Semantically Secure Order-Revealing Encryption:

Multi-Input Functional Encryption Without Obfuscation

Joe Zimmerman

Joint work with Dan Boneh, Kevin Lewi,
Mariana Raykova, Amit Sahai, and Mark Zhandry

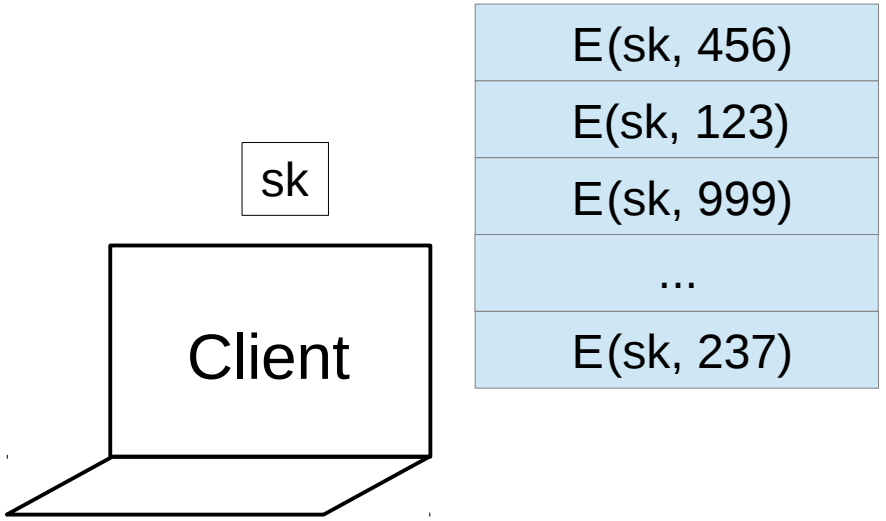
Outline

1. **Motivating example: O*rder-Revealing Encryption***

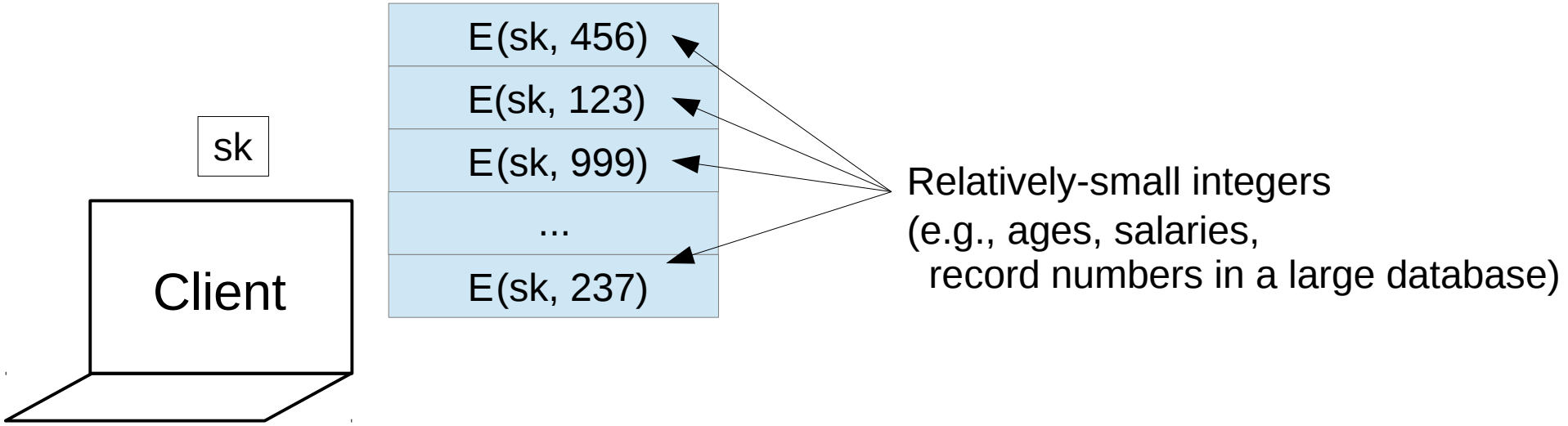
2. Our results
   - Order-Revealing Encryption
   - Generalization: Multi-Input Functional Encryption
     - directly from multilinear maps (no obfuscation)
     - security in the generic model

