECTOR COMMITMENTS [TCC:LY10]

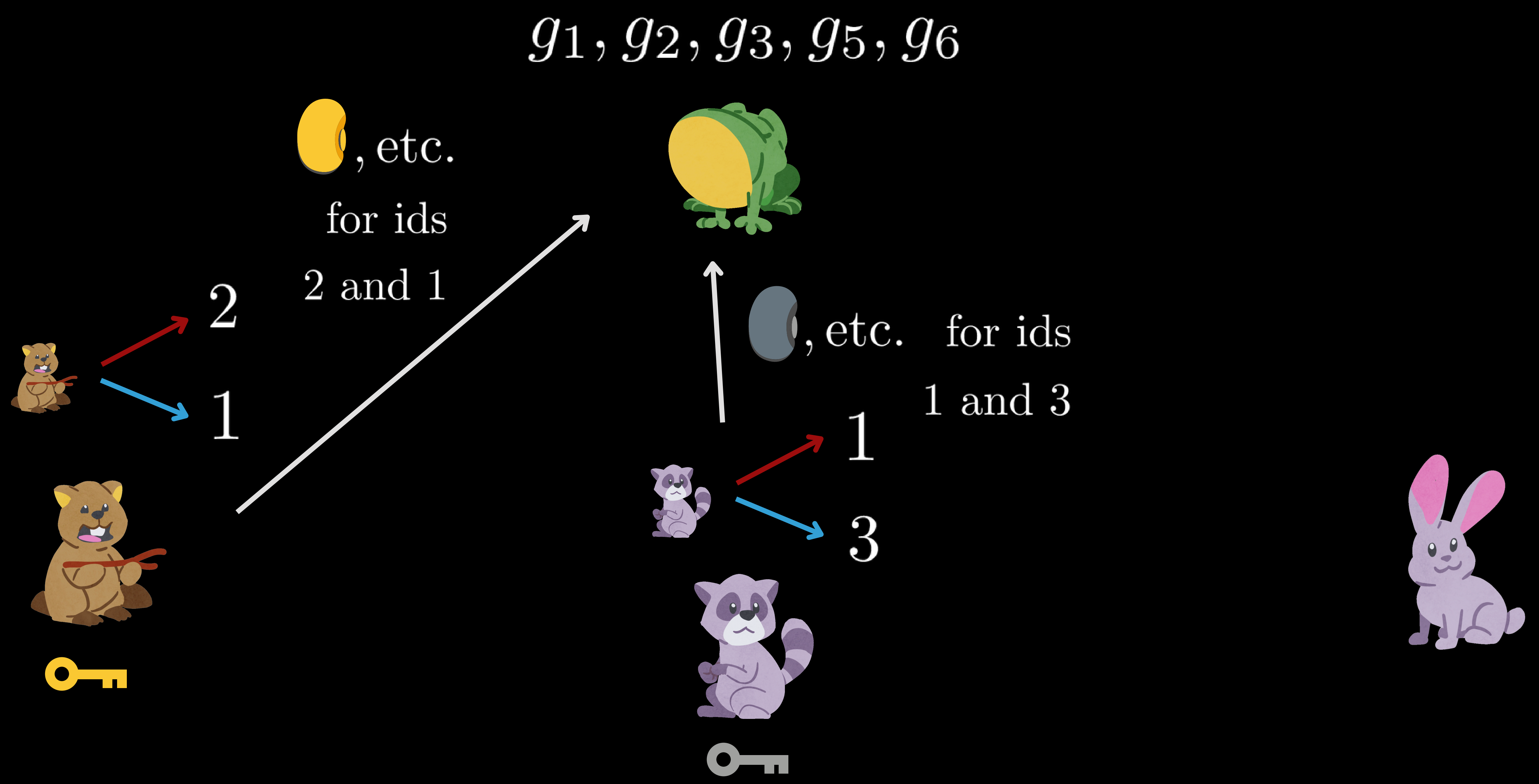
## Updates



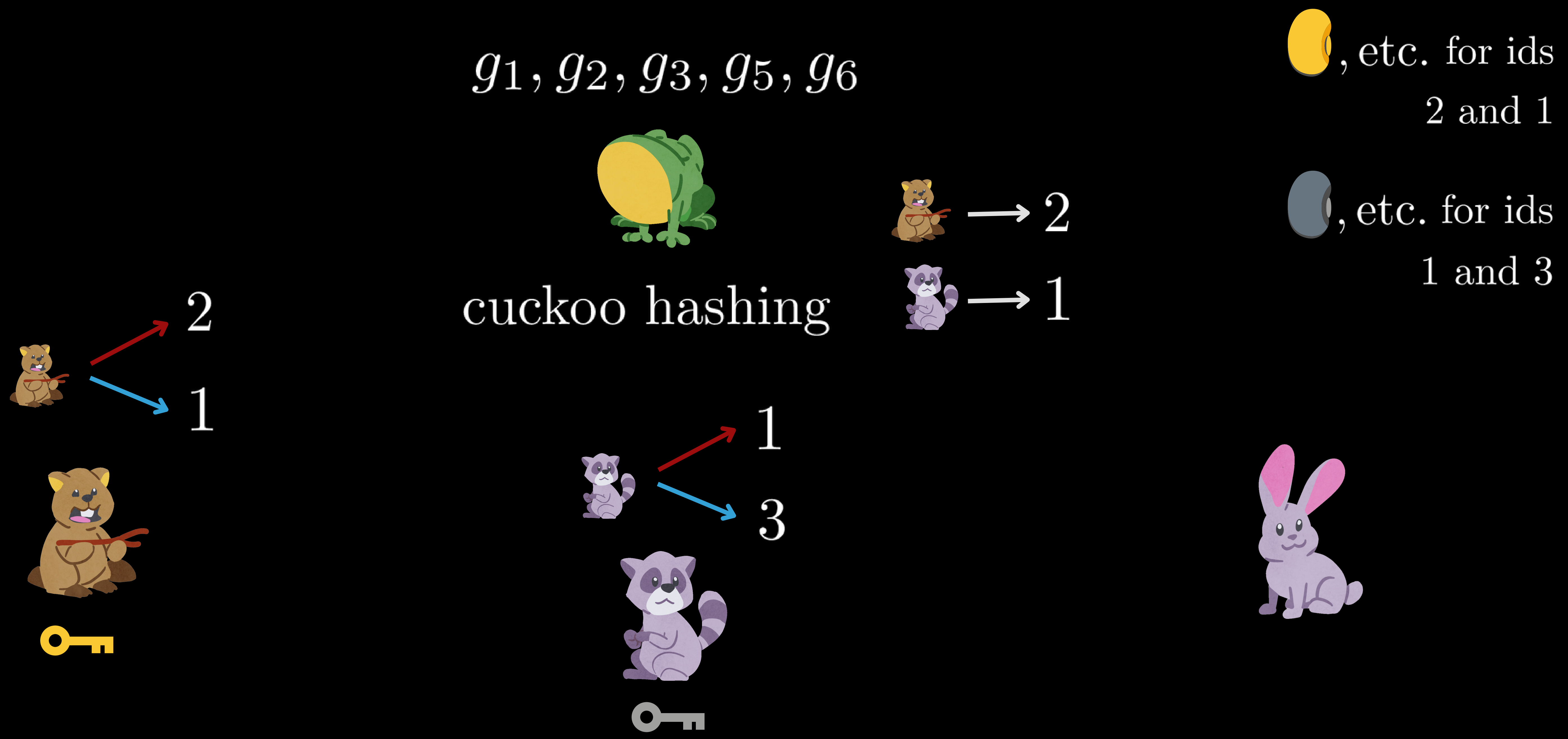
# OUR SCHEME,
# COMBINING GKMR WITH CUCKOO HASHING



