I·C·S
The Institute for Cyber Security

C·SPECC
Center for Security and Privacy Enhanced Cloud Computing

Feasibility Analysis of Access Control Policy Mining

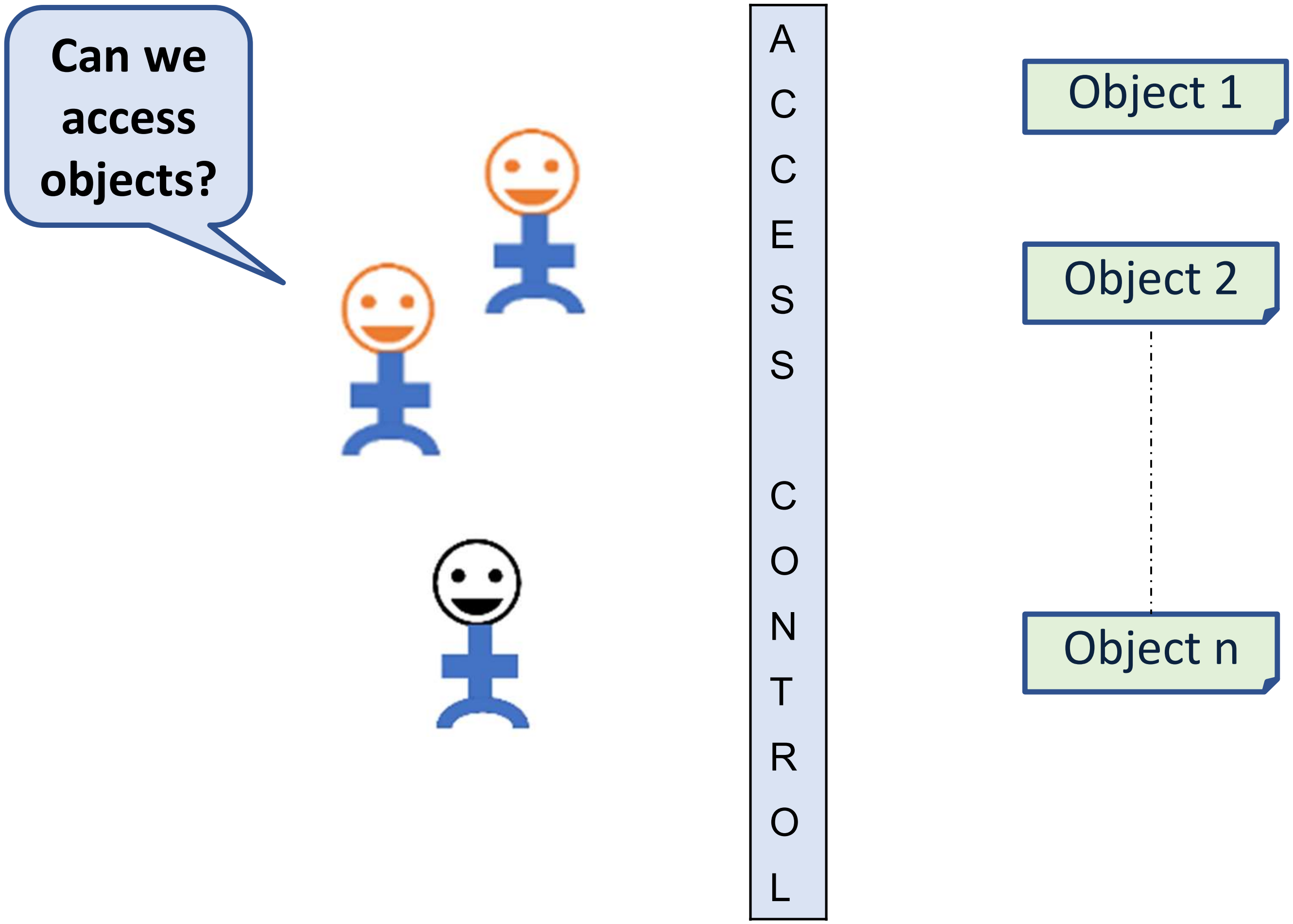
PhD Dissertation Defense

Shuvra Chakraborty
Institute for Cyber Security
Department of Computer Science
The University of Texas at San Antonio

Committee:
Dr. Ravi Sandhu (Advisor and Chair)
Dr. Palden Lama
Dr. Wei Wang
Dr. Xiaoyin Wang
Dr. Ram Krishnan

November 4, 2021

Introduction

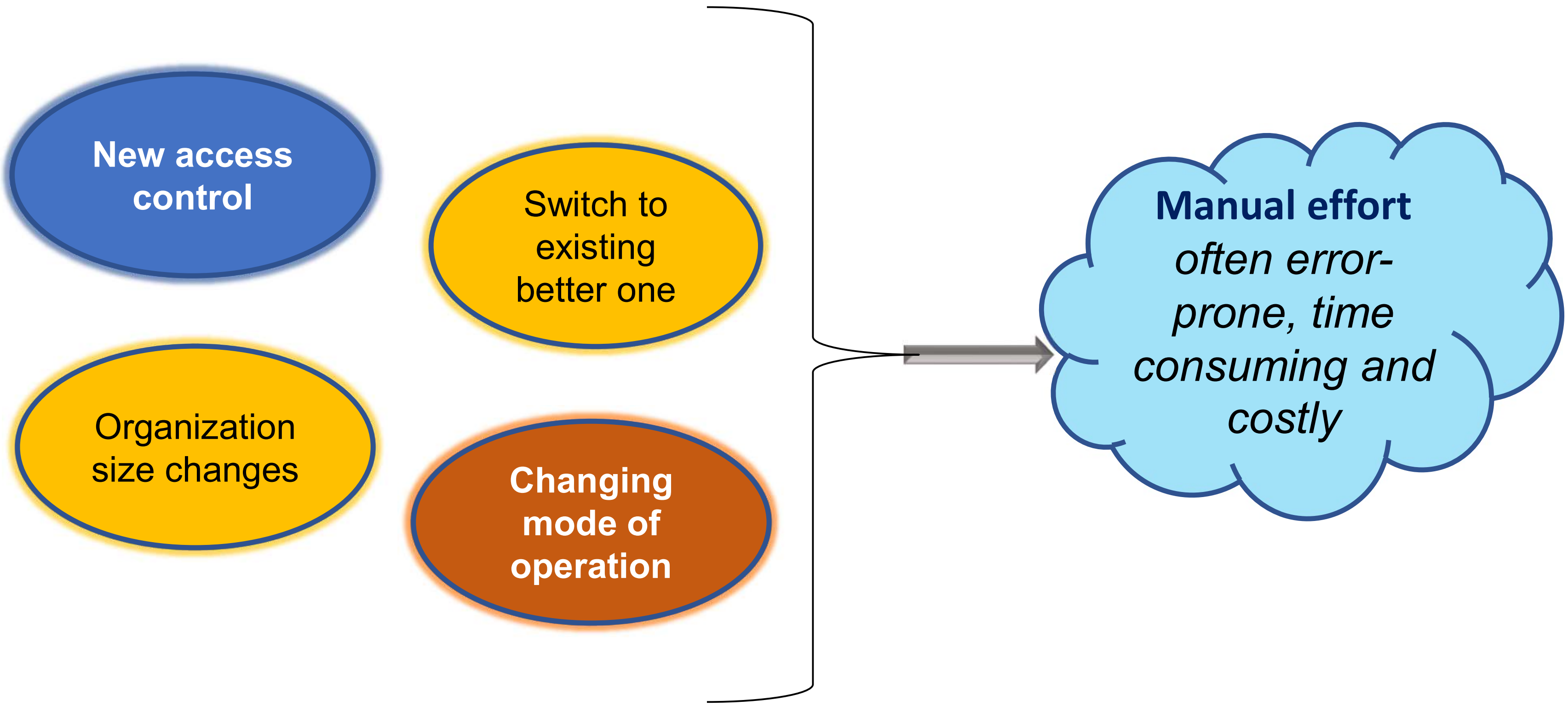


Legitimate users get legitimate access only

i.e., Role-Based Access Control (RBAC), Attribute-Based Access Control (ABAC)

Introduction

❖ **Problem:** migration from an existing access control model to another one



Is automation possible?

Introduction

Access Control List / Log / RBAC + Supporting attribute data → ABAC policy mining