3. Overview of techniques

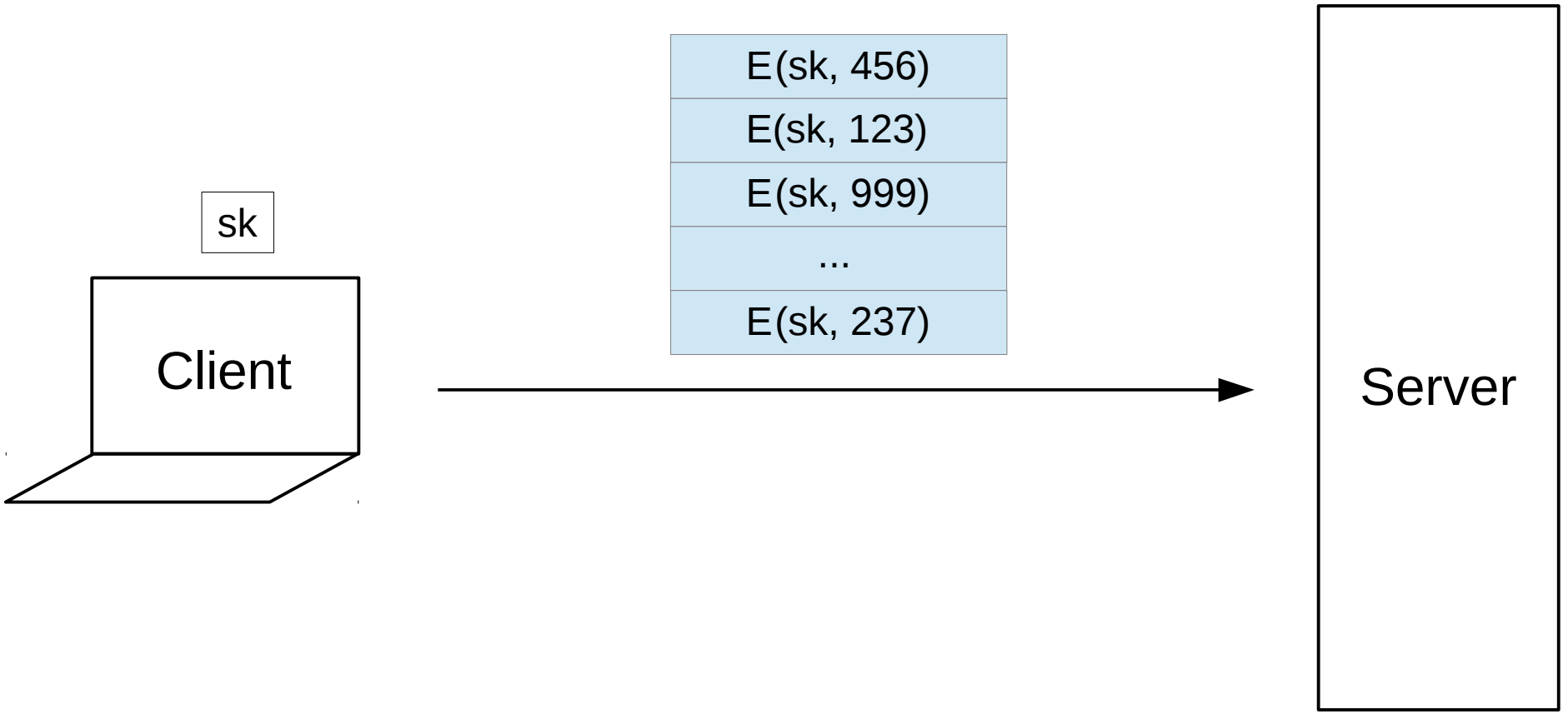
Order-Revealing Encryption



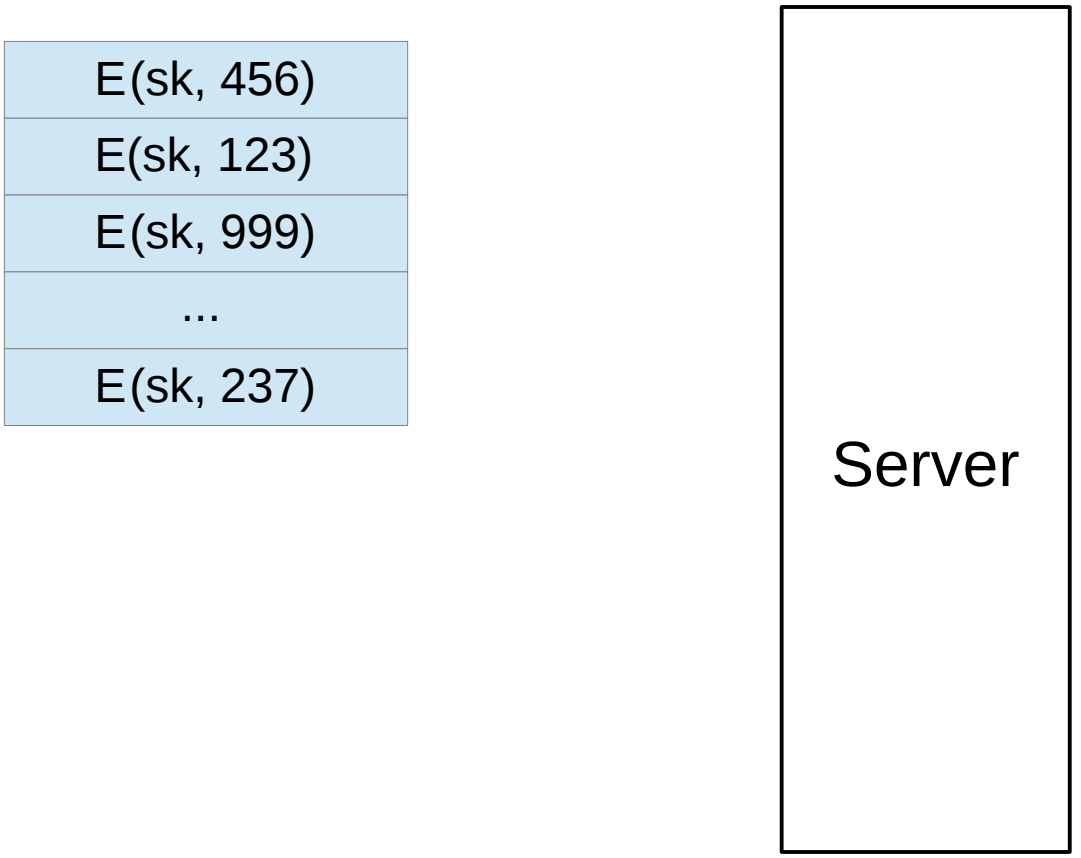
Order-Revealing Encryption



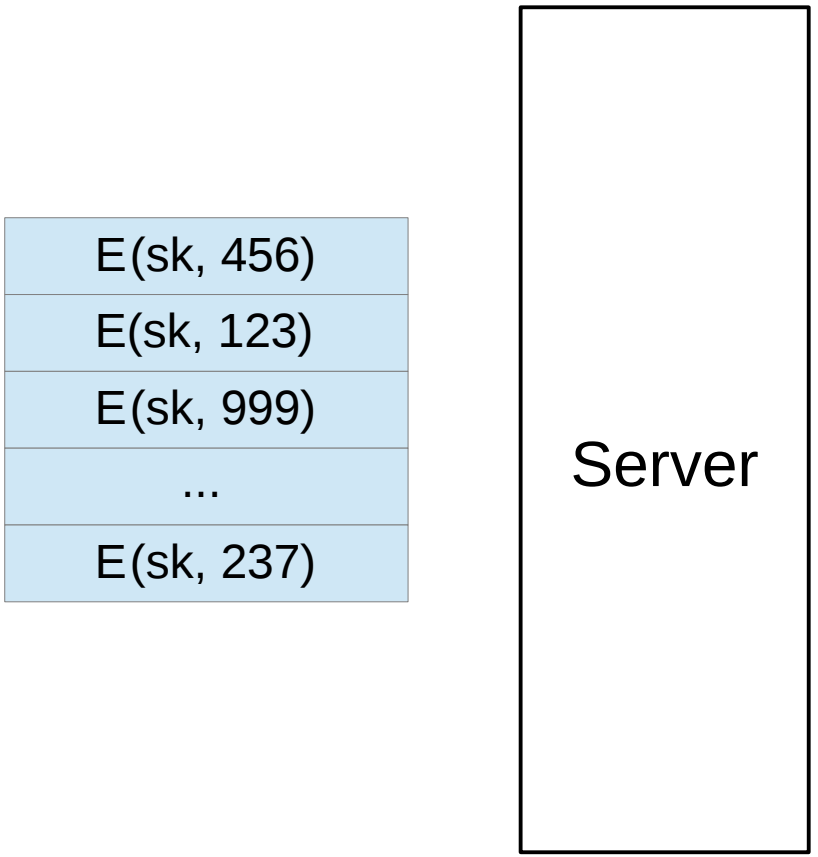
Order-Revealing Encryption



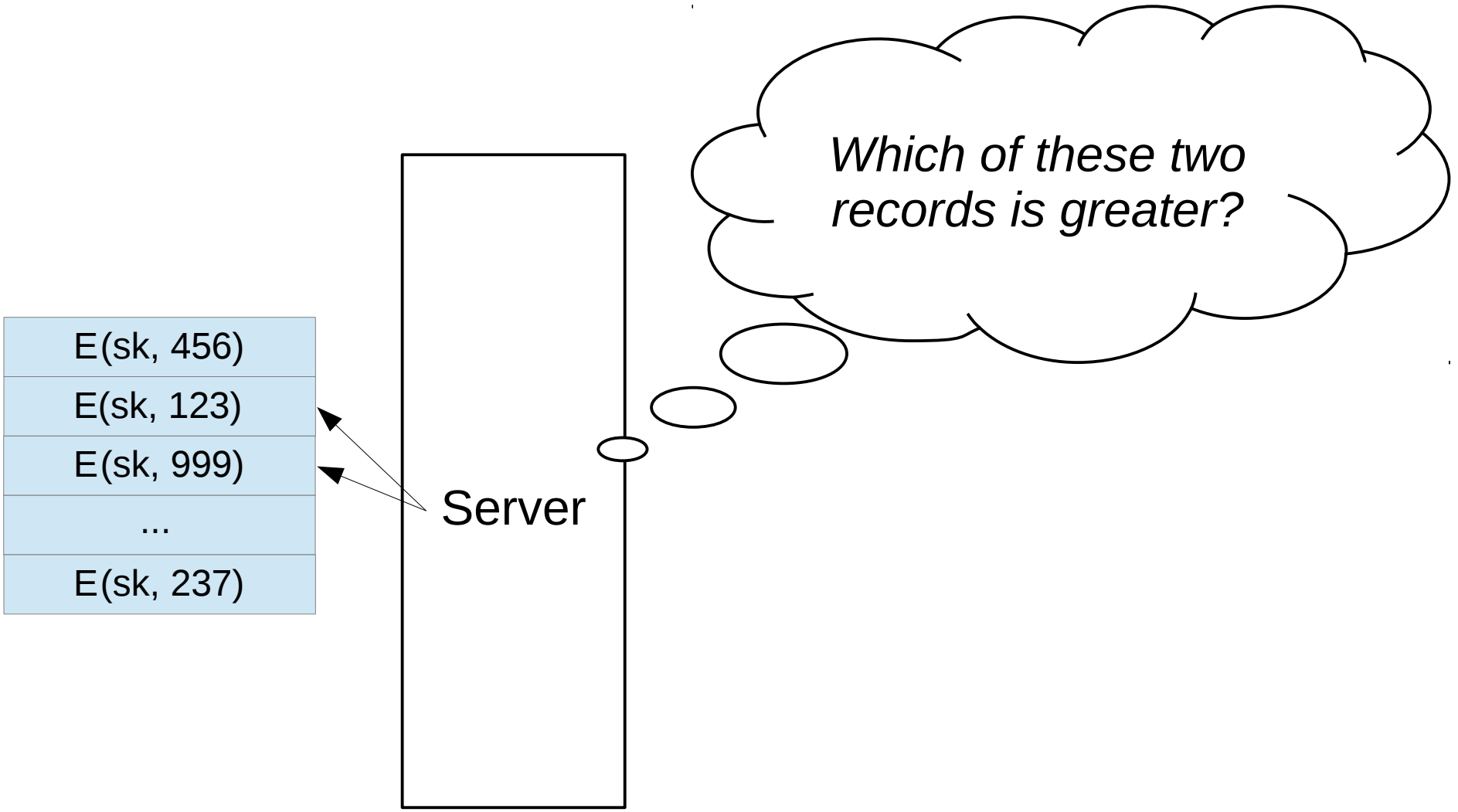
Order-Revealing Encryption



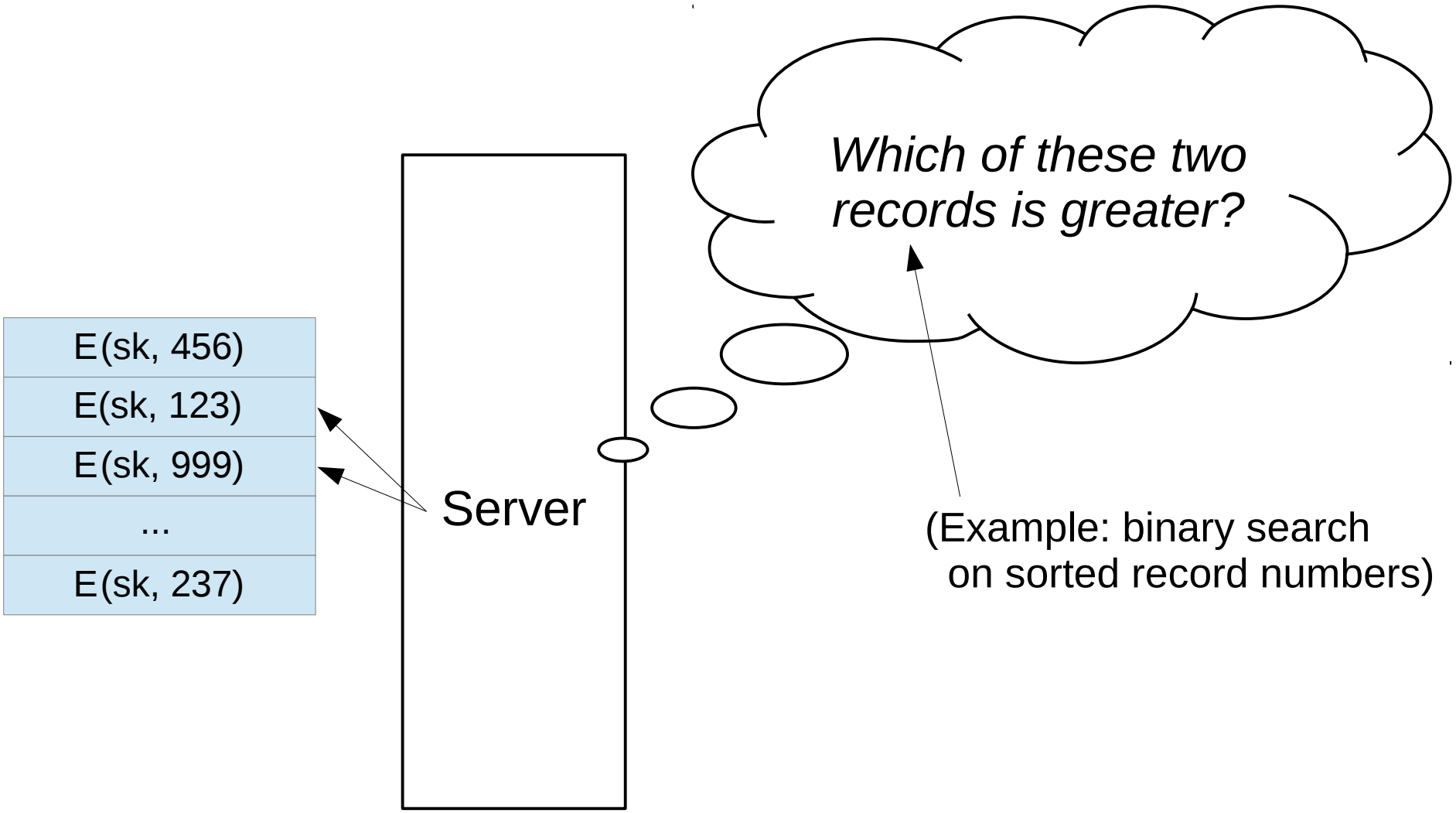
Order-Revealing Encryption



Order-Revealing Encryption



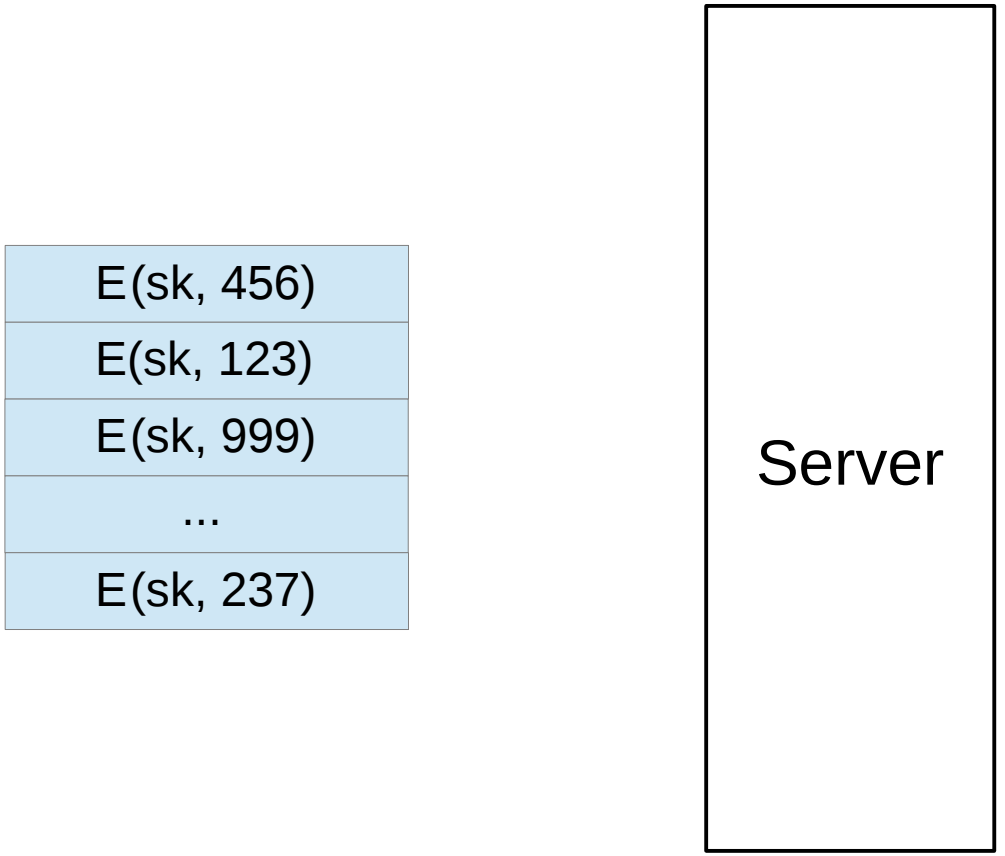
Order-Revealing Encryption



Order-Revealing Encryption

Our goal:

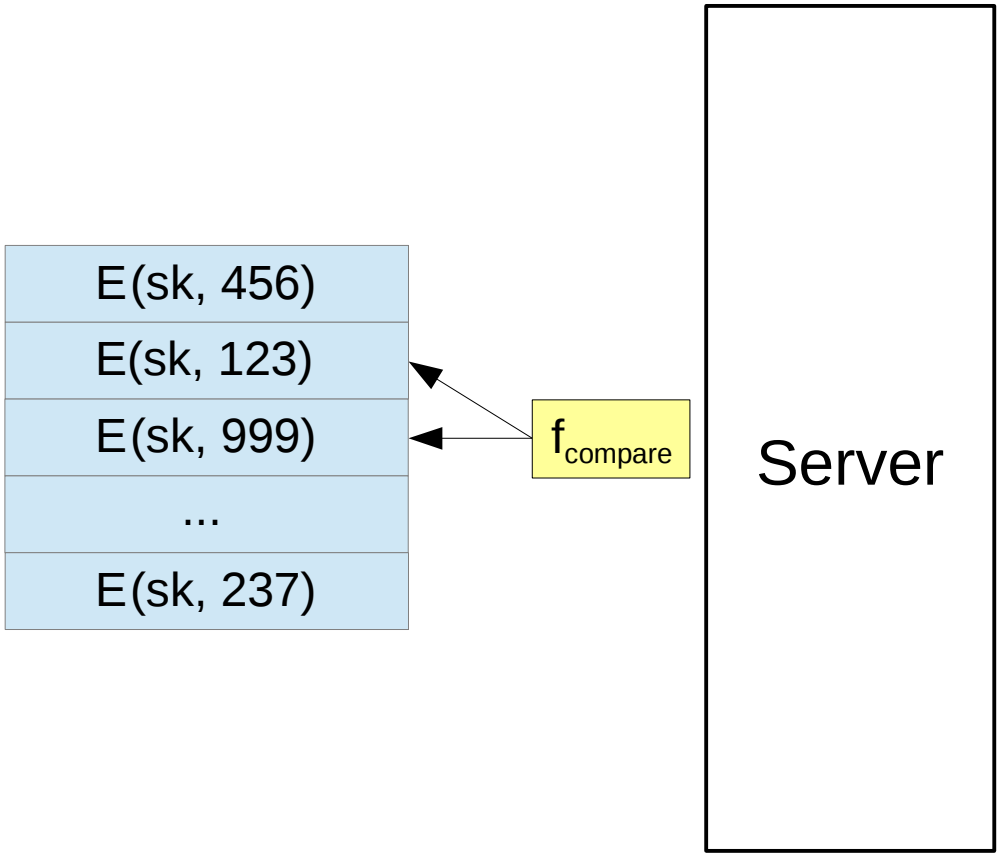
– Order-revealing encryption



Order-Revealing Encryption

Our goal:

– Order-revealing encryption



Order-Revealing Encryption

Our goal:

– Order-revealing encryption

Order-Revealing Encryption

Our goal:

- Order-revealing encryption
- Best-possible CPA security [BCLO09]:
  - CPA modulo adaptive comparison queries

Existing approaches

Approach 1:

– Via *Order-Preserving Encryption (OPE)* [BCLO09]

$$x < y \implies E(sk, x) < E(sk, y)$$

Existing approaches

Approach 1:

- Via *Order-Preserving Encryption (OPE)* [BCLO09]

  $x < y \implies E(sk, x) < E(sk, y)$

- BCLO'09: OPE cannot achieve best-possible CPA security

Existing approaches

Approach 1:

– Via *Order-Preserving Encryption (OPE)* [BCLO09]

$$x < y \implies E(sk, x) < E(sk, y)$$

– BCLO'09: OPE cannot achieve best-possible CPA security

Approach 2:

– Via *Multi-Input Functional Encryption (MIFE)*
[GGG+14, BKS15, BV15, AJ15]

Existing approaches

Approach 1:

- Via *Order-Preserving Encryption (OPE)* [BCLO09]