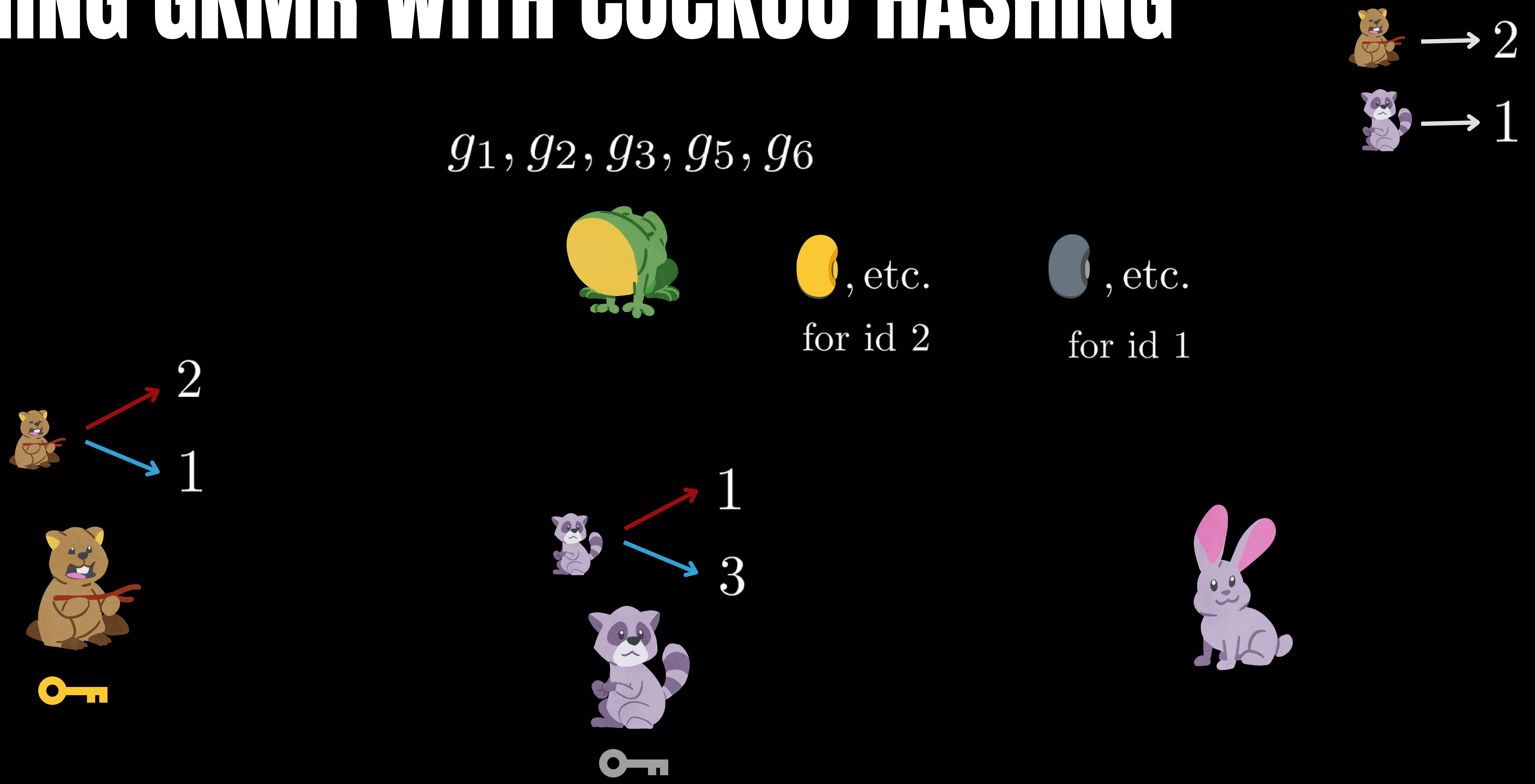
# OUR SCHEME,
# COMBINING GKMR WITH CUCKOO HASHING



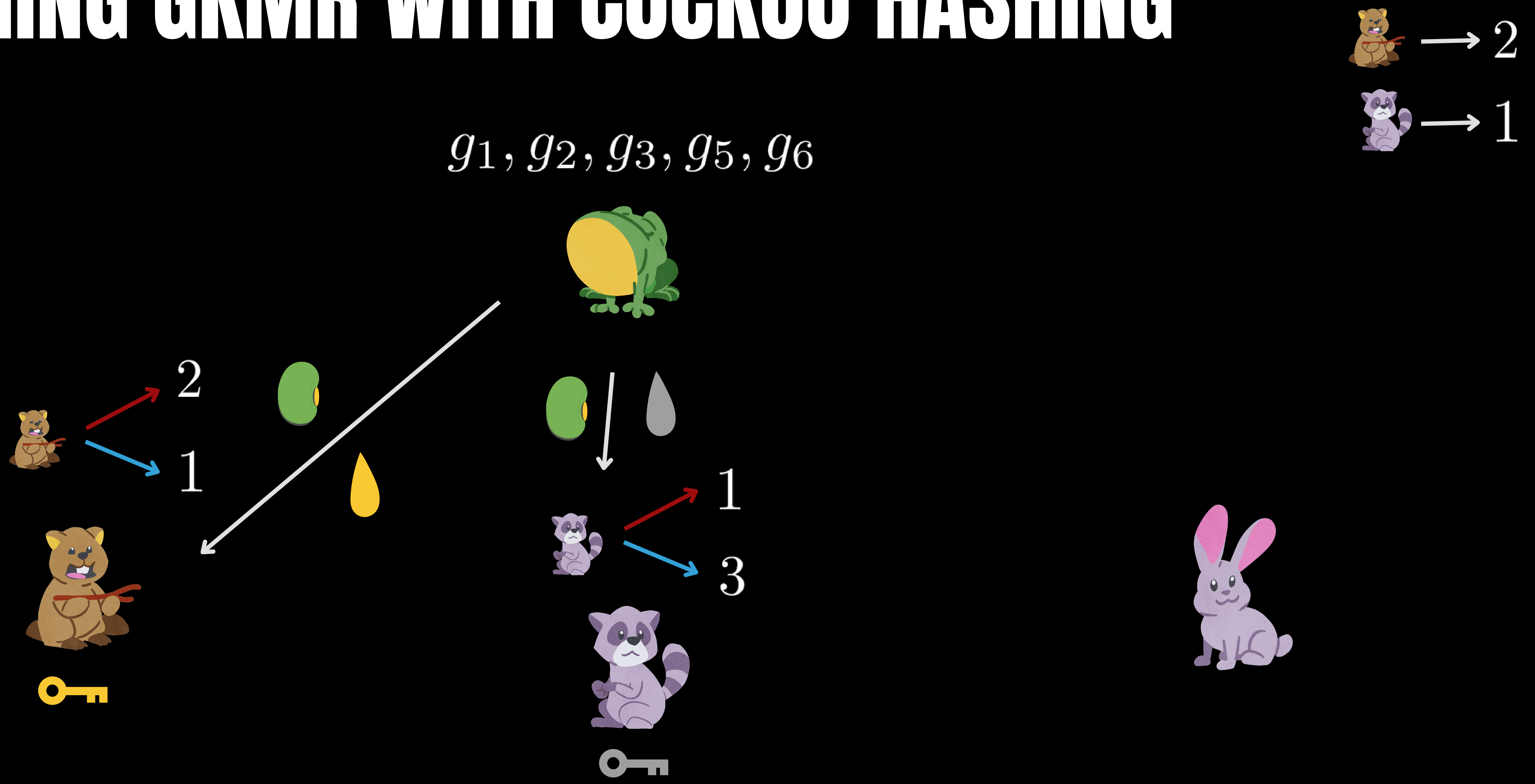
# OUR SCHEME, COMBINING GKMR WITH CUCKOO HASHING



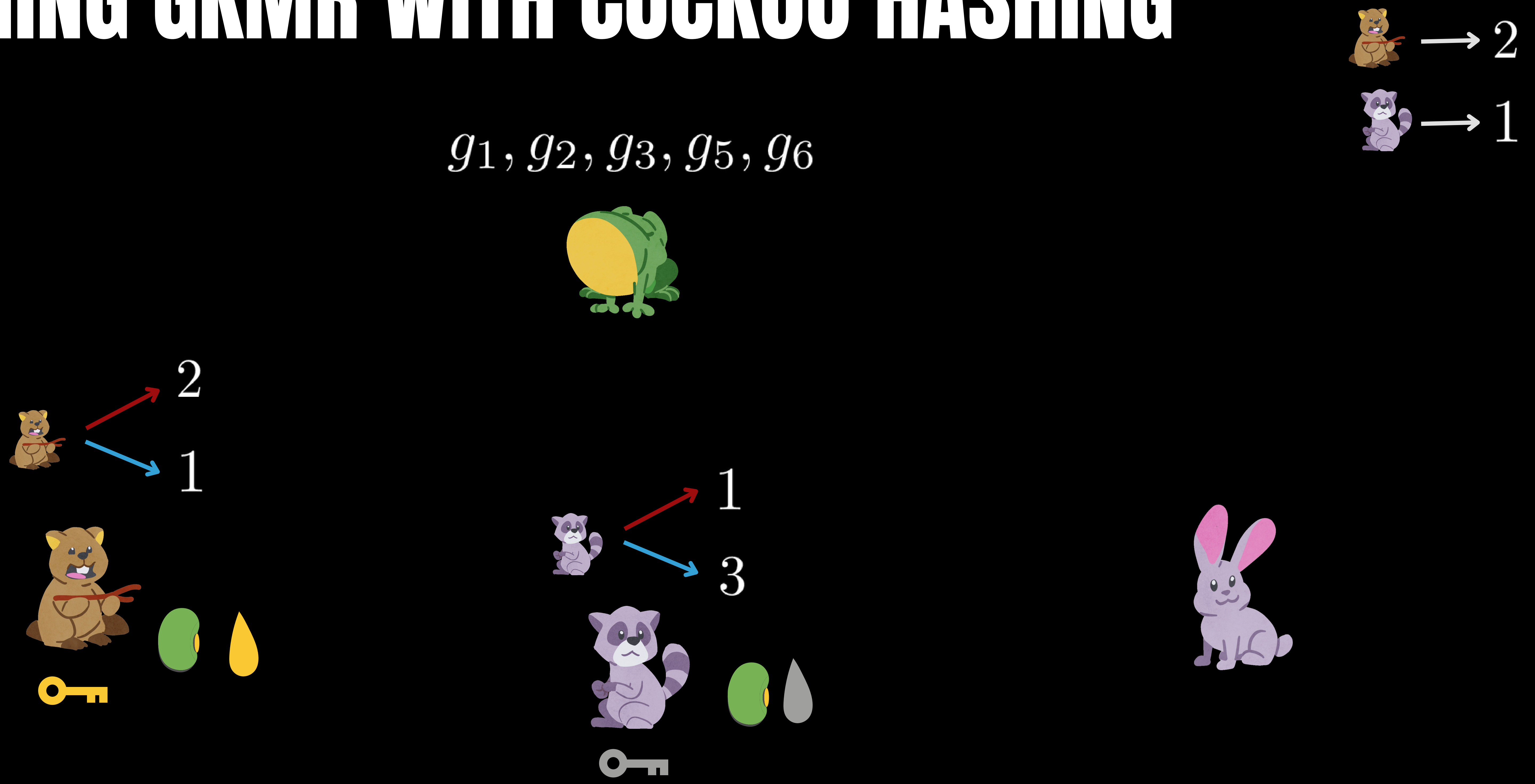
# OUR SCHEME,
# COMBINING GKMR WITH CUCKOO HASHING