Access Control List + Supporting Relationship data → ReBAC policy mining

Given an access control system + Supporting data → Another access control model

General term
Access control policy mining

Mining is partially automated so far…

*** Relationship-Based Access Control (ReBAC)

THESIS STATEMENT

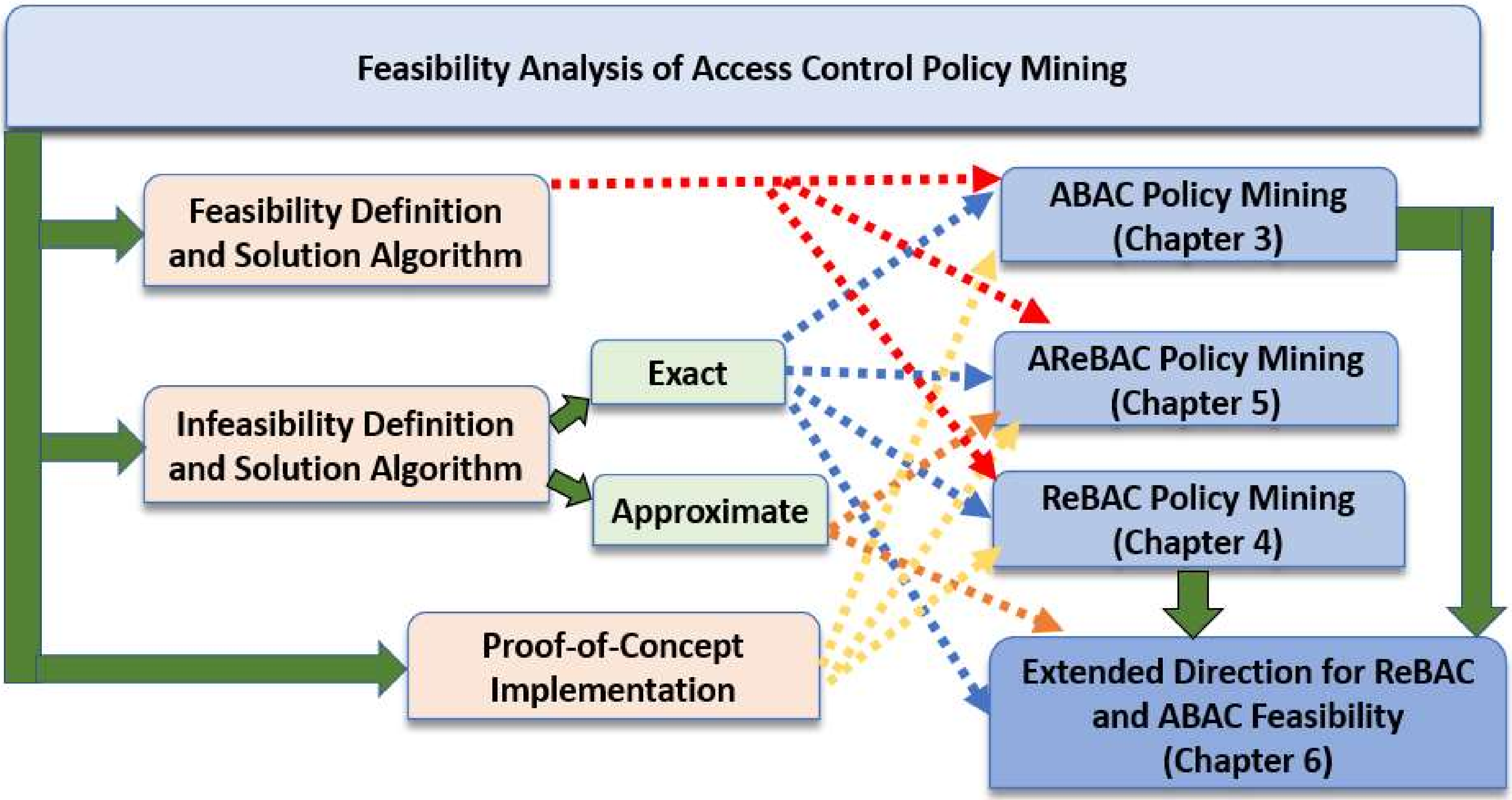
*As a matter of growing real-world challenges and advancements in technology, migration of one access control system to another is an emerging problem. The complete or partially automated solution to this migration process is called the **access control policy mining problem**. During the mining process, a set of assumptions and criteria are imposed to precisely define the migration goals.*

*The **feasibility analysis of the access control policy mining** problem formulates the logical framework of the problem, resolves the infeasibility issues possibly arising during the policy mining process so that the solution can satisfy those imposed criteria, and provides a rigorous foundation for the migration process.*

Thesis Scopes

- Works with the domain of access control models.
  - RBAC, ABAC, ReBAC, etc.
- Performance measurement is limited to mathematical proof and analyzing algorithmic complexity.
- Clear boundary of feasibility issues is yet to be defined.
- Depends on how access control models are defined. A separate study is required to extend this.
- Does not compete with human expertise at all.
- Focusses on <u>exact</u> solutions mostly.

Summary of Contributions



World-Leading Research with Real-World Impact!

Chapter 3

Background

Enumerated Authorization System (EAS) is a tuple

$<U, O, OP, AUTH, checkAccess_{EAS}>$

- U, O, and OP are finite sets of users, objects and operations, respectively.
- AUTH $\subseteq$ UXOXOP

Example 1:

- U = {John, Lina, Ray, Tom}, OP = {read, write}, O = {Obj1, Obj2}

AUTH	Explanation
(John, Obj1, write) (John, Obj2, write) (John, Obj1, read) (Lina, Obj2, write) (Tom, Obj1, read) (Ray, Obj1, read)	e.g., John is allowed to do write operation on Obj2 but read operation is not allowed.

Background

RBAC system is a tuple <U, O, OP, Roles, RPA, RUA, RH, $checkAccess_{RBAC}$>

- ❖ RPA : Role Permission Assignment
- ❖ RUA: Role User Assignment
- ❖ Permission is an object-operation pair
- ❖ RH is the role hierarchy relation

<u>Example 2:</u>

- U = {John, Lina, Ray, Tom}, OP = {read, write}, O = {Obj1, Obj2} [same as Example 1]
- Roles = {R1, R2, R3}
- RPA(R1) = {(Obj1, write)}, RPA(R2) = {(Obj2, write)}, RPA(R3) = {(Obj1, read)}
- RUA(R1) = {John}, RUA(R2) = {Lina}, RPA(R3) = {Ray, Tom}
- RH={(R1,R2), (R1, R3)}   [R1 is a senior role than R2, R3]

EAS and RBAC systems are equivalent

Background

ABAC system is a tuple <U, O, OP, UA, OA, UAValue, OAValue, RangeSet, RuleSet, $\text{checkAccess}_{ABAC}$ >

Example 3

- U, O, OP are same as Example 1
- UA ={Position, Dept.}, OA = {Type}

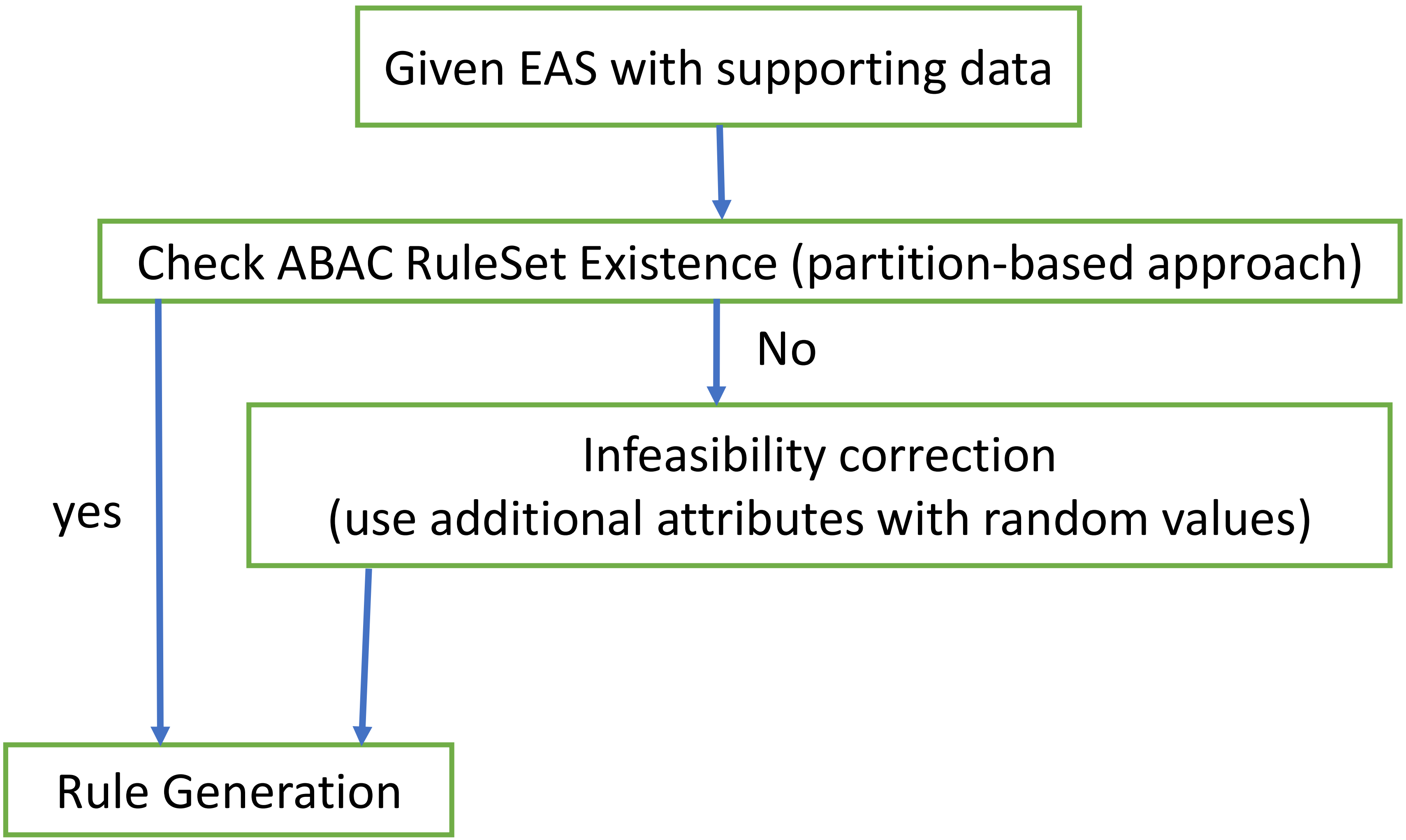
UAValue		
User (U)	**Position**	**Dept.**
John	Officer	CS
Lina	Student	CS
Ray	Officer	CS
Tom	Officer	CS

RangeSet	
Position	{Officer, Student, Faculty}
Dept.	{CS, EE}
Type	{File, Printer, Scanner}

OAValue	
Object (O)	**Type**
Obj1	File
Obj2	Printer

- RuleSet contains one separate rule for each operation, $\{\text{Rule}_{read}, \text{Rule}_{write}\}$
- ***ABAC system is incomplete in Example 3***

Workflow-1



Given EAS with supporting data
Check ABAC RuleSet Existence (partition-based approach)
No
Infeasibility correction
(use additional attributes with random values)
yes
Rule Generation