  $x < y \implies E(sk, x) < E(sk, y)$

- BCLO'09: OPE cannot achieve best-possible CPA security

Approach 2:

- Via *Multi-Input Functional Encryption (MIFE)*
  [GGG+14, BKS15, BV15, AJ15]
- Requires obfuscation of a PRF (e.g., AES)

Existing approaches

Approach 1:

– Via *Order-Preserving Encryption (OPE)* [BCLO09]

$$x < y \implies E(sk, x) < E(sk, y)$$

– BCLO'09: OPE cannot achieve best-possible CPA security

Approach 2:

– Via *Multi-Input Functional Encryption (MIFE)*
[GGG+14, BKS15, BV15, AJ15]

– Requires obfuscation of a PRF (e.g., AES)

Approach 3:

– Via Oblivious RAM methods [OS97,…]

Existing approaches

Approach 1:

- Via *Order-Preserving Encryption (OPE)* [BCLO09]

  $x < y \implies E(sk, x) < E(sk, y)$

- BCLO'09: OPE cannot achieve best-possible CPA security

Approach 2:

- Via *Multi-Input Functional Encryption (MIFE)*
  [GGG+14, BKS15, BV15, AJ15]
- Requires obfuscation of a PRF (e.g., AES)

Approach 3:

- Via Oblivious RAM methods [OS97,…]
- Requires interaction with client

Outline

1. Motivating example: O*rder-Revealing Encryption*

2. **Our results**
   - Order-Revealing Encryption
   - Generalization: Multi-Input Functional Encryption
     - directly from multilinear maps (no obfuscation)
     - security in the generic model

3. Overview of techniques

Our results

– First *implementable* order-revealing encryption scheme with best-possible CPA security

Our results

– First *implementable* order-revealing encryption scheme with best-possible CPA security

– Uses multilinear maps [GGH13a, CLT13, GGH14, CLT15]