# OUR SCHEME,
# COMBINING GKMR WITH CUCKOO HASHING

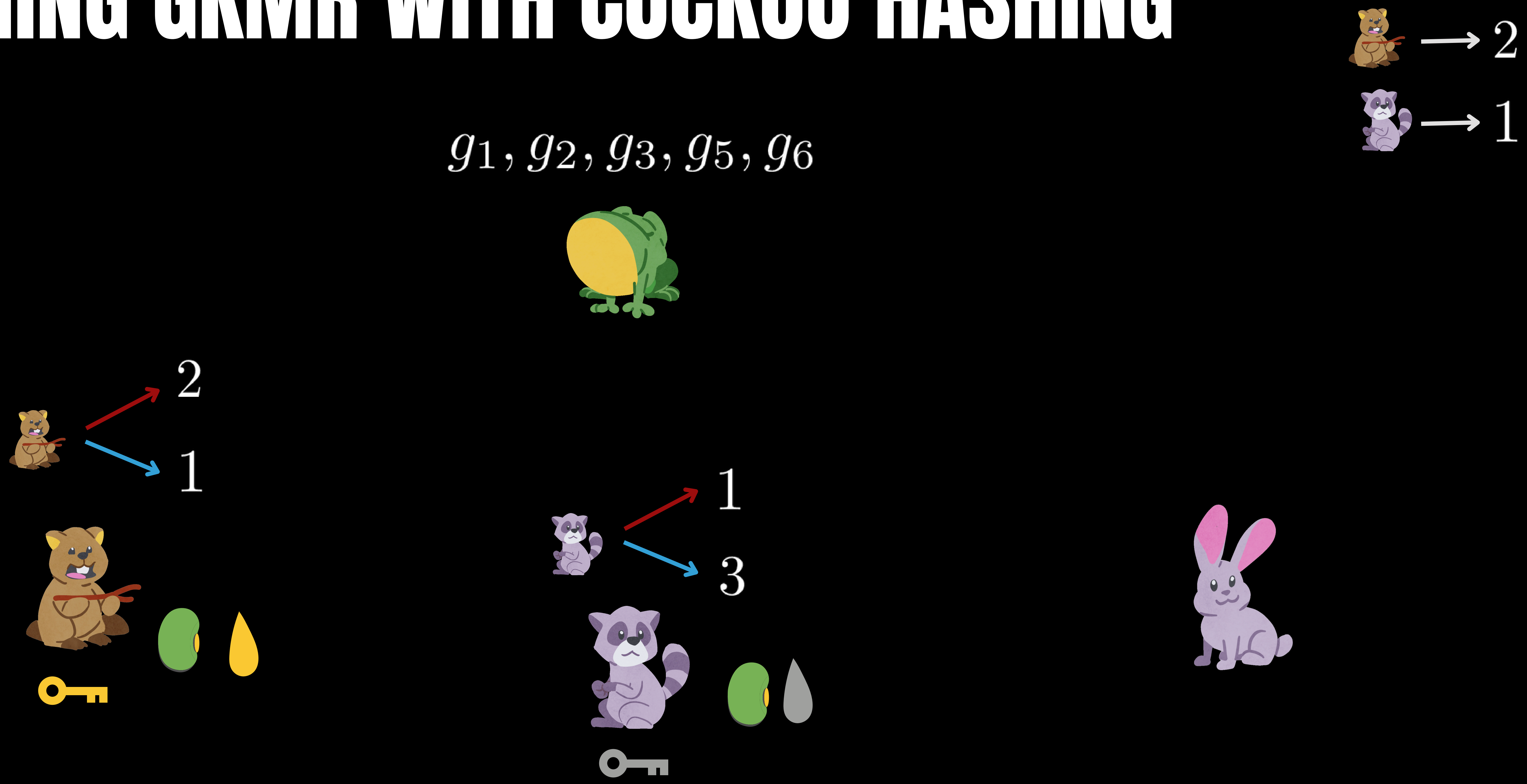


# OUR SCHEME,
# COMBINING GKMR WITH CUCKOO HASHING

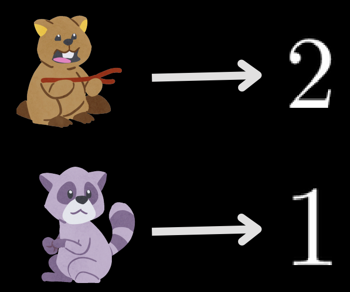


# OUR SCHEME, COMBINING GKMR WITH CUCKOO HASHING

**Problem**

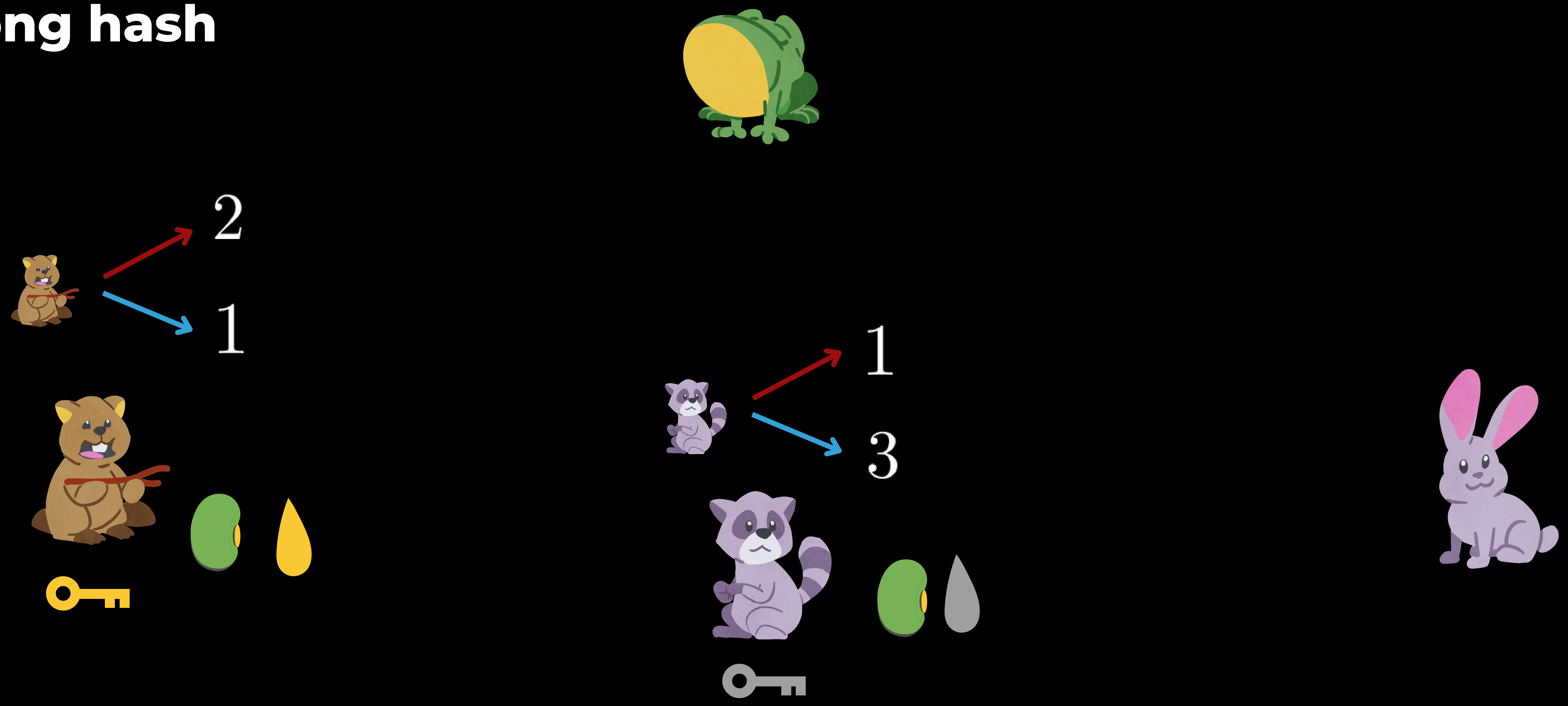


# OUR SCHEME, COMBINING GKMR WITH CUCKOO HASHING

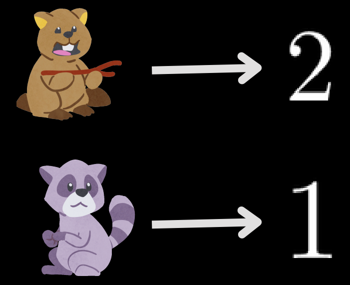



**Problem:**
**what if encryptors use the wrong hash function?**

$g_1, g_2, g_3, g_5, g_6$

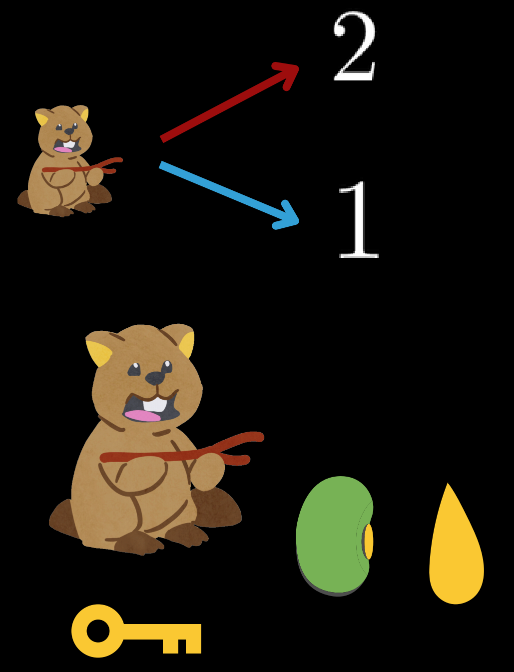


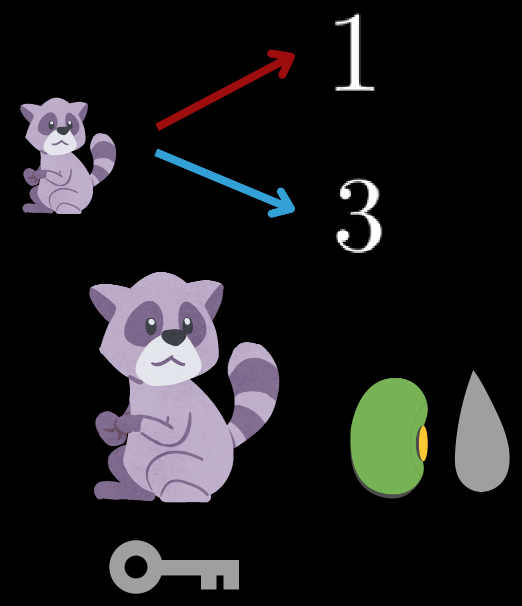
# OUR SCHEME,
# COMBINING GKMR WITH CUCKOO HASHING