Unrepresented Partition

{Stud., Fac., Off.}

{CS, EE}

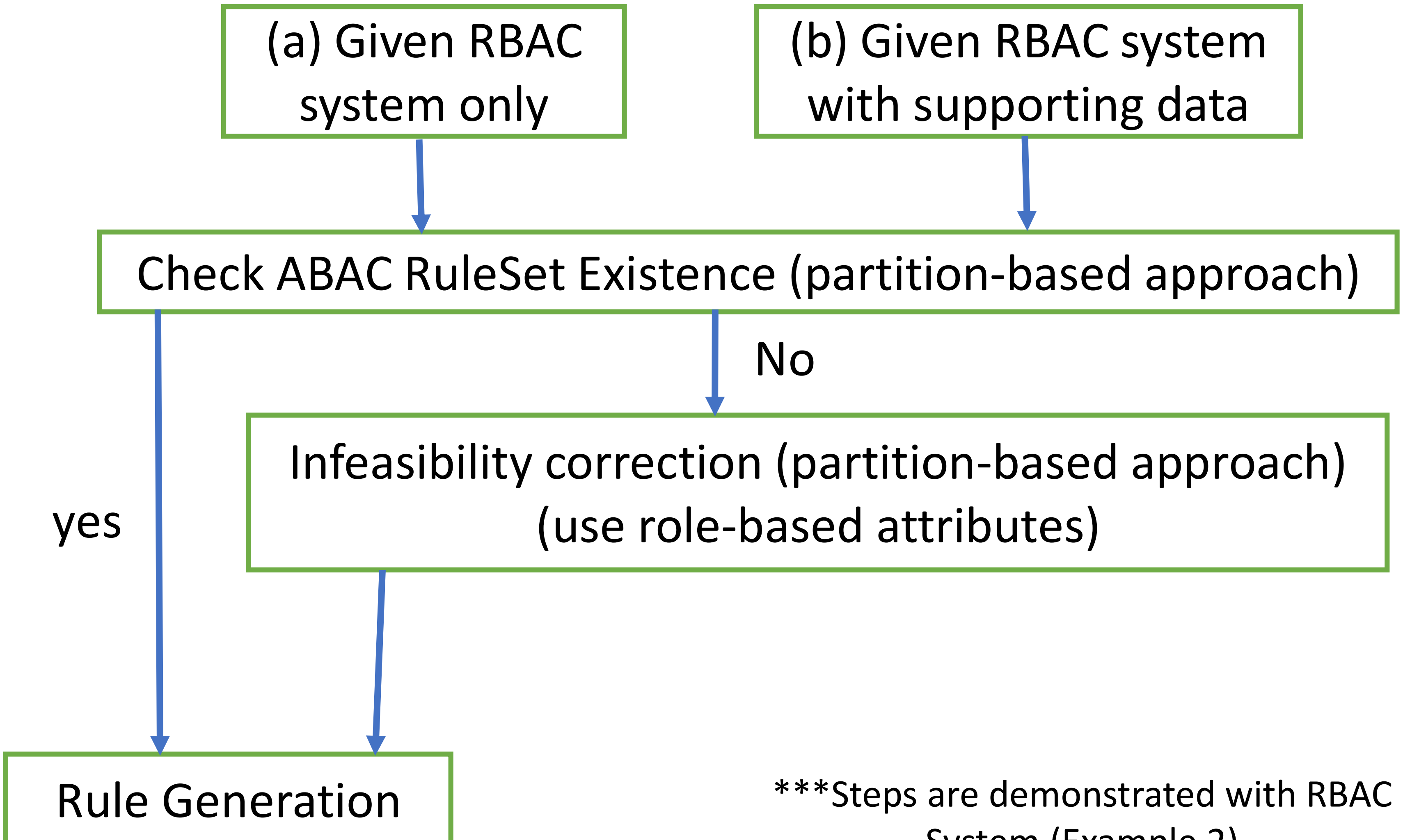
{F., Pr., Sc.}

<u>Represented: 4</u>
e.g., (Off., CS, F.), (Stud., CS, Pr.)

<u>Unrepresented: 14</u>
e.g., (Fac., CS, Pr.), (Stud., EE, Pr.)

Outcome of peculiarity in attribute value assignment

Workflow-2



***Steps are demonstrated with RBAC System (Example 2)

(a) RBAC only

Step 1. Generate role-based attribute set

- For a user u, role-based user attribute denotes the set of roles possessed by u
- For an object-operation pair (obj, op), role-based object attribute denotes the set of roles where each role contains permission (obj, op)

UAValue	
User(U)	**uroleAtt**
John	{R1, R2, R3}
Lina	{R2}
Ray	{R3}
Tom	{R3}

OAValue		
Object(O)	**$oroleAtt_{write}$**	**$oroleAtt_{read}$**
Obj1	{R1}	{R1, R3}
Obj2	{R1, R2}	{}

Next step: partition set is generated on set UXO based on similarity in attribute value assignment

I·C·S The Institute for Cyber Security

Step 2

C·SPECC Center for Security and Privacy Enhanced Cloud Computing

Partition set w.r.t. write

Ray, Obj1
Tom, Obj1

Ray, Obj2
Tom, Obj2

John, Obj2

Lina, Obj1

John, Obj1

Lina, Obj2

Partition set w.r.t. read

Ray, Obj1
Tom, Obj1

Ray, Obj2
Tom, Obj2

John, Obj2

Lina, Obj1

John, Obj1

Lina, Obj2

Partition set is conflict-free w.r.t. read and write → YES

Step 3

- Given an operation op, if partition set is conflict-free and each partition is uniquely identified by the set of (attribute name, value) pair then RuleSet can be generated [Proved]
- A conjunction of (attribute name, value) pair is made for each conflict-free bold black partition and OR'ed to Rule_{op}

e.g., $\text{Rule}_{read} \equiv$ <**(uroleAtt(u) = {R3} $\wedge$ oroleAtt_{write} (o)={R1} $\wedge$ oroleAtt_{read} (o) = {R1, R3})** **V** **(uroleAtt(u) = {R1, R2, R3} $\wedge$ oroleAtt_{write}(o)= {R1} $\wedge$ oroleAtt_{read} (o)= {R1, R3})**>

*****Rule_{write} can be constructed same way**
***RuleSet = {Rule_{write}, Rule_{read}}**
*****Equivalent ABAC system generation is always possible!**

I·C·S
The Institute for Cyber Security

(b) With supporting data

C·SPECC
Center for Security and Privacy Enhanced Cloud Computing

Role Based Access Control System

Equivalent ABAC system

Supporting Data

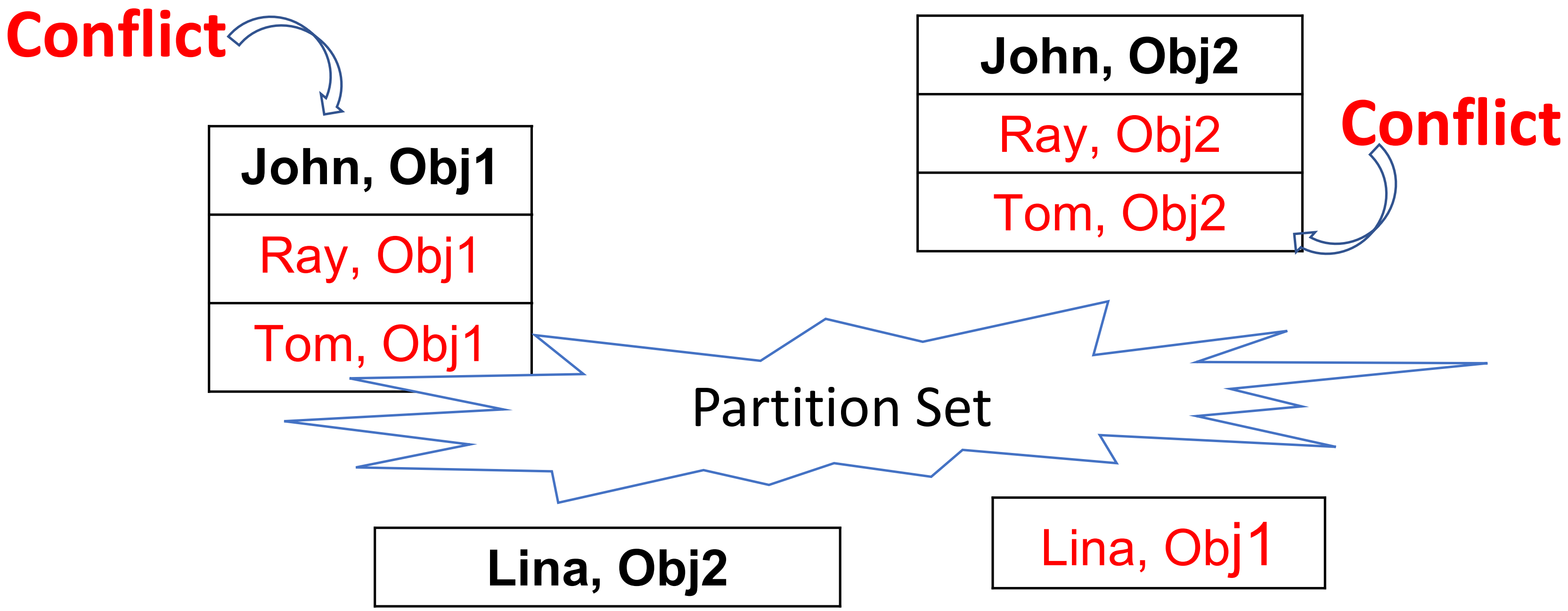
UAValue		
User (U)	**Position**	**Dept.**
John	Officer	CS
Lina	Student	CS
Ray	Officer	CS
Tom	Officer	CS

OAValue	
Object (O)	**Type**
Obj1	File
Obj2	Printer

RangeSet	
Position	{Officer, Student, Faculty}
Dept.	{CS, EE}
Type	{File, Printer, Scanner}

Step 1: Generate partition set based on similarity in attribute value assignment. Partition set might have conflicts!

Step 1

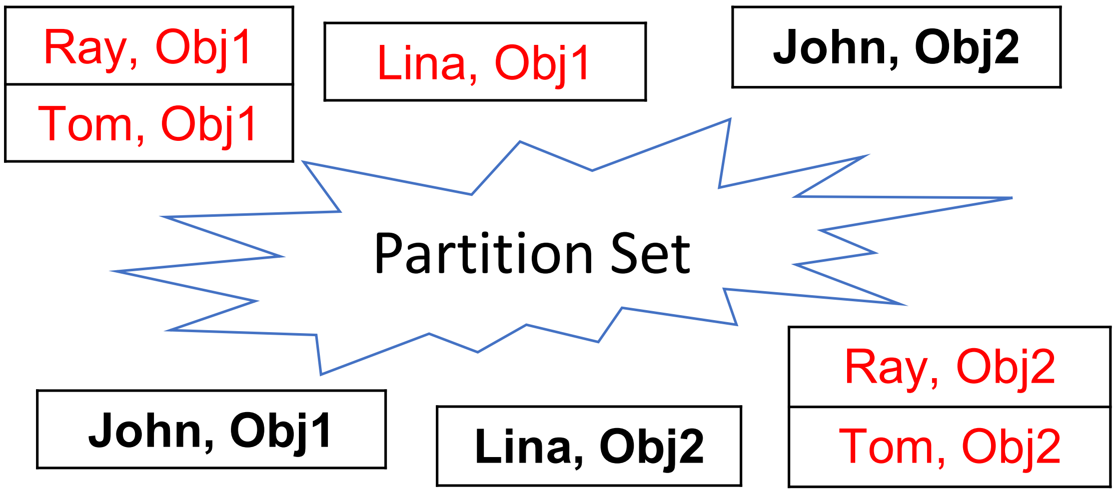


*Partition set has conflict w.r.t. write → YES
Next step: Apply infeasibility correction