Our results

- First *implementable* order-revealing encryption scheme with best-possible CPA security
- Uses multilinear maps [GGH13a, CLT13, GGH14, CLT15]
- No obfuscation
  - Inspired by obfuscation techniques

Our results

- First *implementable* order-revealing encryption scheme with best-possible CPA security

- Uses multilinear maps [GGH13a, CLT13, GGH14, CLT15]

- No obfuscation
  - Inspired by obfuscation techniques

- To compare 16-bit numbers:
  |CT| = 264 ring elements in a 9-linear map

Our results

– Prove best-possible CPA in generic model [GGH13a, BR13]

Our results

- Prove best-possible CPA in generic model [GGH13a, BR13]

- Generalization:
  - Other functions besides comparisons; multiple inputs

Our results

- Prove best-possible CPA in generic model [GGH13a, BR13]

- Generalization:
  - Other functions besides comparisons; multiple inputs
  - Best-possible CPA generalizes to property-preserving encryption [PR12]

Our results

- Prove best-possible CPA in generic model [GGH13a, BR13]

- Generalization:
  - Other functions besides comparisons; multiple inputs
  - Best-possible CPA generalizes to property-preserving encryption [PR12]
  - Instance of *Multi-Input Functional Encryption* [GGG+14] with a single secret key

Our results

- Prove best-possible CPA in generic model [GGH13a, BR13]

- Generalization:
  - Other functions besides comparisons; multiple inputs
  - Best-possible CPA generalizes to property-preserving encryption [PR12]
  - Instance of *Multi-Input Functional Encryption* [GGG+14] with a single secret key

- Not fast, but *implementable*!

Outline

1. Motivating example: O*rder-Revealing Encryption*

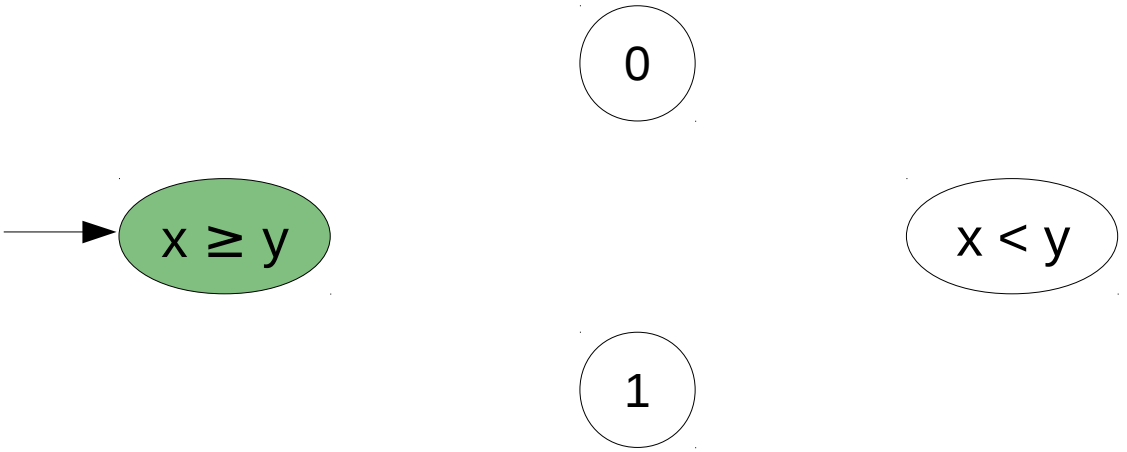
2. Our results
   - Order-Revealing Encryption
   - Generalization: Multi-Input Functional Encryption
     - directly from multilinear maps (no obfuscation)
     - security in the generic model

3. Overview of techniques

Overview of Techniques

– View comparison function as (multi-input) finite automaton

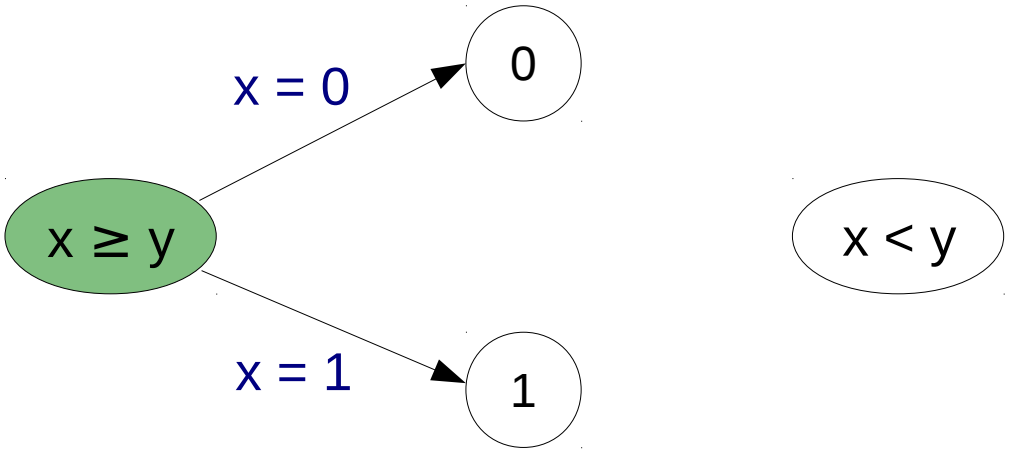
Compare(x = 10, y = 11):



Overview of Techniques

– View comparison function as (multi-input) finite automaton

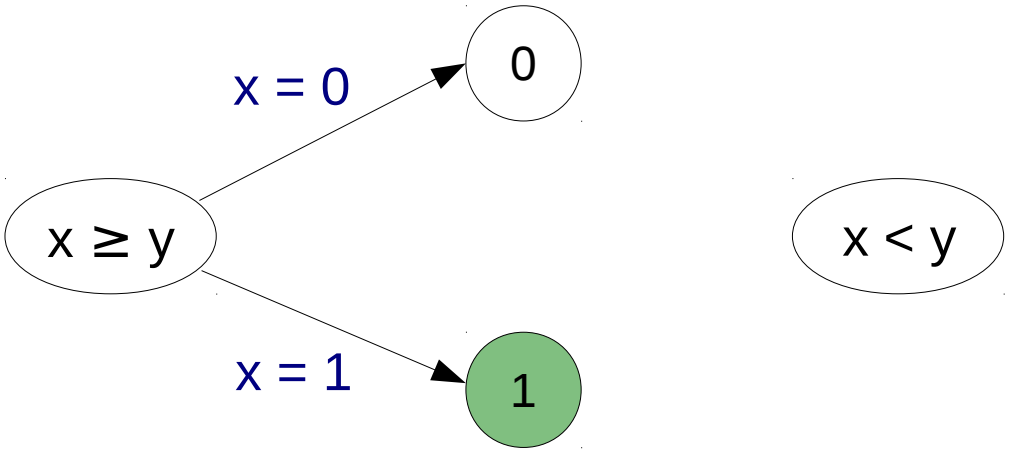
Compare(x = 10, y = 11):



Overview of Techniques

– View comparison function as (multi-input) finite automaton

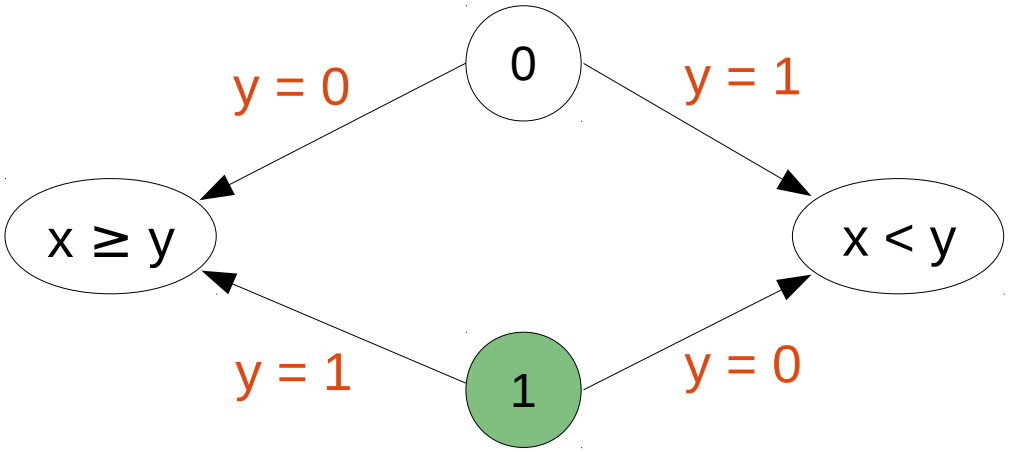
Compare(x = 10, y = 11):



Overview of Techniques

– View comparison function as (multi-input) finite automaton

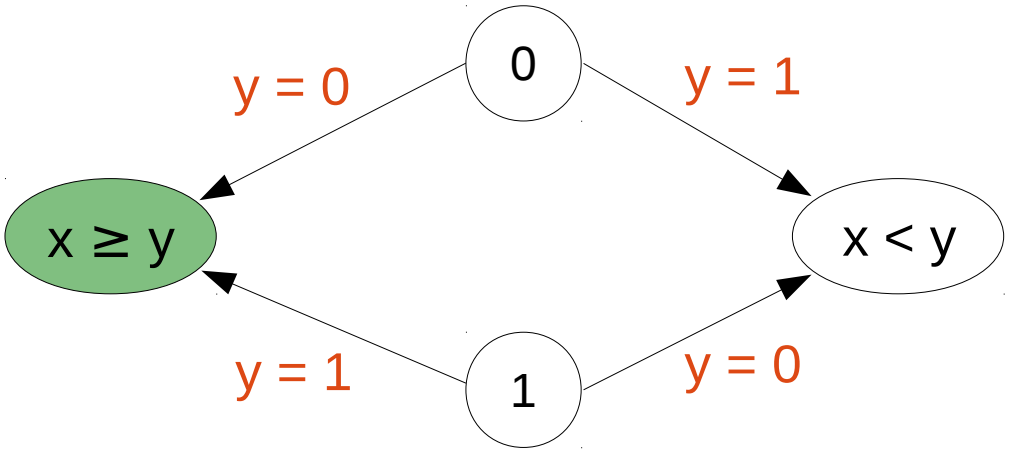
Compare(x = 10, y = 11):