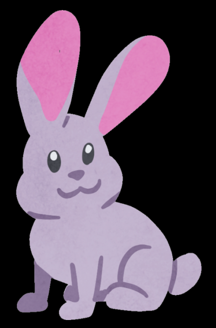



**Encryption needs to be not only with respect to the position, but also the identity.**

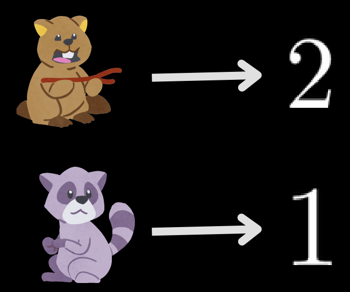
$g_1, g_2, g_3, g_5, g_6$



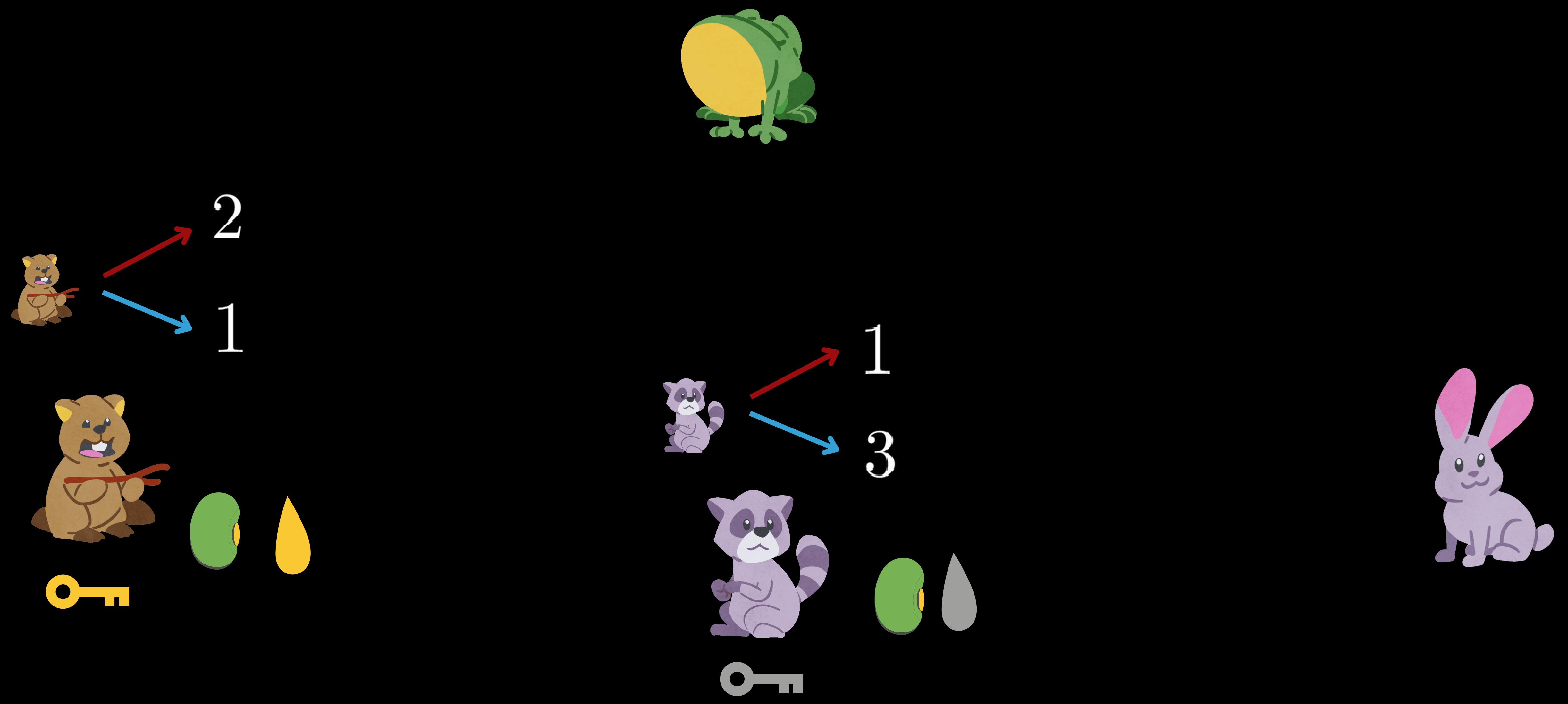



# OUR SCHEME,
# COMBINING GKMR WITH CUCKOO HASHING

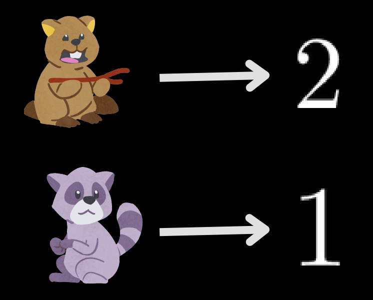


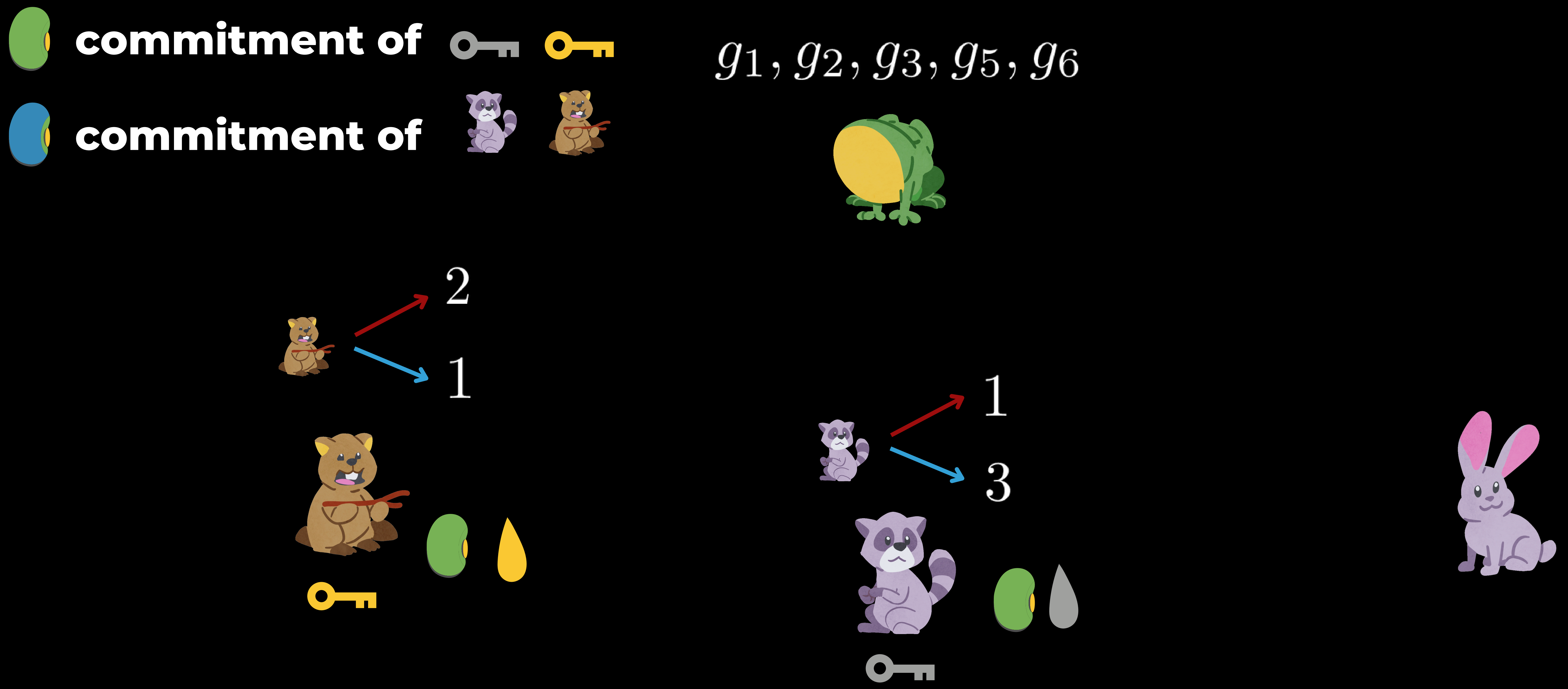

commitment of

$$g_1, g_2, g_3, g_5, g_6$$