The Institute for Cyber Security

Partition set: corrected

Center for Security and Privacy Enhanced Cloud Computing



UAValue	
User(U)	uroleAtt
John	{R1, R2, R3}
Lina	{R2}
Ray	{R3}
Tom	{R3}

OAValue		
Object (O)	oroleAtt_{write}	oroleAtt_{read}
Obj1	{R1}	{R1, R3}
Obj2	{R1, R2}	{}

Rule_{write} ≡ <(Position(u) = officer Λ Dept(u) = CS Λ uroleAtt(u)={R1, R2, R3} Λ Type(o) = File) V (Position(u) = officer Λ Dept(u) = CS Λ uroleAtt(u)={R1, R2, R3} Λ Type(o) = Printer) V (Position(u) = student Λ Dept(u) = CS Λ Type(o) = Printer)>

***RuleSet = {Rule_{write}, Rule_{read}}**

Summary

- **Formalized notion: feasibility of ABAC policy mining for the first time**
- The overall asymptotic complexity of ABAC RuleSet Existence problem is $O(|OP| \times (|U| \times |O|))$
- The overall asymptotic complexity of ABAC RuleSet Infeasibility Correction is: $O(|OP| \times (|U| \times |O|)^3)$

Challenges

- Ensure minimal partition split
- More compact set of rule generation
- Negative ABAC rules
- Exact solution
  - *Reduce number of split partitions
  - *Change number of attributes required
  - *Effect on changing existing attribute set

Chapter 4

Background: ReBAC

- *ReBAC ≡ Relationship-Based Access Control*
  - ReBAC expresses authorization in terms of various direct and indirect relationships amongst entities, most commonly between users
  - Access conditions are usually based on type, depth, or strength of relationships
- Assumption
  - Relationship Graph (RG) where users(node) are connected(edge) by social relationships(edge label). Each edge in the RG is labeled with a relation type
  - Only user-to-user relationships are considered

Problem Statement

The feasibility analysis of the ReBAC policy mining problem studies whether the migration process from a given authorization set to ReBAC policy is feasible or not under the set of imposed criteria:

- Relationship Graph (RG) is given
- ReBAC rule structure is given
- <u>Use of entity ID is not allowed</u>
  - <u>Existing literature allows ID</u>
- Equivalent set of ReBAC rules are required

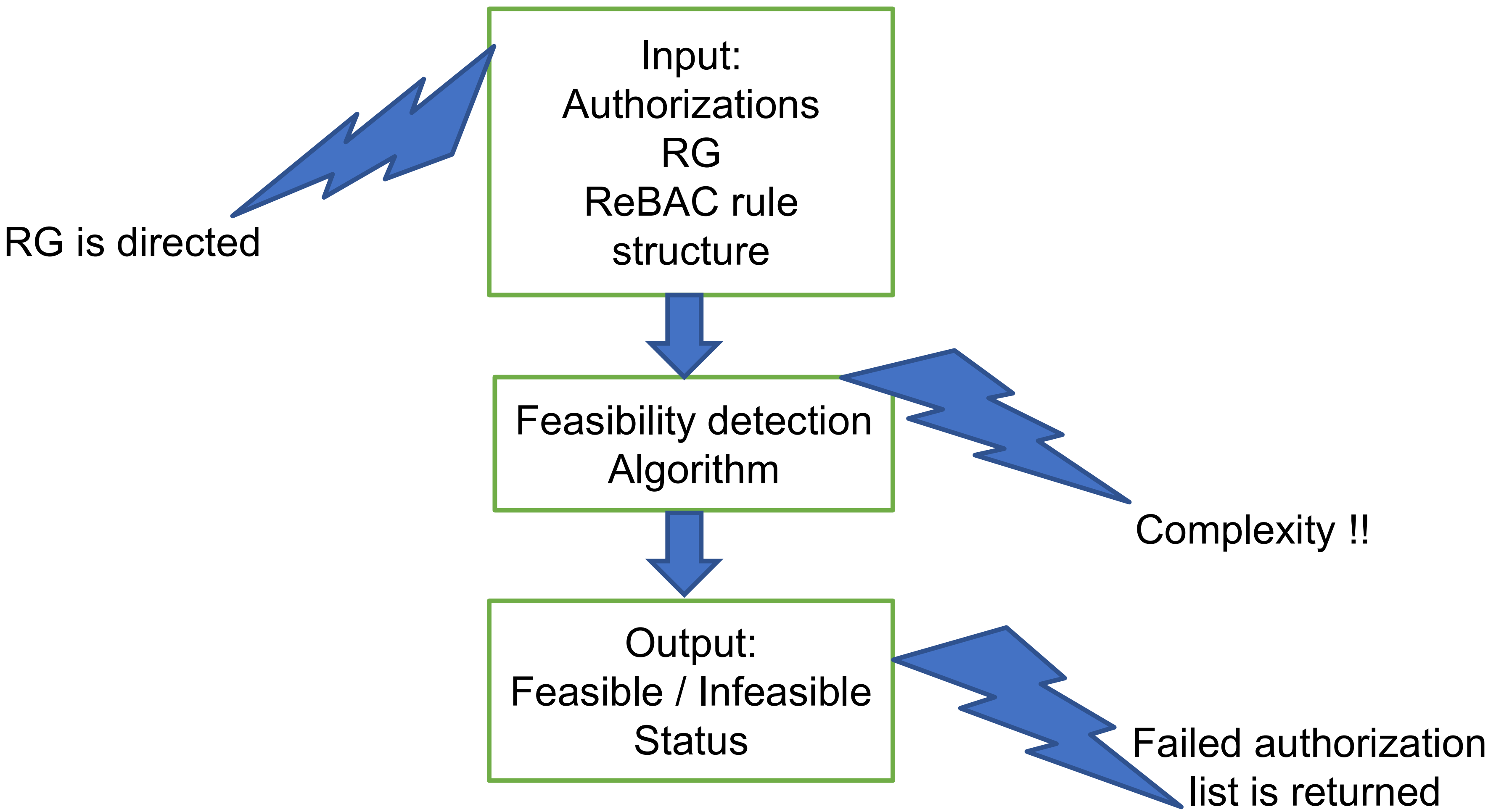
- <u>Solution is guaranteed even if inconsistency arises</u>
  - <u>Infeasibility problem</u>

ReBAC Rule Structure

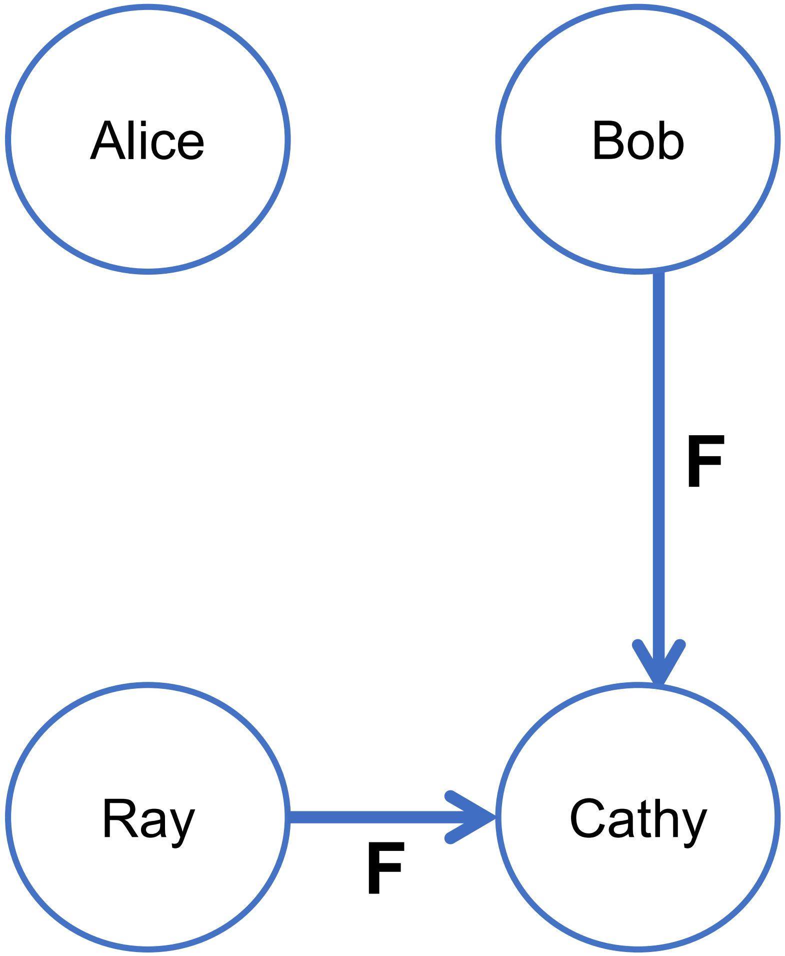
$$Rule_{op} ::= Rule_{op} \vee Rule_{op} \mid pathRuleExpr$$
$$pathRuleExpr ::= pathRuleExpr \wedge pathRuleExpr \mid pathLabelExpr$$
$$pathLabelExpr ::= pathLabelExpr.pathLabelExpr \mid edgeLabel$$
$$edgeLabel ::= \sigma, \sigma \in \Sigma$$

- Evaluation of access request (a, b, op)
  - for each pathLabelExpr in $Rule_{op}$ substitute True if there exists a simple path p from a to b in RG with path label pathLabelExpr, otherwise substitute False
  - the resulting boolean expression evalutes true → grant, deny otherwise