Overview of Techniques

– View comparison function as (multi-input) finite automaton

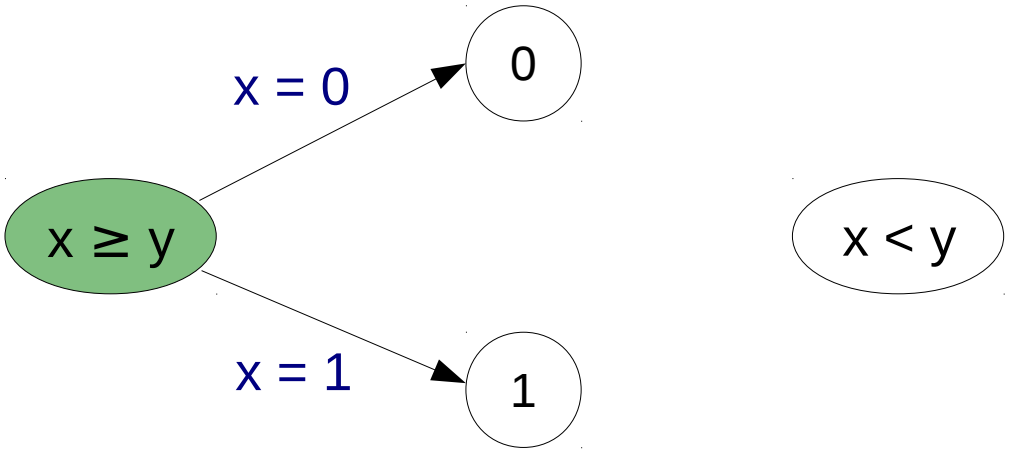
Compare(x = 10, y = 11):



Overview of Techniques

– View comparison function as (multi-input) finite automaton

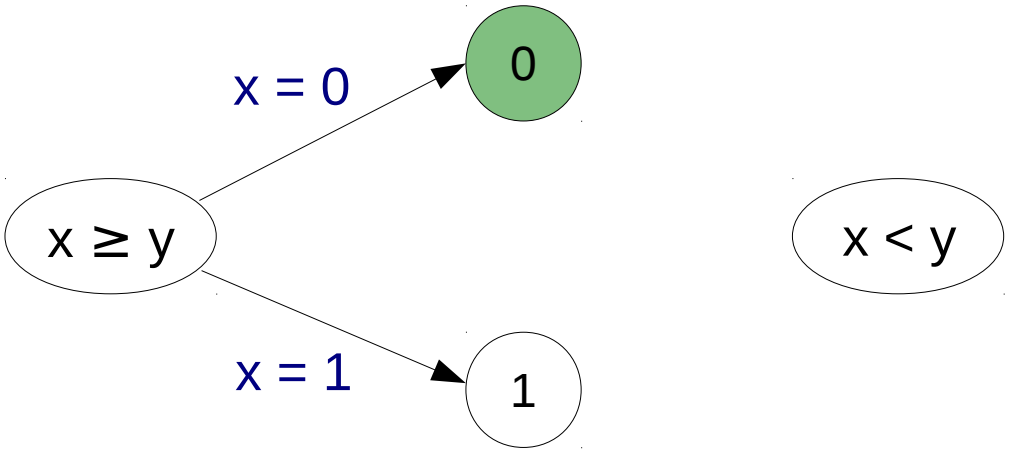
Compare(x = 10, y = 11):



Overview of Techniques

– View comparison function as (multi-input) finite automaton

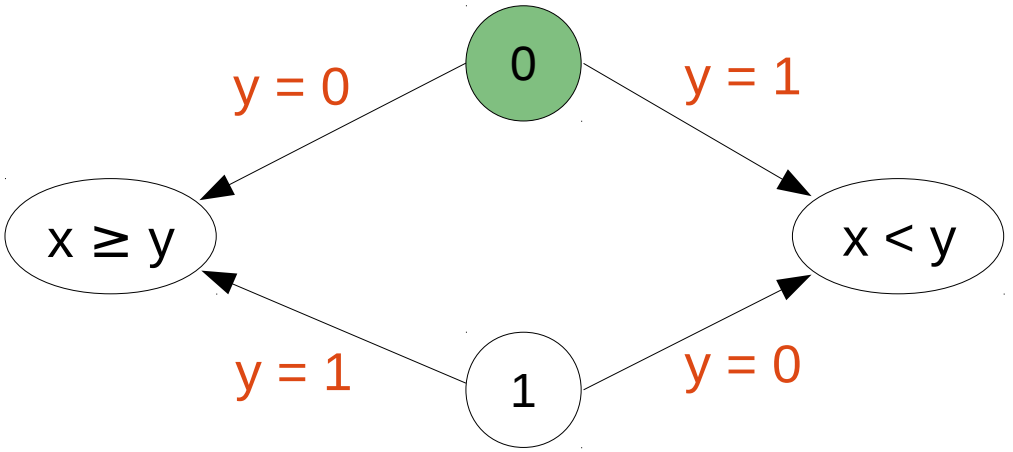
Compare(x = 10, y = 11):



Overview of Techniques

– View comparison function as (multi-input) finite automaton

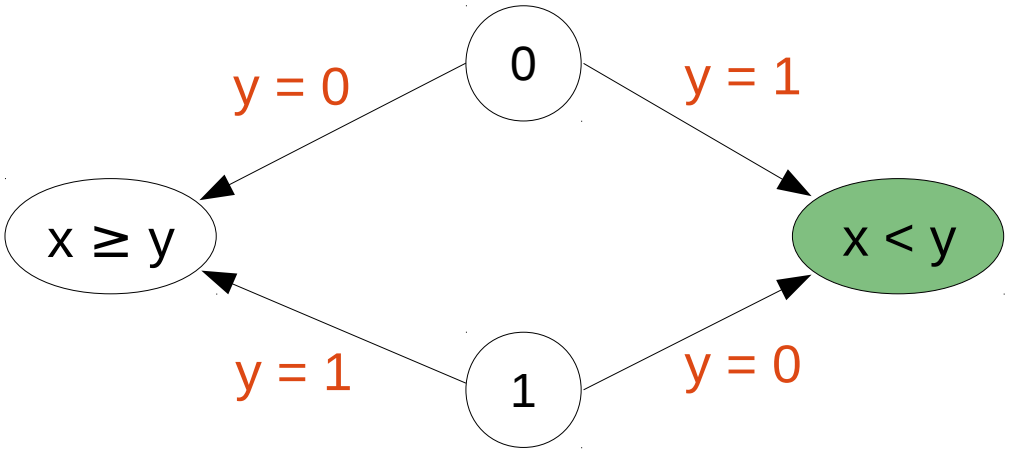
Compare(x = 10, y = 11):



Overview of Techniques

– View comparison function as (multi-input) finite automaton

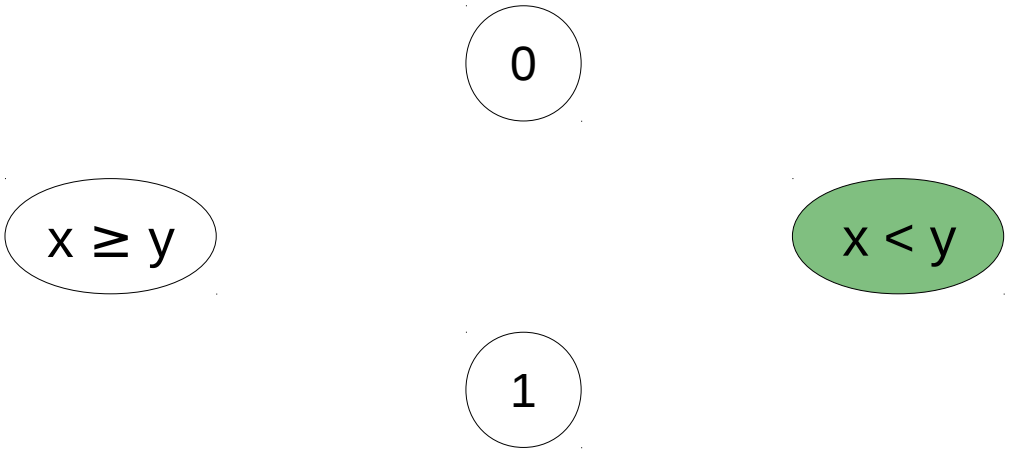
Compare(x = 10, y = 11):



Overview of Techniques

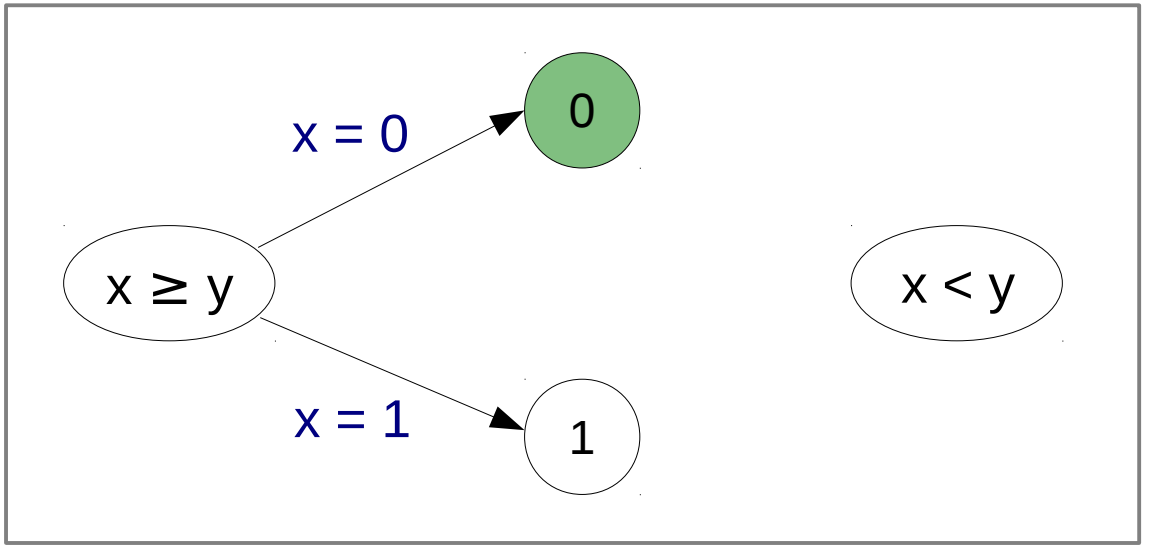
– View comparison function as (multi-input) finite automaton