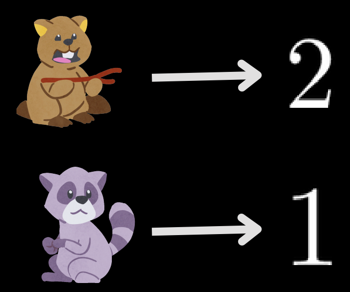


# OUR SCHEME, COMBINING GKMR WITH CUCKOO HASHING



# OUR SCHEME,
# COMBINING GKMR WITH CUCKOO HASHING



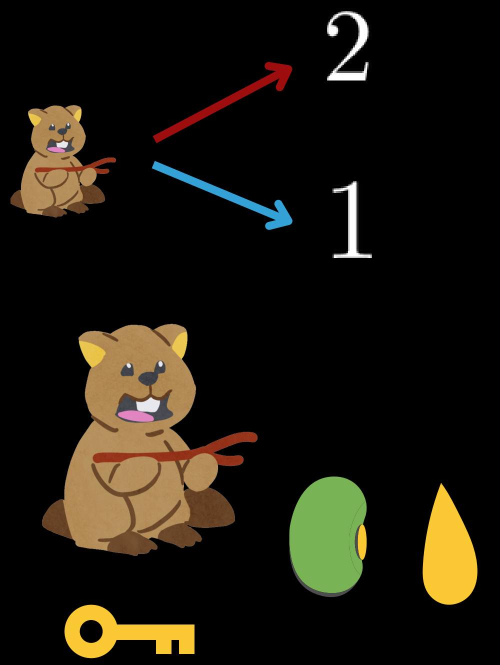

commitment of

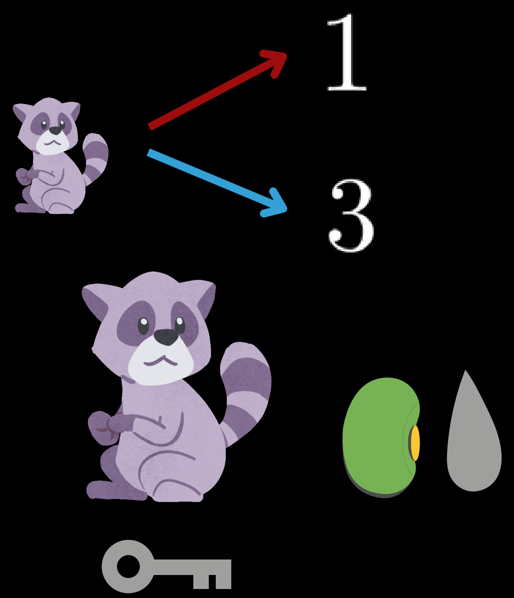
commitment of

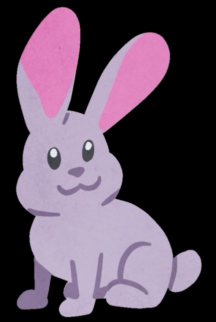
opening of in $g_2$

opening of in $g_2$

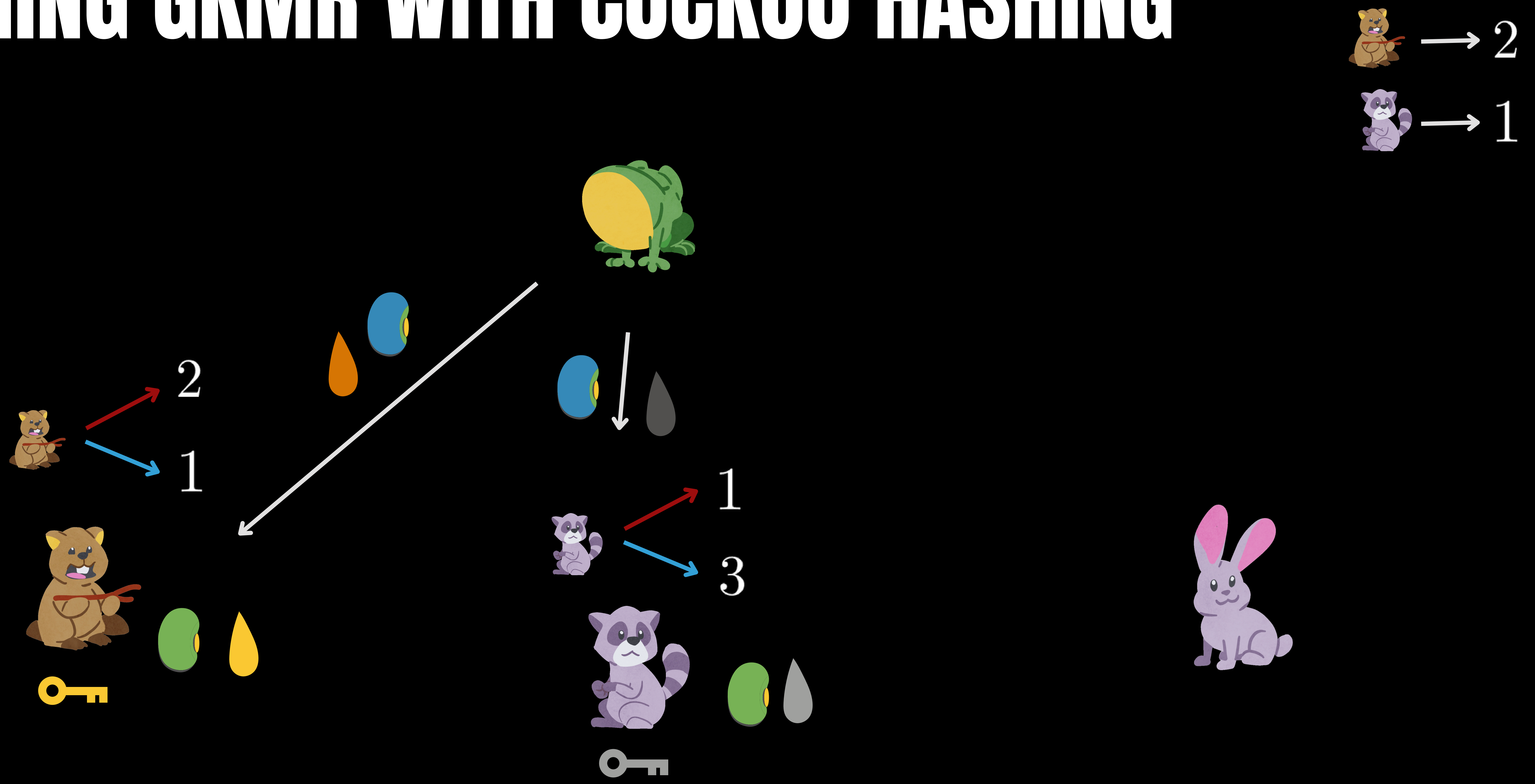
(in position 2)