Feasibility Detection



RG Example



Feasible

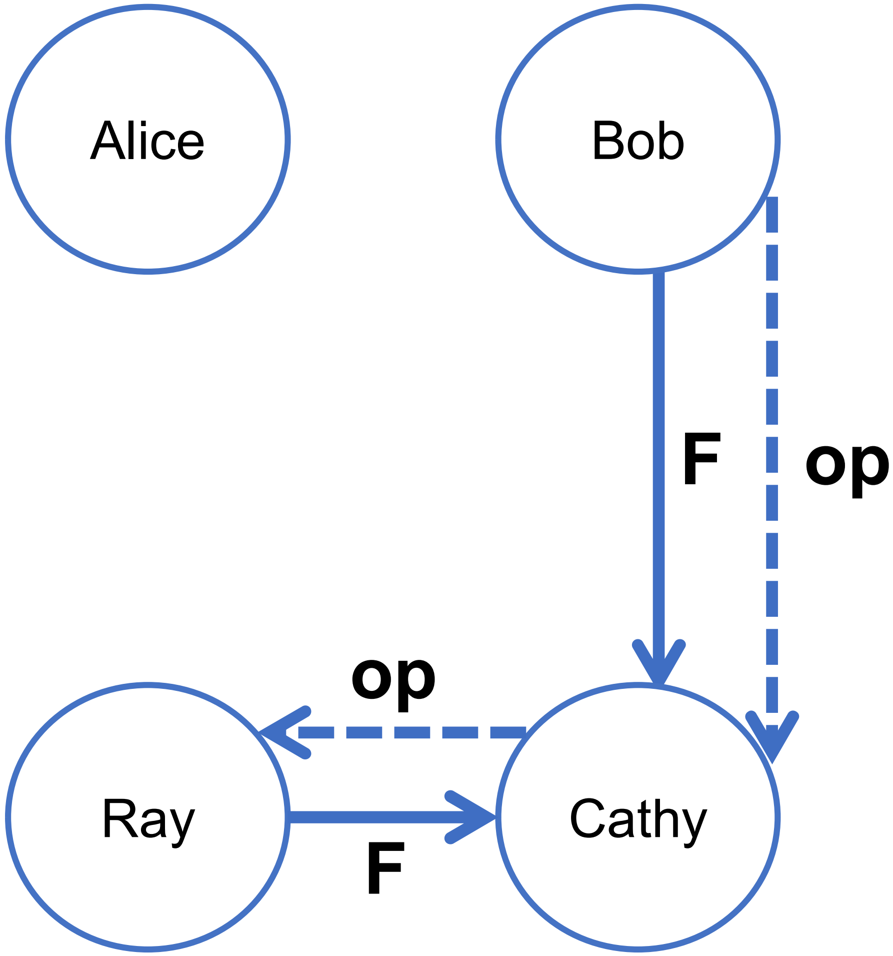
(Bob, Cathy, op)
(Ray, Cathy, op)

Rule_{op} = F

Infeasible

i) (Bob, Cathy, op)
ii) (Cathy, Ray, op)

Solution 1

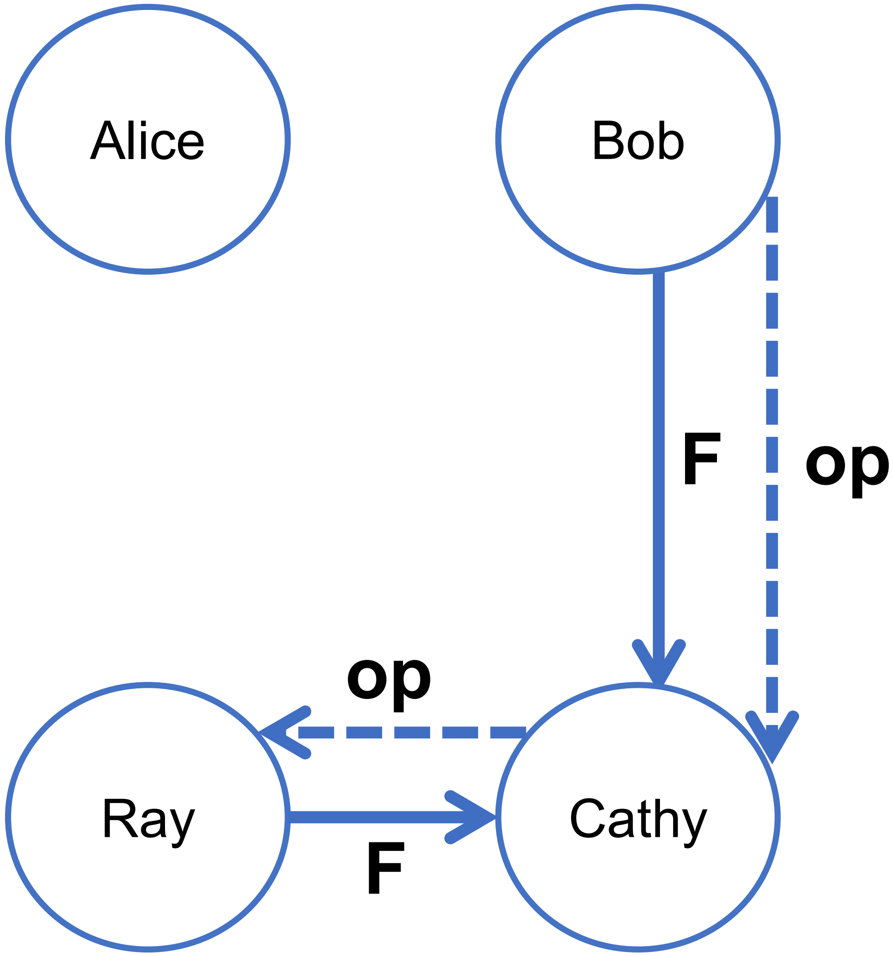


Infeasible

i) **(Bob, Cathy, op)**
ii) **(Cathy, Ray, op)**

Rule$_{op}$ = op

Solution 1



Infeasible

i) **(Bob, Cathy, op)**
ii) **(Cathy, Ray, op)**

Rule$_{op}$ = op

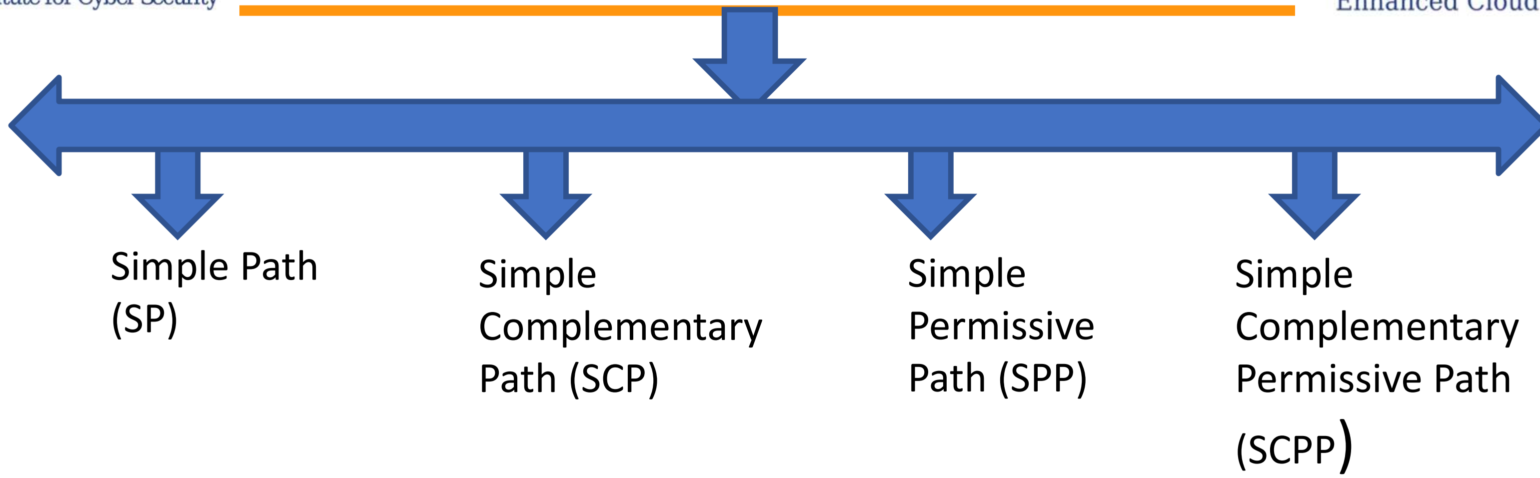
Simple

Operation ∩ Relationship types={}

Minimal edges not guaranteed

|Authorization| edges at worst!