Compare(x = 10, y = 11):



Overview of Techniques

– Represent automaton as sequence of transition matrices



x = 0:

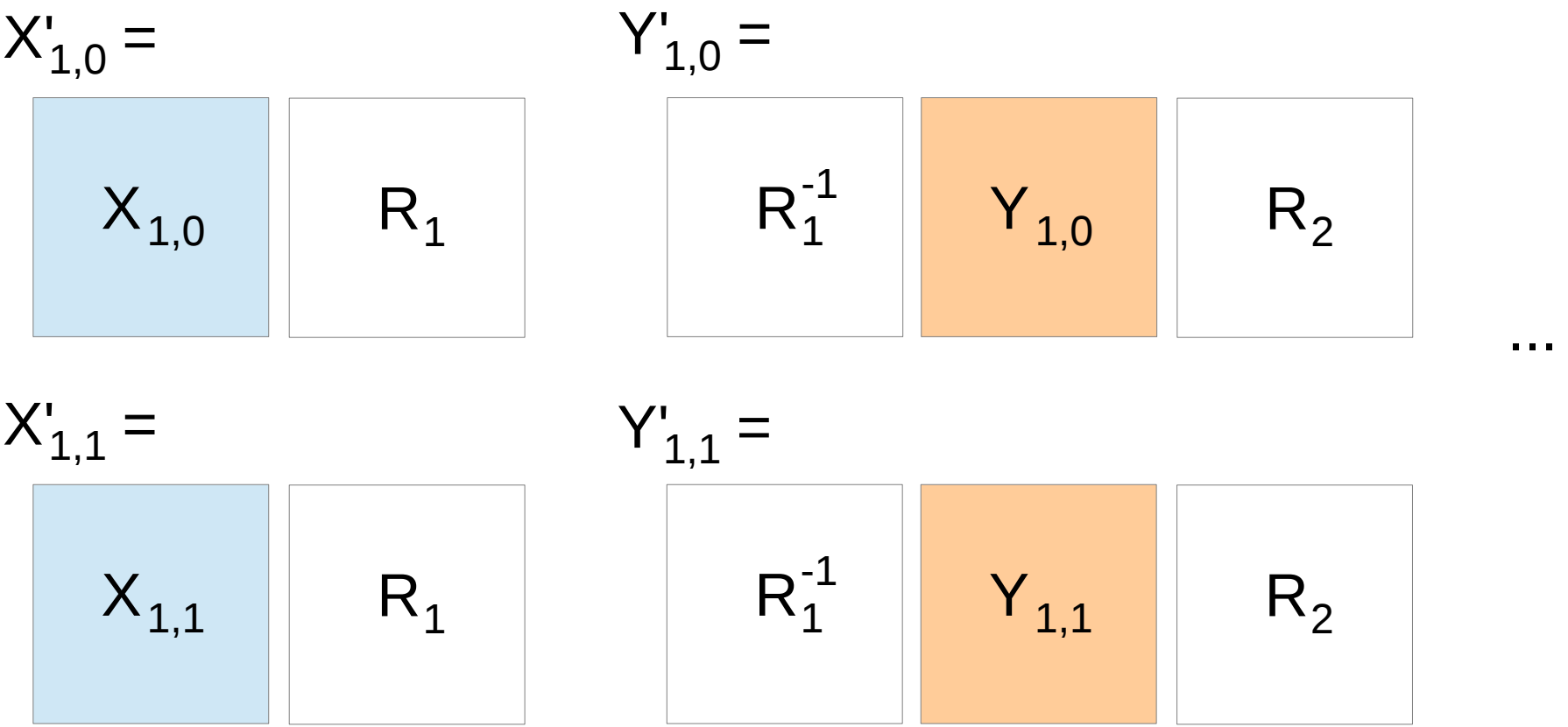
$$\begin{bmatrix} 1 & 1 & & \\ & & 1 & \\ & & & 1 \end{bmatrix}$$

x = 1:

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

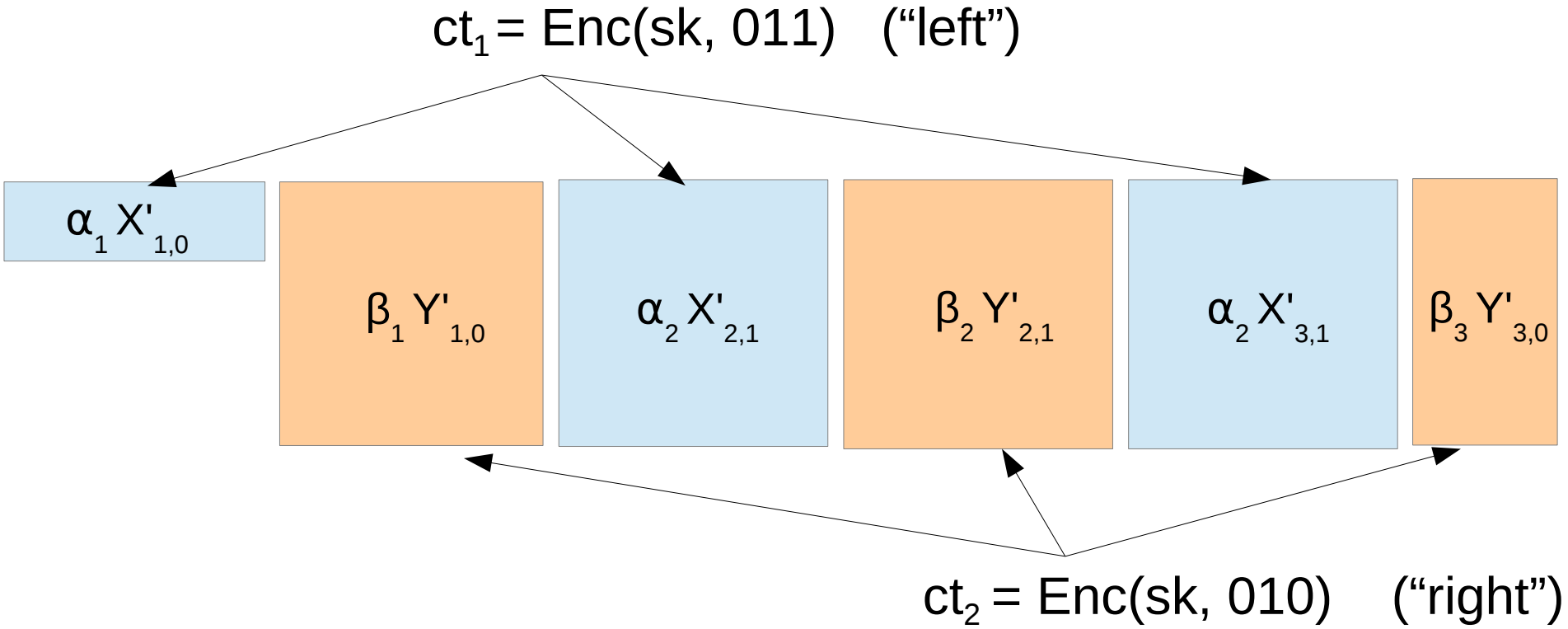
Overview of Techniques

– Randomize matrices via Kilian's protocol [Kil88]



Overview of Techniques

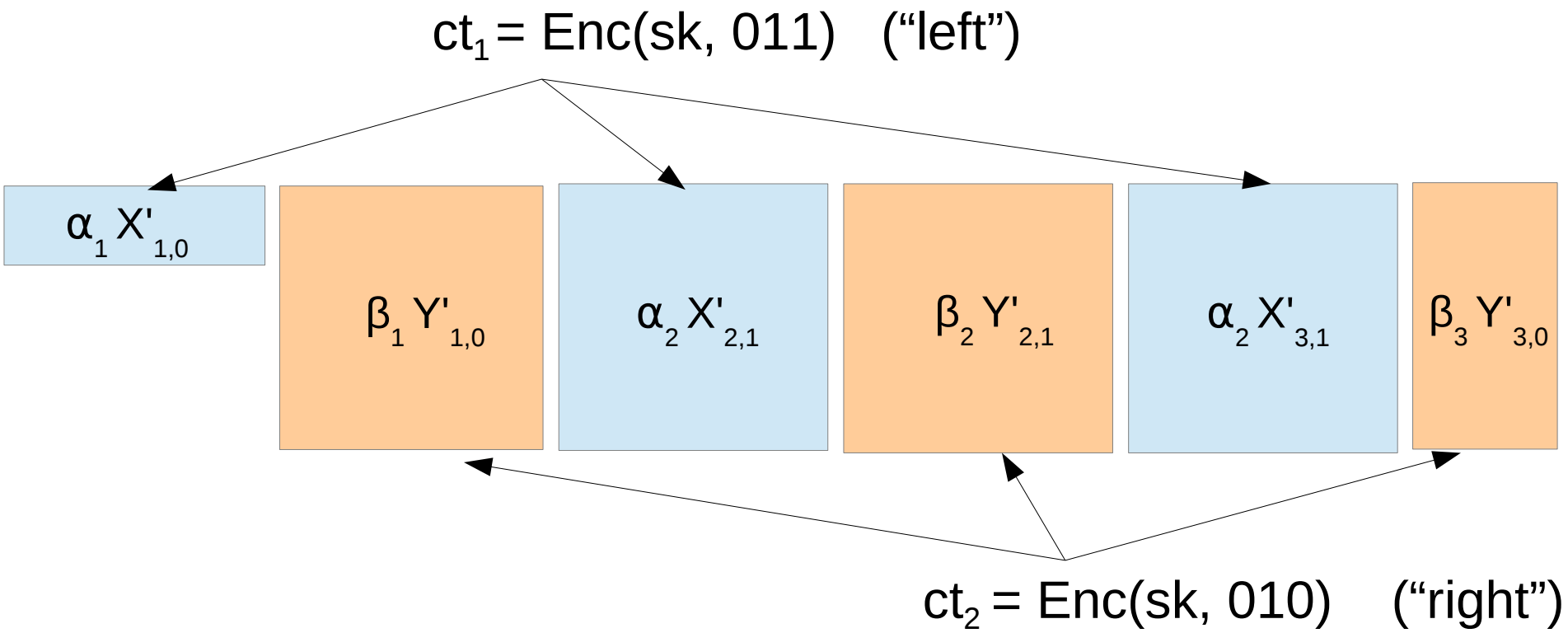
– Encode randomized matrices in multilinear map