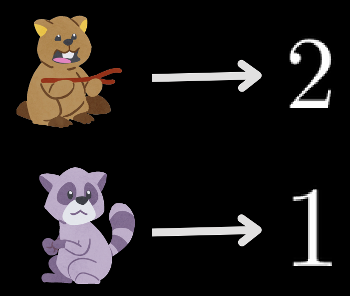



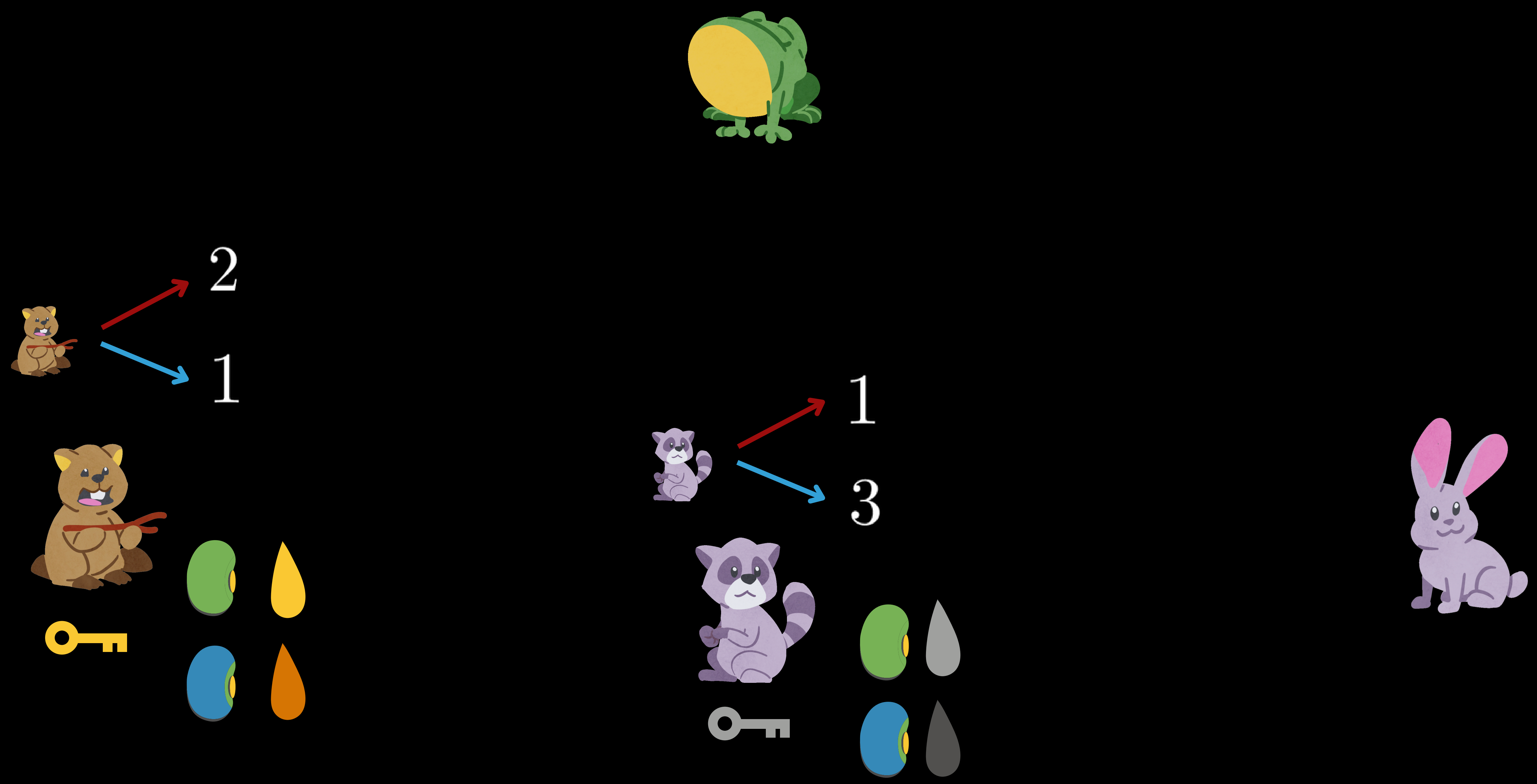



# OUR SCHEME,
# COMBINING GKMR WITH CUCKOO HASHING



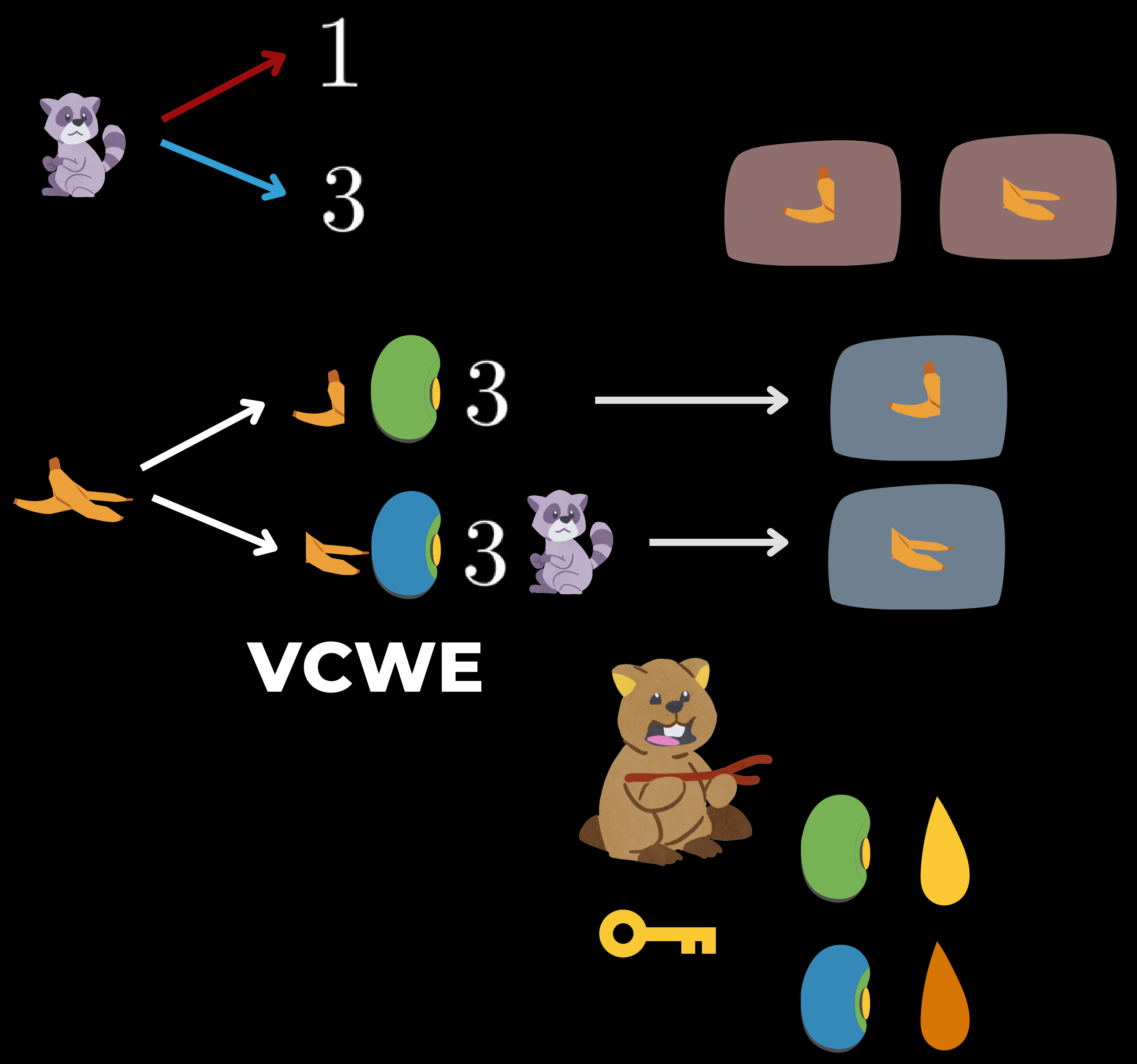
# OUR SCHEME,
# COMBINING GKMR WITH CUCKOO HASHING