Path Variations

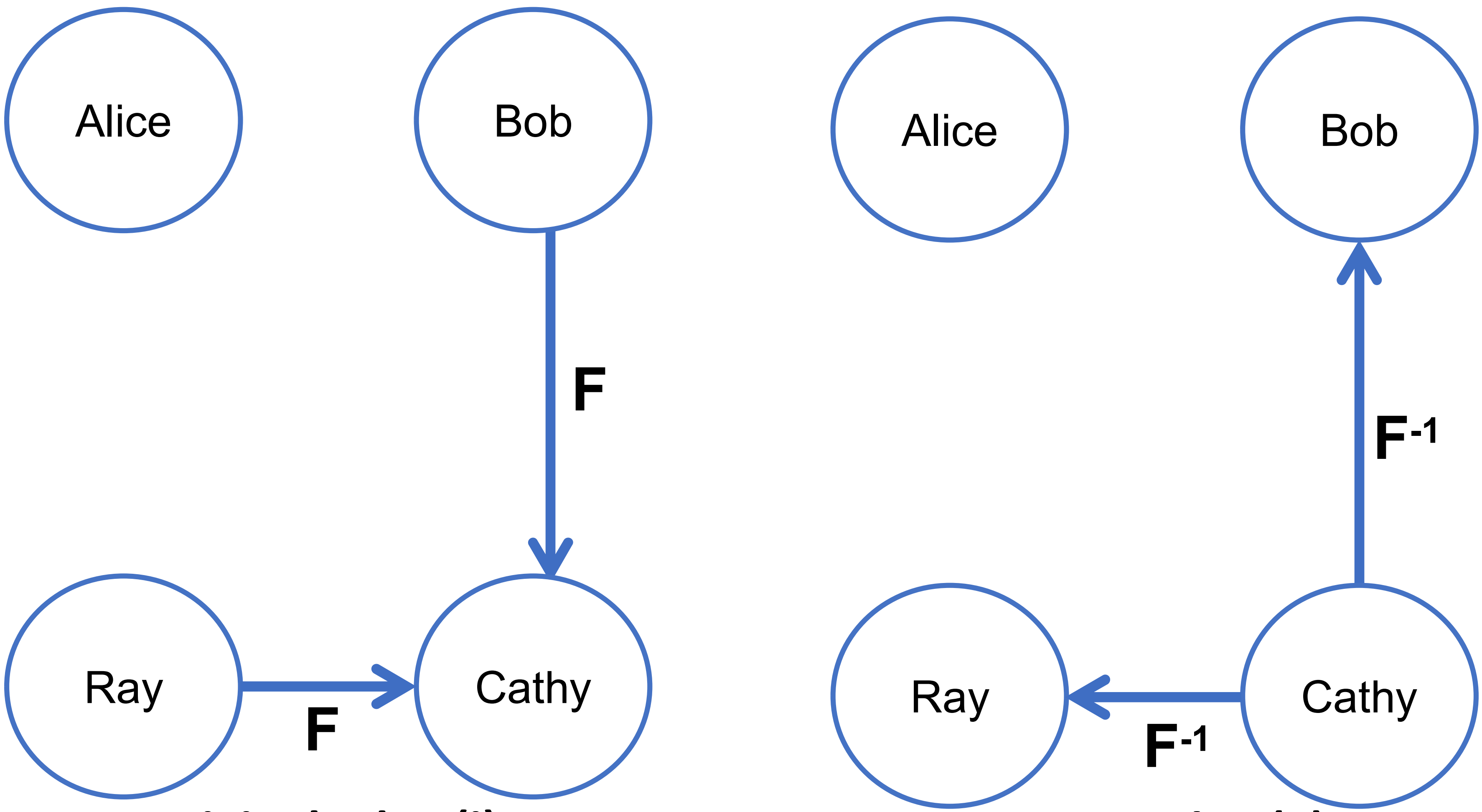


Characteristics	SCP	SPP	SCPP
$(a,b,\sigma) \rightarrow (a,b,\sigma) \in E, \sigma \in \Sigma$	✓	✓	✓
$(a,b,\overline{\sigma}) \rightarrow (a,b,\sigma) \notin E, \overline{\sigma} \in \overline{\Sigma}$	✓		✓
$(a,b,\sigma^{-1}) \rightarrow (b,a,\sigma) \in E, \sigma^{-1} \in \Sigma^{-1}$		✓	✓
$(a,b,\overline{\sigma}^{-1}) \rightarrow (b,a,\sigma) \notin E, \overline{\sigma}^{-1} \in \overline{\Sigma}^{-1}$			✓

Table represents path variations with original, non-relationship, inverse and non-relationship inverse edges (row 1, 2, 3, and 4, respectively).

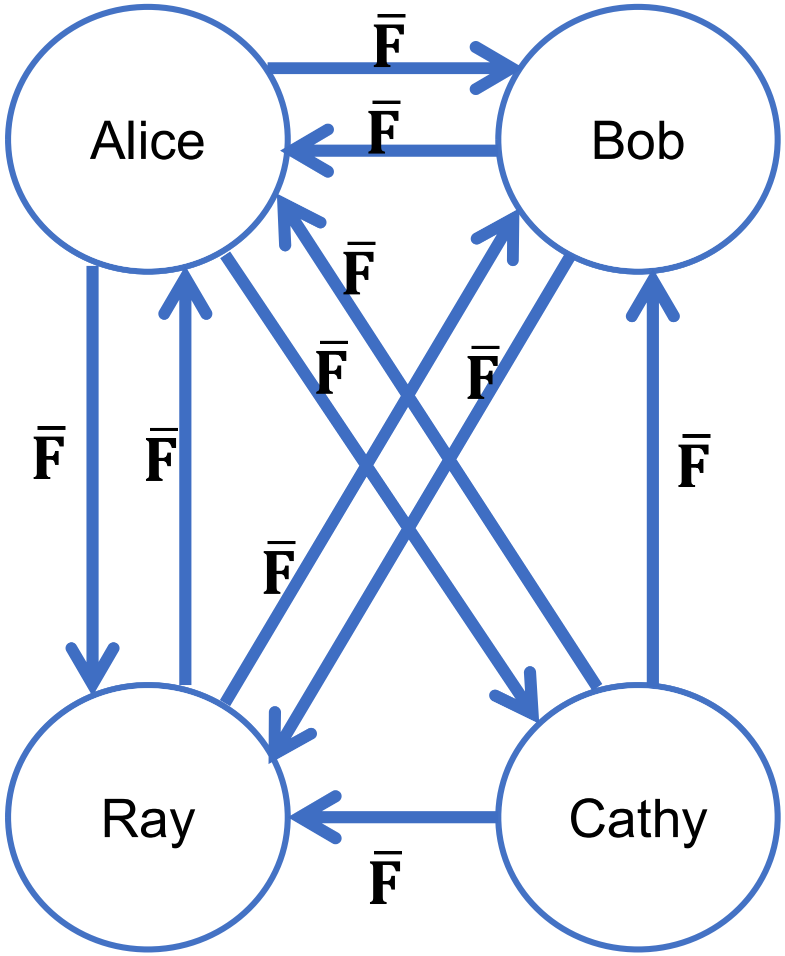
- a,b: users, E and ∑ are the sets of edges and relationship type specifiers

Path Variations Cont.

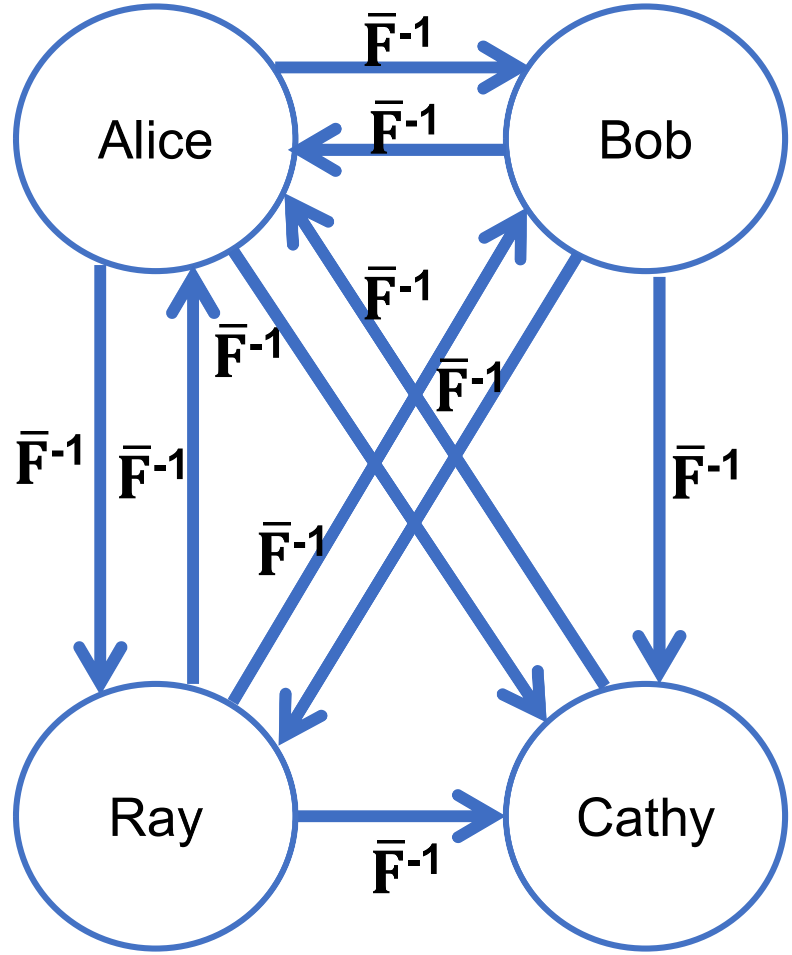


Original edge (i) **Inverse edge (ii)**

Path Variations Cont.



Non-relationship edge (iii)



Non-relationship inverse edge (iv)

RREP Variations

RREP-0 → SP (i)

RREP-1 → SCP (i + iii)

RREP-2 → SPP (i + ii)

RREP-3 → SCPP (i + ii + iii + iv)

Rule minimization techniques are described in the paper

Challenges

- ❖ Complexity is exponential
- ❖ Inexact solution
- ❖ More path variations
- ❖ Cope up with changes in rule structures
- ❖ Other infeasibility solutions
- ❖ Extend beyond user-user context

Chapter 5

Background: AReBAC

❖ *AReBAC ≡ Attribute-aware ReBAC*

- Integrate attribute information with ReBAC
- Makes policy generation more flexible and convenient
- Attribute-aware Relationship Graph (ARG)

Assumption

- ARG where users(node) are connected(edge) where user and edge have attributes
- Each user and edge have corresponding user and edge attribute values, respectively
- Only user-to-user relationships are considered

Problem Statement

The feasibility analysis of the AReBAC policy mining problem studies whether the migration process from a given authorization set to AReBAC policy is feasible or not under the set of imposed criteria:

- Attribute-aware Relationship Graph (ARG) is given
- AReBAC rule structure is given
- <u>Use of entity ID is not allowed</u>
    - <u>Existing literature allows ID</u>
- Equivalent set of AReBAC rules are required

- <u>Solution is guaranteed even if inconsistency arises</u>
    - <u>Infeasibility problem</u>

AReBAC Rule Structure

$Rule_{op} ::= Rule_{op} \vee Rule_{op}$ | $pathRuleExpr$ | Attexp

$pathRuleExpr ::= pathRuleExpr \wedge pathRuleExpr$ | $(pathLabelExpr)$

$pathLabelExpr ::= pathLabelExpr.pathLabelExpr$ | $edge$Expr

Attexp ::= Attexp ∧ Attexp | uexp = value | vexp = value

edgeExp ::= edgeExp ∧ edgeExp | edgeuexp = value | edgevexp = value | edgeattexp = value

- Evaluation of access request (a, b, op)
  - Checks with user attribute values of a and b
  - If there exists simple path from a to b in ARG, Checks with them too!
  - The resulting boolean expression evalutes to true → grant, deny otherwise