Overview of Techniques

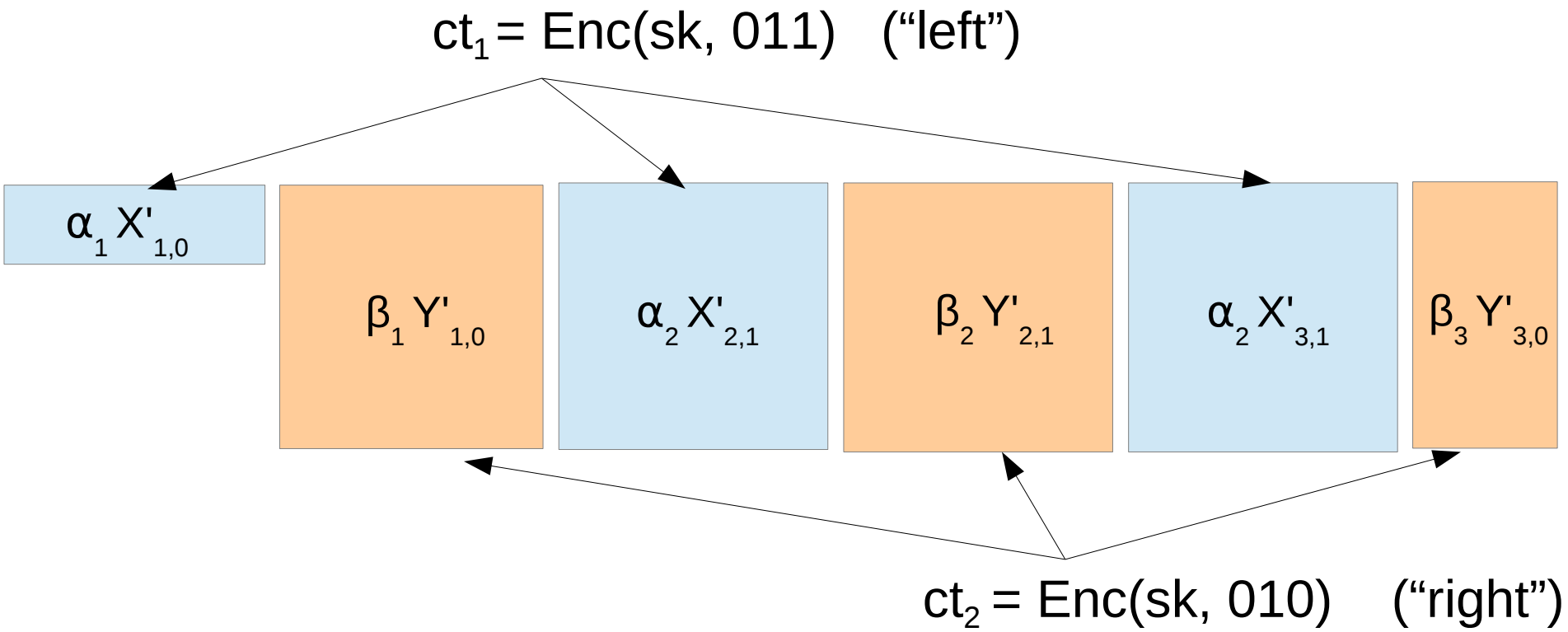
– Encode randomized matrices in multilinear map

– To compare: compute matrix products via map operations



Overview of Techniques

- Encode randomized matrices in multilinear map
- To compare: compute matrix products via map operations
- Zero-test to determine output of comparison

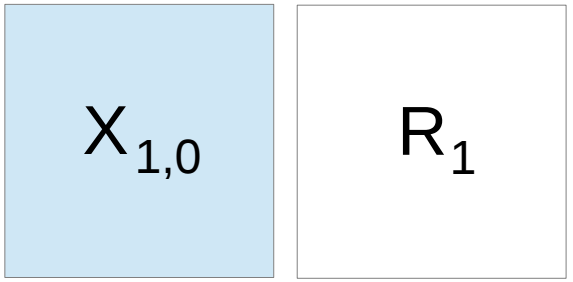


Order-Revealing Encryption

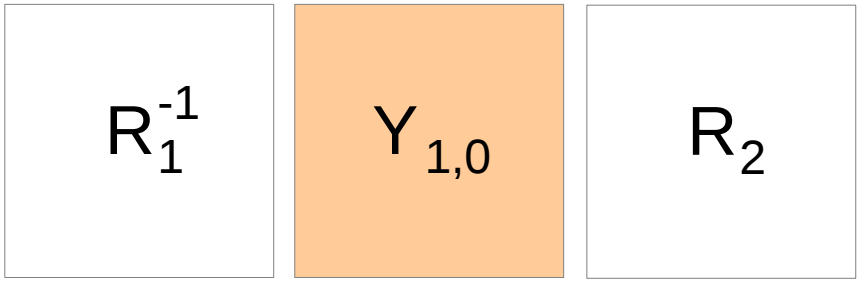
Problem 1:

– R matrices must be the same for every “left” ciphertext, same for every “right” ciphertext

$X'_{1,0}$ =

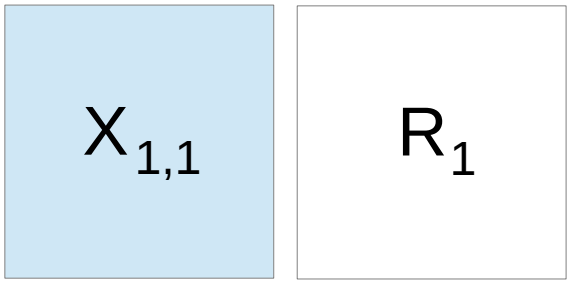


$Y'_{1,0}$ =

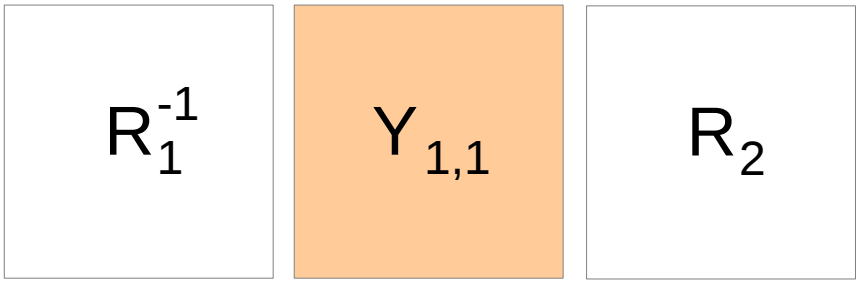


...

$X'_{1,1}$ =



$Y'_{1,1}$ =

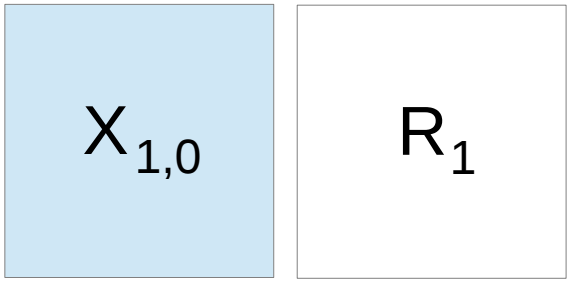


Order-Revealing Encryption

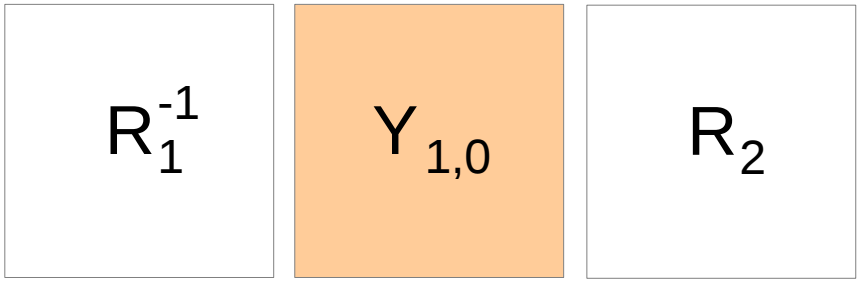
Problem 1:

- R matrices must be the same for every “left” ciphertext, same for every “right” ciphertext
- “Mix-and-match” attacks

$X'_{1,0}$ =

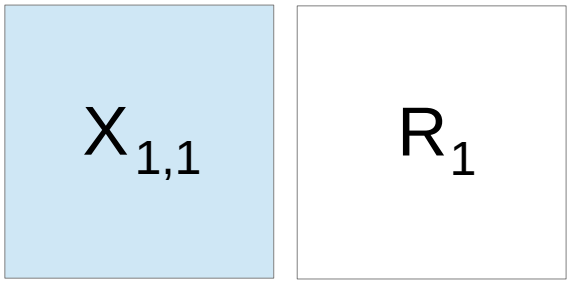


$Y'_{1,0}$ =

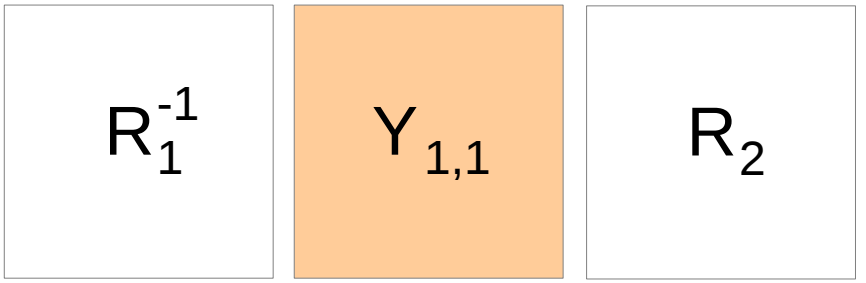


...