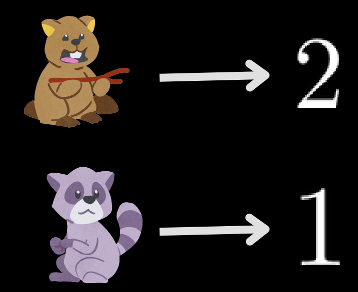


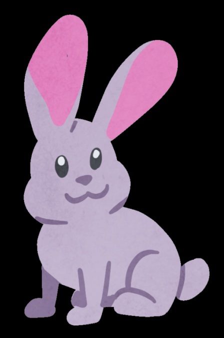



# OUR SCHEME,
# COMBINING GKMR WITH CUCKOO HASHING

# OUR SCHEME,
# COMBINING GKMR WITH CUCKOO HASHING

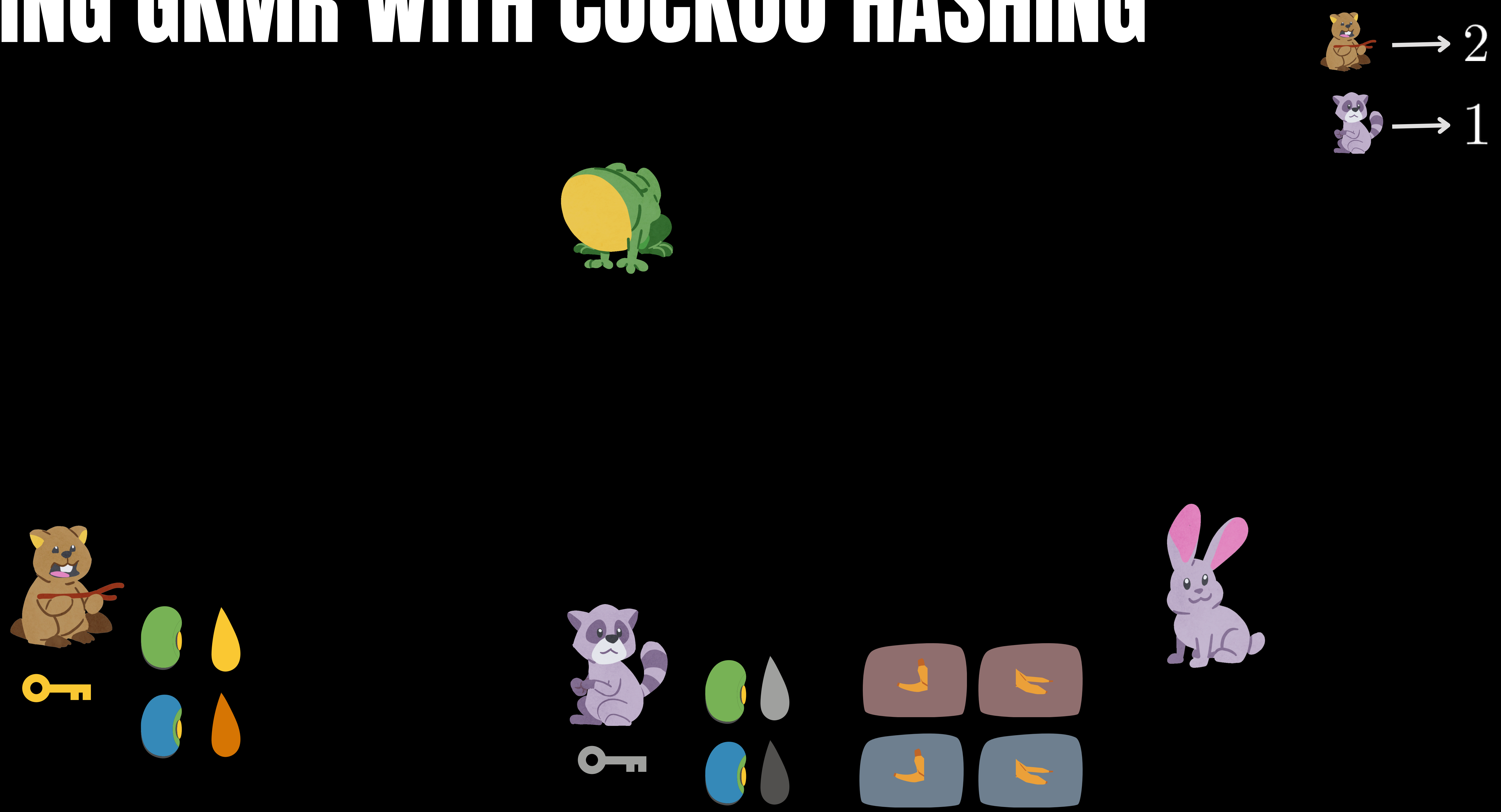


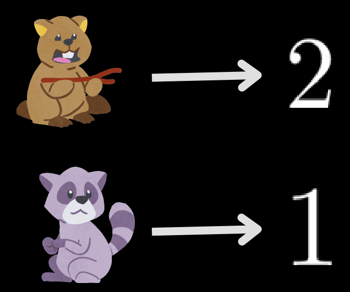



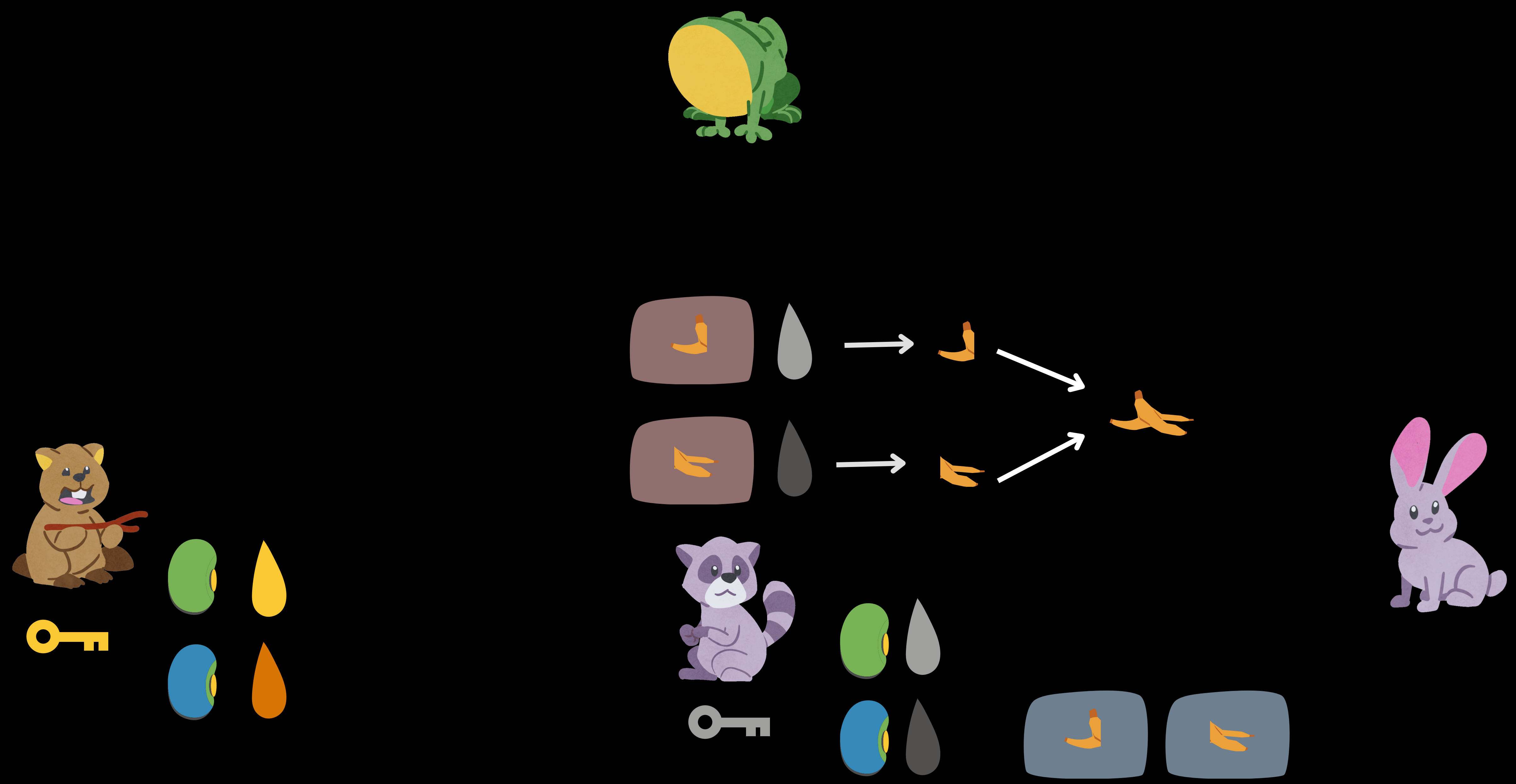
# OUR SCHEME,
# COMBINING GKMR WITH CUCKOO HASHING