Feasibility Detection

Input:
Authorizations
ARG
AReBAC rule structure

ARG is directed

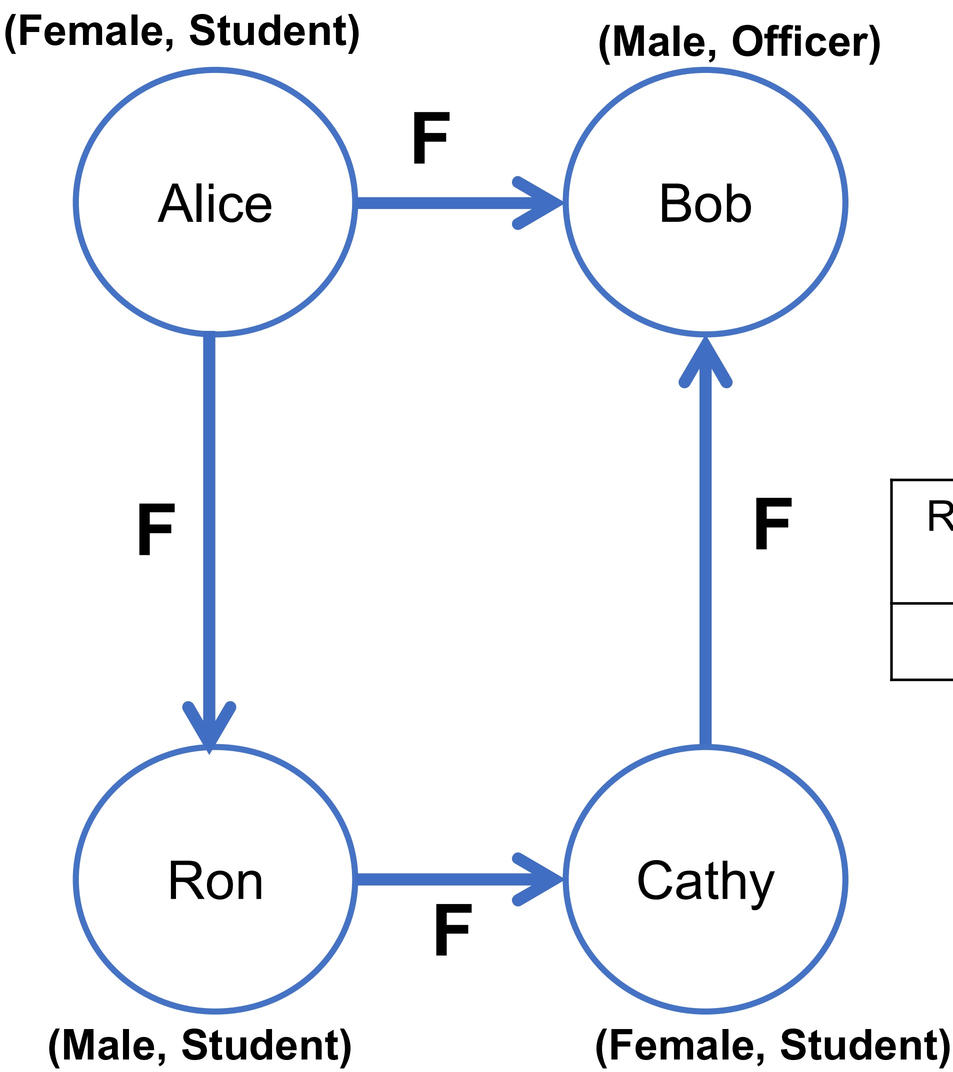
Feasibility detection Algorithm

Complexity !!

Output:
Feasible / Infeasible Status

Failed authorization list is returned

ARG Example

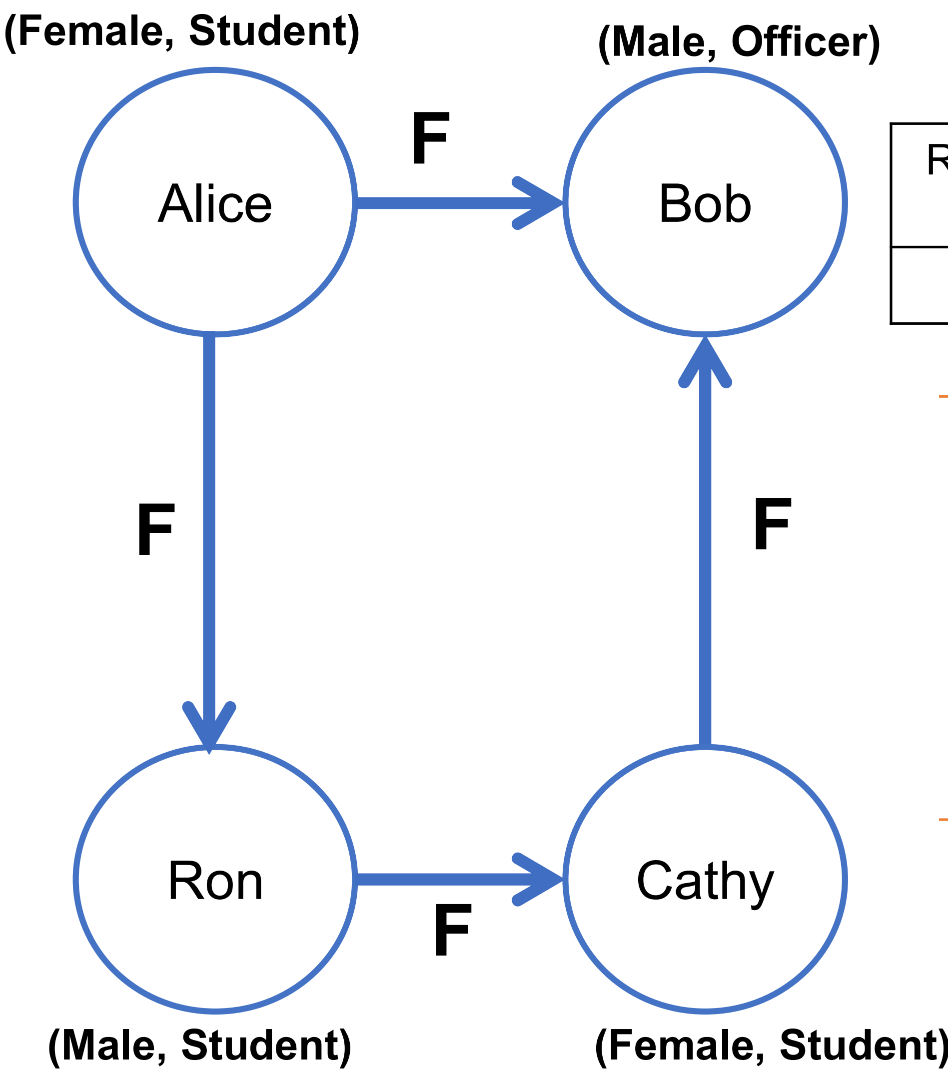


UA = {Gender, Profession}
EA = {Relation-type}

ReBAC	ABAC	AReBAC	AUTH
✗	✗	✓	(Alice, Ron, op)

Feasible

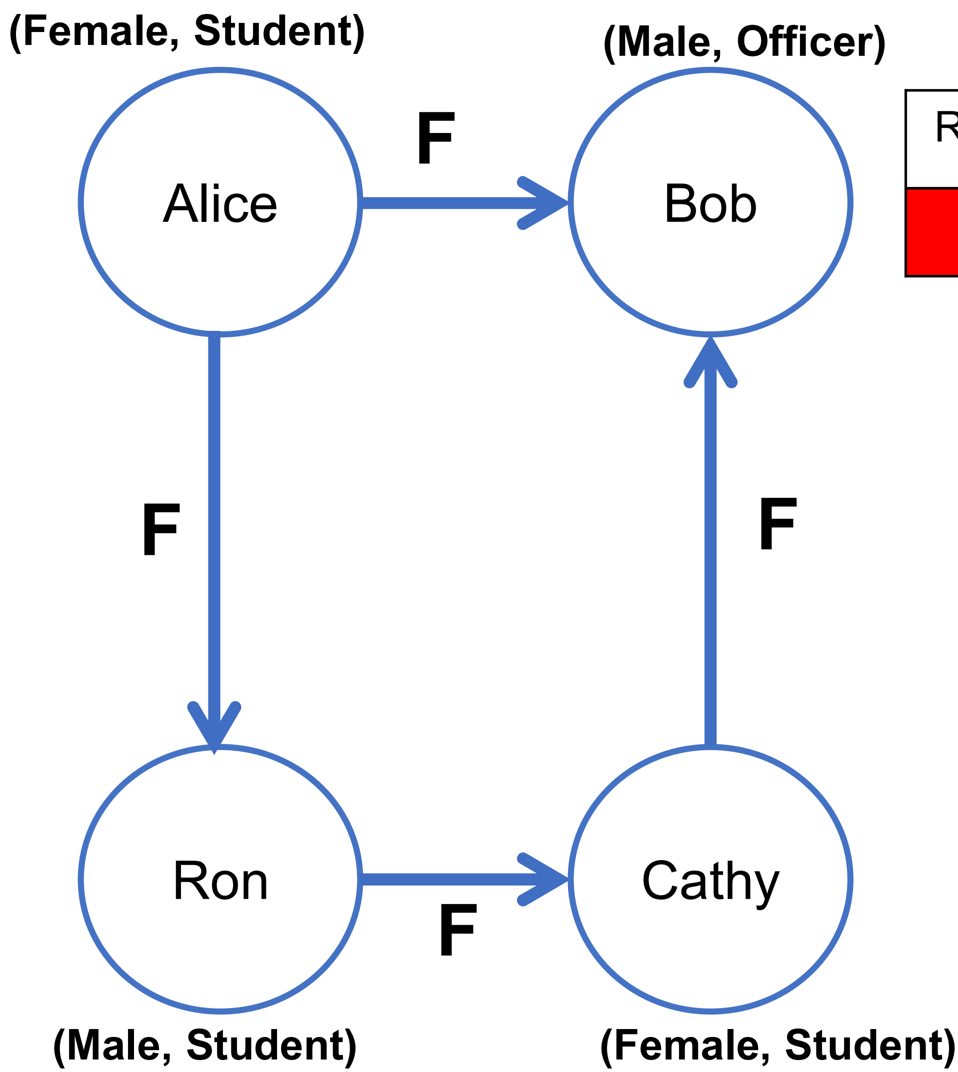
ARG Example



ReBAC	ABAC	AReBAC	AUTH
×	×	✓	(Alice, Ron, op)

$Rule_{op}$ = (Gender(e.u) = Female ∧ Profession(e.u) = Student ∧ Relation-type(e) = F ∧ Gender(e.v) = Male ∧ Profession(e.v) = Student)