$X'_{1,1}$ =



$Y'_{1,1}$ =



Order-Revealing Encryption

Solution:

Order-Revealing Encryption

Solution:

– Use index sets in asymmetric multilinear map to enforce consistency

Order-Revealing Encryption

Solution:

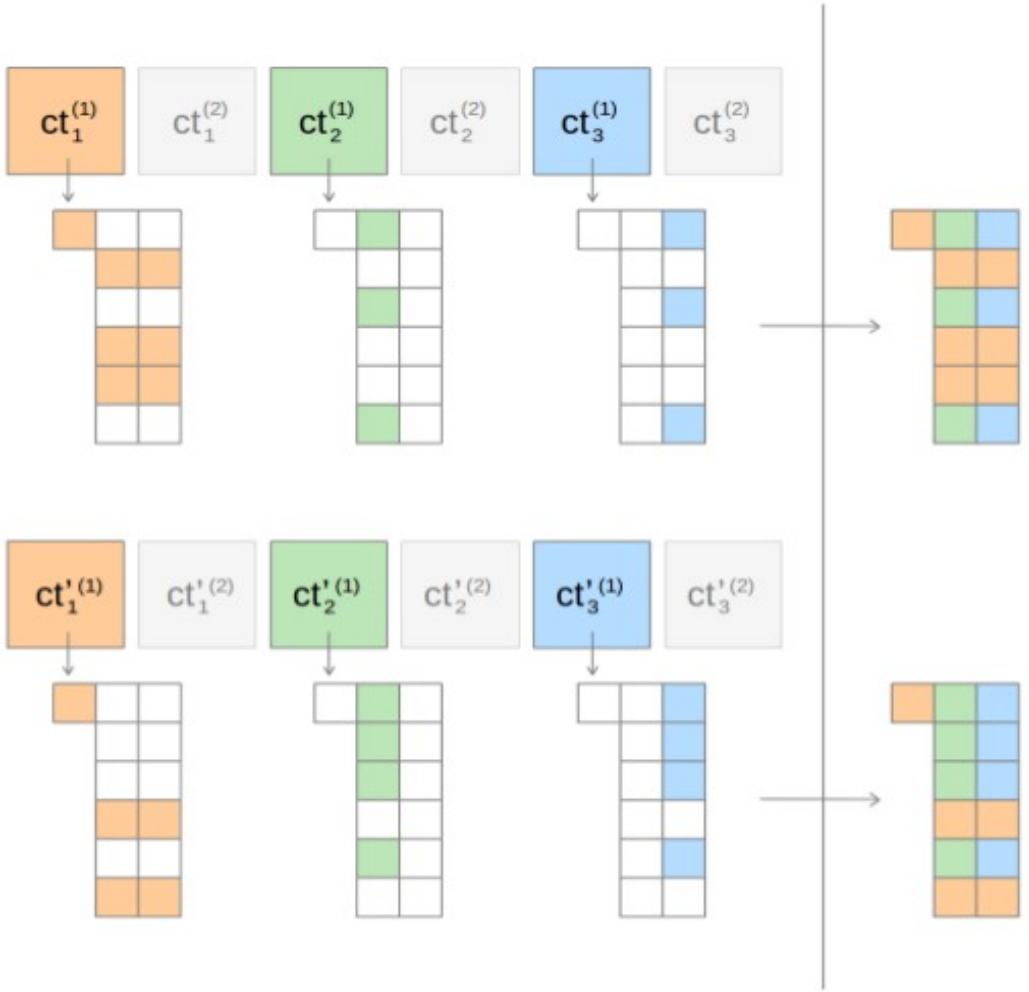
- Use index sets in asymmetric multilinear map to enforce consistency
- Encryption must be stateless → use *randomized* index sets

Order-Revealing Encryption

Our design:

– Index sets are drawn from “exclusive partition families”

– Generalize “straddling sets” [BGK+13]



Order-Revealing Encryption

Problem 2:

– To prove security, standard approach is via Kilian's simulation

Order-Revealing Encryption

Problem 2:

- To prove security, standard approach is via Kilian's simulation
  - Only valid for full-rank matrices

Order-Revealing Encryption

Problem 2:

- To prove security, standard approach is via Kilian's simulation
  - Only valid for full-rank matrices
  - Our matrices may not have full rank

Order-Revealing Encryption

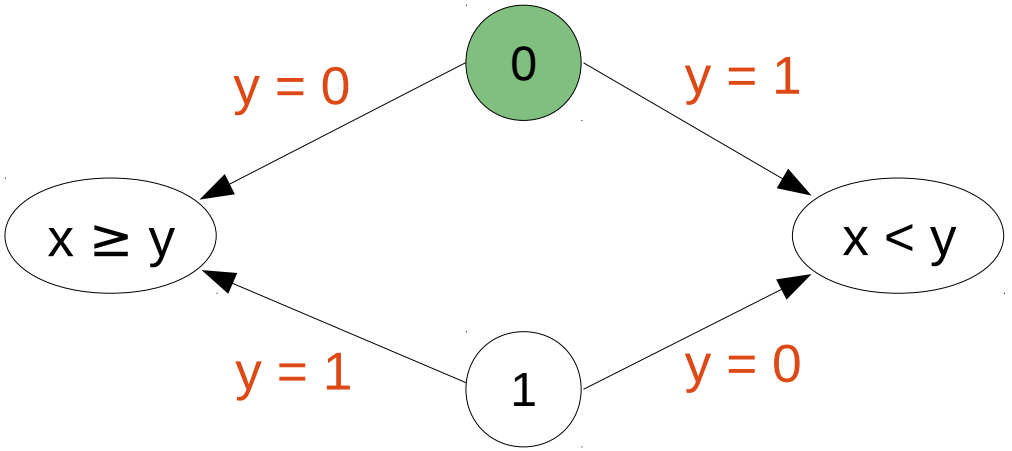
Problem 2:

- To prove security, standard approach is via Kilian's simulation
  - Only valid for full-rank matrices
  - Our matrices may not have full rank
    - e.g., for comparison functionality: steps of finite automaton are not injective

Order-Revealing Encryption

Problem 2:

- To prove security, standard approach is via Kilian's simulation
  - Only valid for full-rank matrices
  - Our matrices may not have full rank
    - e.g., for comparison functionality: steps of finite automaton are not injective



Order-Revealing Encryption

Solution:

Order-Revealing Encryption

Solution:

– Use Kilian's protocol, but not its simulation theorem

Order-Revealing Encryption

Solution:

- Use Kilian's protocol, but not its simulation theorem

- Instead prove that distributions are indistinguishable to cross-linear tests [SZ14]

Order-Revealing Encryption

Solution:

- Use Kilian's protocol, but not its simulation theorem
- Instead prove that distributions are indistinguishable to cross-linear tests [SZ14]
- Index sets enforce that the adversary gets only such tests

Order-Revealing Encryption

Main theorems:

Order-Revealing Encryption

Main theorems:

– Our construction achieves best-possible CPA security for order-revealing encryption

Order-Revealing Encryption

Main theorems:

- Our construction achieves best-possible CPA security for order-revealing encryption
  - Proof in generic multilinear map model

Order-Revealing Encryption

Main theorems:

- Our construction achieves best-possible CPA security for order-revealing encryption
  - Proof in generic multilinear map model

- Generalization: best-possible CPA security for Multi-Input Functional Encryption with a single secret key (1SK-MIFE)

Order-Revealing Encryption

Main theorems:

- Our construction achieves best-possible CPA security for order-revealing encryption
  - Proof in generic multilinear map model

- Generalization: best-possible CPA security for Multi-Input Functional Encryption with a single secret key (1SK-MIFE)
  - For branching programs, our 1SK-MIFE is *implementable*

Order-Revealing Encryption

Main theorems:

- Our construction achieves best-possible CPA security for order-revealing encryption
  - Proof in generic multilinear map model

- Generalization: best-possible CPA security for Multi-Input Functional Encryption with a single secret key (1SK-MIFE)
  - For branching programs, our 1SK-MIFE is *implementable*
  - Can also realize secret-key MIFE with multiple keys at the cost of a universal program transformation

Conclusion

– First *implementable* order-revealing encryption scheme with best-possible CPA security

Conclusion

- First *implementable* order-revealing encryption scheme with best-possible CPA security
  - To compare 16-bit numbers:
    |CT| = 264 ring elements in a 9-linear map

Conclusion

- First *implementable* order-revealing encryption scheme with best-possible CPA security
  - To compare 16-bit numbers:
    |CT| = 264 ring elements in a 9-linear map
  - Open problem: practical ORE

Conclusion

- First *implementable* order-revealing encryption scheme with best-possible CPA security
  - To compare 16-bit numbers:
    |CT| = 264 ring elements in a 9-linear map
  - Open problem: practical ORE
- Construction generalizes to implementable MIFE for many functionalities of interest