# OUR SCHEME,
# COMBINING GKMR WITH CUCKOO HASHING

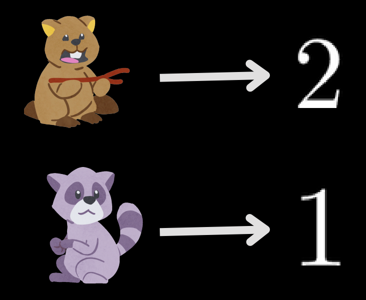


# OUR SCHEME,
# COMBINING GKMR WITH CUCKOO HASHING

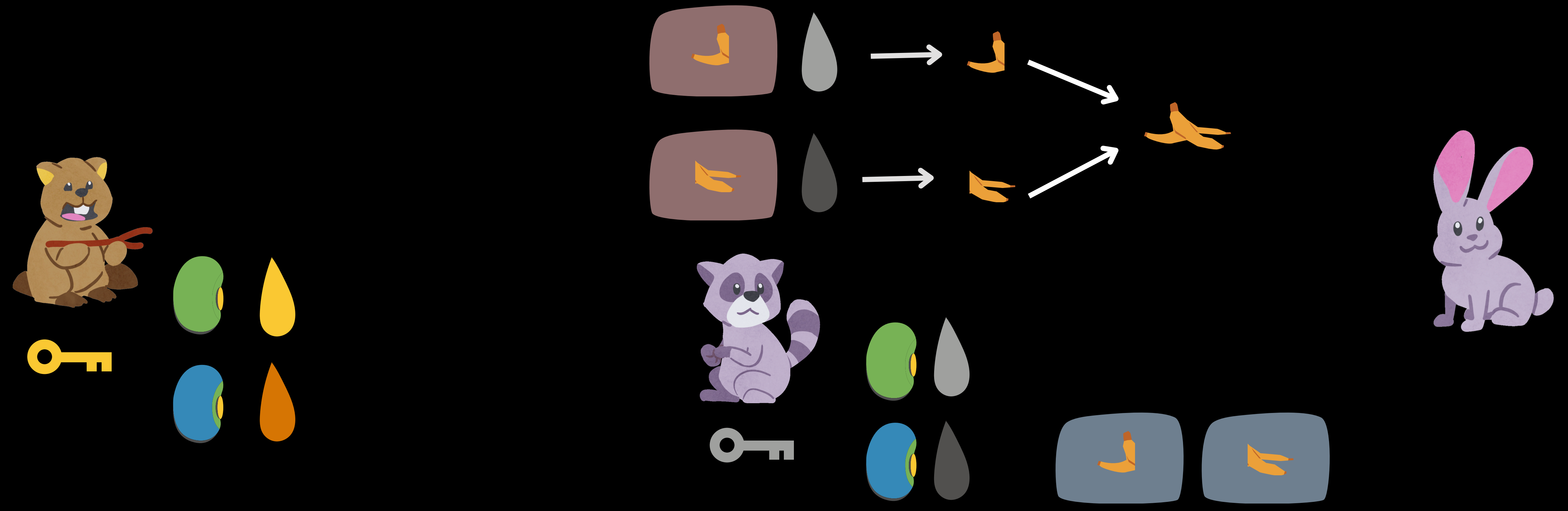


# OUR SCHEME,
# COMBINING GKMR WITH CUCKOO HASHING

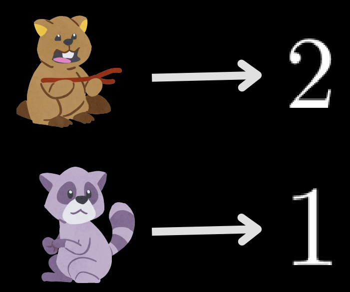


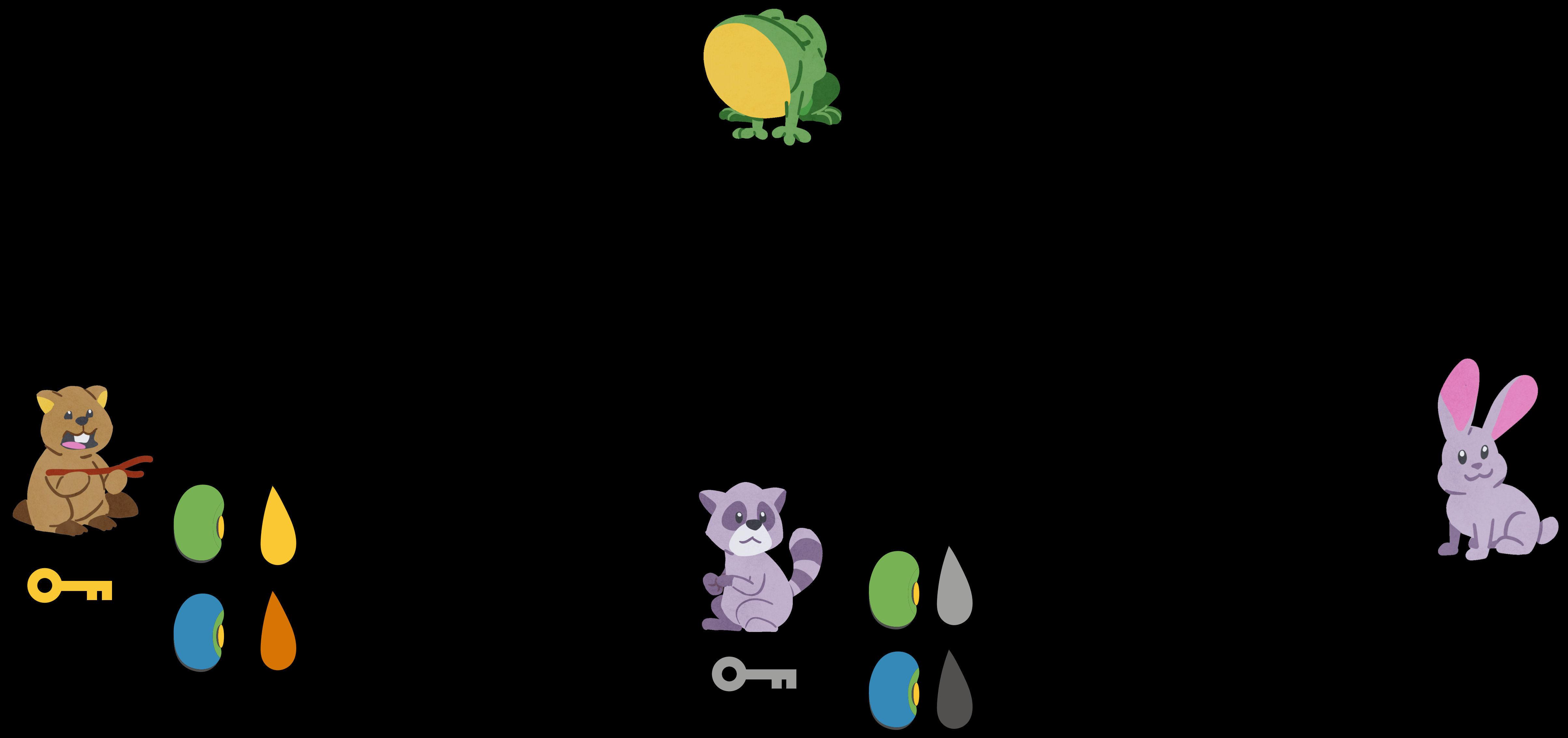

**Without both openings for position 1,**
**without committed to in in position 1,**
**without made for this cuckoo hashing,**
**nothing could be inferred.**



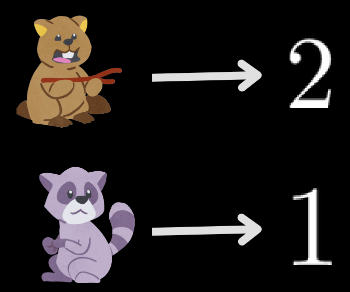
# OUR SCHEME,
# COMBINING GKMR WITH CUCKOO HASHING

**updatable as before**



# OUR SCHEME,
# COMBINING GKMR WITH CUCKOO HASHING




**updatable as before**
**if the cuckoo hashing changes,**
**the commitments and opening change.**

# OUR SCHEME, COMBINING GKMR WITH CUCKOO HASHING

| | Setting | $\mathcal{ID}$ | Compactness | $\lvert\mathsf{ct}\rvert$ | #updates | $\lvert\mathsf{pp}\rvert + \lvert\mathsf{crs}\rvert$ |
|---|---|---|---|---|---|---|
| [HLWW23] | Pairings (C) | $\{0,1\}^*$ | Adaptive | $O(\lambda \log n)$ | $\log n$ | $O(\lambda n^{2/3} \log n)$ |
| [GKMR22] | Pairings (P) | $[1,n]$ | Adaptive | $4 \log n$ | $\log n$ | $O(\sqrt{n} \log n)$ |
| Ours P1 | Pairings (P) | $\{0,1\}^*$ | Adaptive | $6\lambda \log n$ | $\log n$ | $O(\sqrt{\lambda n} \log n)$ |
| Ours P2 | Pairings (P) | $\{0,1\}^*$ | Selective | $12 \log n$ | $\log n$ | $O(\sqrt{n} \log n)$ |
| [DKL+23] | Lattices | $\{0,1\}^*$ | Adaptive | $(2\lambda + 1) \log n$ | $\log n$ | $O(\log n)$ |
| Ours L | Lattices | $\{0,1\}^*$ | Selective | $4 \log^2 n$ | $\log n$ | $O(\log n)$ |

Table 1: Comparison of the schemes resulting from different instantiations of our compiler. $n$ is the maximum number of users to be registered. Parings (P) indicates prime order groups and Pairings (C) composite order groups respectively. $\lvert\mathsf{ct}\rvert$ in the pairing construction is measured in group elements and in the Lattice constructions LWE ciphertexts.

# OTHER CONTRIBUTION

**KEY-VALUE MAP COMMITMENTS**

**for large keys, with updates, using pairings**

# OTHER CONTRIBUTION

**KEY-VALUE MAP COMMITMENTS**
**for large keys, with updates, using pairings**

**equivalence of vector commitments**
**and universal accumulators**

# THANK YOU

WORK BY DARIO FIORE [1], DIMITRIS KOLONELOS [1, 2] & PAOLA DE PERTHUIS [3,4]

FULL VERSION: HTTPS://EPRINT.IACR.ORG/2023/1389
PUBLISHED AT ASIACRYPT 2023

PRESENTATION BY PAOLA DE PERTHUIS

1: institute iMdea software
2: UNIVERSIDAD POLITÉCNICA MADRID
3: cosmian
4: ENS ÉCOLE NORMALE SUPÉRIEURE