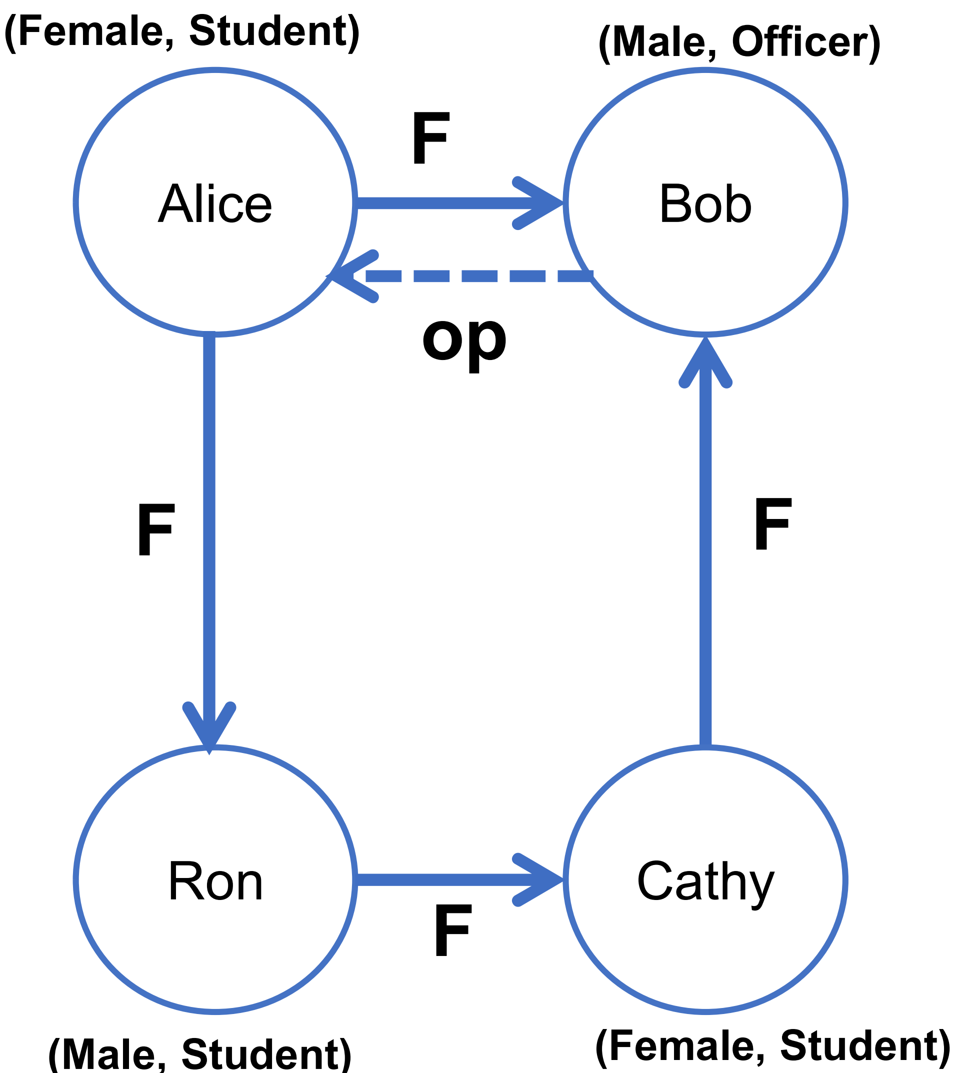
ARG Example



ReBAC	ABAC	AReBAC	AUTH
✗	✗	✗	(Bob, Alice, op)

Infeasible

Infeasibility Solution



Infeasible

(Bob, Alice, op)

Rule_{op} = (Relation-type(e) = op)

Simple

Minimal edges not guaranteed

|Authorization| edges at worst!

Approximate solution

Summary

- Complexity is exponential
- Inexact solution
- Path variations can be used
- Cope up with changes in rule structures
- Path with cycle
- Other infeasibility solutions
- Extend beyond user-user context

Chapter 6

Extended ReBAC RuleSet Existence Problem (ERREP)
Extended ABAC RuleSet Existence Problem (EAREP)

ERREP

EAS + ReBAC System
(Identical user set)

AUTH(EAS) ≠ AUTH(ReBAC)

ERREP

ERREP-0
*AUTH(EAS) ⊂ AUTH(ReBAC)

ERREP-1
*AUTH(ReBAC) ⊂ AUTH(EAS)

ERREP-2
*AUTH(EAS) ∩ AUTH(ReBAC)≠Ø

AUTH(EAS) and AUTH(ReBAC) denote the authorizations allowed by EAS and ReBAC system, respectively

Exact and approximate solutions are presented

EAREP

EAS + ABAC System (Identical user and object sets)

AUTH(EAS) ≠ AUTH(ABAC)

EAREP

EAREP-0

*AUTH(EAS) ⊂ AUTH(ABAC)

EAREP-1

*AUTH(ABAC) ⊂ AUTH(EAS)

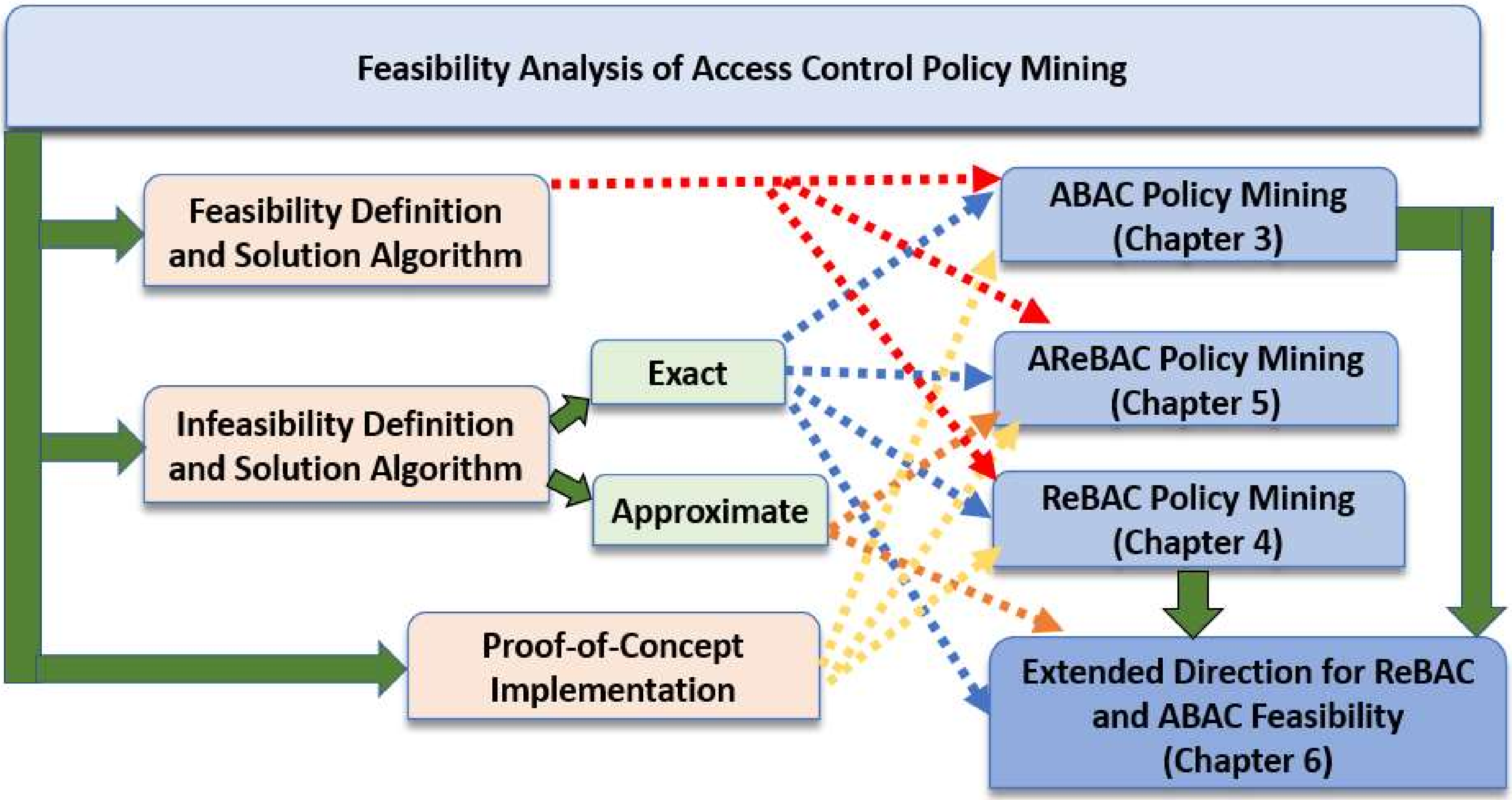
EAREP-2

*AUTH(EAS) ∩ AUTH(ABAC)≠∅

AUTH(EAS) and AUTH(ABAC) denote the authorizations allowed by EAS and ABAC system, respectively

Exact and approximate solutions are presented

Conclusion and Future Work



Publications

1. Shuvra Chakraborty and Ravi Sandhu, On Feasibility of Attribute-Aware Relationship-Based Access Control Policy Mining. In Proc. 33rd Annual IFIP WG 11.3 Working Conference on Data and Applications Security and Privacy (DBSec), Virtual Event, July 19-20, 2021. [Published]

2. Shuvra Chakraborty and Ravi Sandhu, Formal Analysis of ReBAC Policy Mining Feasibility. In Proceedings of the 11th ACM Conference on Data and Application Security and Privacy (CODASPY), Virtual Event, April 26-28, 2021. [Published]

3. Shuvra Chakraborty, Ravi Sandhu and Ram Krishnan, On the Feasibility of RBAC to ABAC Policy Mining: A Formal Analysis. In Proceedings of the 8th International Conference on Secure Knowledge Management in Artificial Intelligence Era (SKM), Goa, India, December 21-22, 2019. [Published]

Publications

4. Shuvra Chakraborty, Ravi Sandhu and Ram Krishnan, On the Feasibility of Attribute-Based Access Control Policy Mining. In Proceedings of the 20th IEEE Conference on Information Reuse and Integration (IRI), Los Angeles, California, July 30-August 1, 2019. [Published]

5. Extending the Feasibility of Relationship and Attribute-Based Access Control Policy Mining [To be submitted soon]

Posters

1. Shuvra Chakraborty, Ram Krishnan, and Ravi Sandhu. *Attribute-Based Access Control Policy Mining Problem: Thinking Differently.* Women in CyberSecurity, March 2019, Pittsburgh, PA. **[Presented]**

2. Shuvra Chakraborty and Ravi Sandhu. *Exploring Feasibility Problem in Access Control Policy Mining Domain*. Women in CyberSecurity, September 2021, Denver, CO. **[Presented]**

I·C·S
The Institute for Cyber Security

C·SPECC
Center for Security and Privacy
Enhanced Cloud Computing

Thank you

Questions?