IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 5 January 2023, accepted 19 January 2023, date of publication 6 February 2023, date of current version 14 February 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3242863

SURVEY

Effective Collaboration in the Management of Access Control Policies: A Survey of Tools

RACHAEL FERNANDEZ[1], PETER C.-H. CHENG[2], ARMSTRONG NHLABATSI[1], KHALED MD. KHAN[1], (Senior Member, IEEE), AND NOORA FETAIS[1], (Senior Member, IEEE)
[1]Qatar University, Doha, Qatar
[2]University of Sussex, BN1 9RH Brighton, U.K.

Corresponding author: Noora Fetais (n.almarri@qu.edu.qa)

This work was supported by the National Priorities Research Program (NPRP), Qatar National Research Fund (a member of Qatar Foundation), under Grant 12S-0115-190002. The publication of this article was funded by Qatar National Library.

ABSTRACT Access control (AC) tools implement security policies for controlling access to various assets, including file systems, physical resources, and social media posts. They are also used as pedagogical tools for exploring and understanding intricate details of complex security policies. However, current tools are not developed based on the actual needs of security and policy professionals. They are not equipped to support basic and vital operations like providing a policy overview, policy comparisons, identifying and resolving policy conflicts. In this paper, we explore (a) the specific challenges faced in the collaboration between access control policy makers and implementers, and (b) the limitations that current tools have towards addressing these challenges. We argue that a lack of effective collaboration between policy makers and implementers may lead to a misunderstanding of security policy semantics. The main reason for this problem is that policy makers and implementers use different technical languages for communication. The lack of a common technical language leads to a miscommunication between the two parties. The key aim of our work is to review the currently available research-based access control tools and to identify their pros and cons. To accomplish this, we have reviewed a set of access control tools that have a wide variety of features and applications. We have also identified a set of tasks that these access control tools possess to help the work of policy professionals who are involved in the creation, management and maintenance of security policies. We also compared the functionalities of these tools, the different types of security policies that they support, and their visualizations. Together, these comparisons provide a clear understanding of what current access control systems lack and how they can be improved in order to support effective collaboration between policy makers and policy implementers. We have also found that many of these tools could be more accessible to non-technical policy professionals to understand the semantics of security policies if these tools provide features for visualizing security policies.

INDEX TERMS Access control, security policy, information visualization.

I. INTRODUCTION

The successful specification and implementation of access control are critical to upholding the security practices and policies of any organisation. Companies are under constant threat of security breaches from not only hackers but also from insider attacks as employees may compromise the availability, integrity and confidentiality of information assets [1]. While malicious intentions of the employees may lead to purposeful compromise of an organisation's security, a lack of awareness of the employees is another reason that may unintentionally compromise the security of the information assets. Several security breaches have resulted from a lack of understanding of the semantics of access control policies and the implications of policy settings that protect information assets by the employees [2]. Security

The associate editor coordinating the review of this manuscript and approving it for publication was Petros Nicopolitidis.

policies not easily understood by average users make it hard for them to grasp the implications of security settings even when they are merely setting security rules for social networking sites [3]. Lam and Churchill [4] investigated how users assign access controls to their photos on the photo-sharing website Flickr. They found that only 20% of the users who participated in the study had changed their default privacy settings from ''public'' and assigned some permissions on their photo collections. Ahern et al. [5] also studied the use of privacy settings in another mobile photo-sharing application and found that users were often granted more access than they intended to. The complexity of security policies makes it hard for users to understand the implications of their security settings. Visualization of these complex policies might aid users in understanding these policies.

The cognitive effort required to understand the complexity of access control policies is greater than what policy professionals can manage [2]. Needless to say that this cognitive effort is much greater for average users. The complexity of access control policies range from simple to highly complex based on the number of rules and constraints that are added to the policy. A simple AC policy might grant a user access to a particular file or folder using a single or a small number of access control rules. In general, an access control policy is made up of multiple *access control rules* which govern access to various electronic resources. A complex policy can be made up of several AC rules with various constraints for accessing resources. For example, the simple case of allowing a user to access a file can be made complex by enabling the user to access a file only after entering a password. If the resource is more important, then additional restrictions mandating that the password should be changed every 90 days, that the user would be logged out after 30 minutes of inactivity, or that the privileges the user has on the file will be varied depending on his or her current geographic location, or the time of day, can also be added. This version of the policy is more complex and would be made up of several AC rules.

Access control tools help to overcome these difficulties by supporting the process of *creating, implementing* and *managing* policies that make it easier for end-users and security professionals. These tools help the users understand complex policies by breaking them down in a simple manner. Some tools provide visualizations features to describe the structure of policies. Though there are various access control tools, Bauer et al. [6] identified that these tools are neither based on the actual tasks performed by policy professionals nor equipped to solve the issues faced by them. Bauer et al. were the first to explicitly recognise that two sets of policy professionals are involved in creating and implementing policies - *policy makers* who are responsible for creating the rationale of the policies; and *policy implementers* who are responsible for implementing the policies in the system. Bauer et al. also identified the problems faced by these policy professionals. On the other hand, the ADAGE tool [7] was the first to identify that policy authoring is a collaborative task. Still, it did not acknowledge that the two sets of professionals are required to create and implement policies. In particular, our work explores the specific challenges faced in the collaboration between access control policy makers and implementers, and the limitations that current tools have towards addressing these challenges. We argue that challenges of collaboration between policy makers and implementers lead to a misunderstanding of security policy semantics. A key challenge is that policy makers and implementers use different technical languages for communication. A lack of a common technical language leads to a miscommunication between the two parties. By technical language, we mean the different means of communication that the policy makers and implementers use in terms of vocabulary and context to create and manipulate policies when working collaboratively. For example, policy makers look at policies from a business perspective by identifying the need for the policy, identifying the people who will be affected by the policy and how they will be affected, followed by securing the approval to enforce the policy and finally communicating the new policy to the employees. On the other hand, the policy implementers think about which access control policy language can be used to write policies in some formal computational notation, the constraints that can be applied, and how new policies can coexist with other implementations. Hence, the key aim of our work is to identify how policy makers and implementers can use visualizations to facilitate effective collaboration. We refer to the problem of effective communication between policy makers and implementers as the *policy collaboration problem*, and this is the main focus of our work.

In addition to identifying the day-to-day tasks of the policy professionals, Bauer et al. [8] also recognised the issues faced by policy makers and implementers. Some of the issues faced by policy makers include the inability to view, edit and verify the implemented policy as they are not technically skilled.On the other hand, policy implementers cannot detect semantic inconsistencies in the policy due to a lack of knowledge about the rationale behind the formulation of the policies they implement. They also face difficulties in tracking and revoking privileges that are granted as exceptions to policies. For example, consider an employee A is granted additional privileges as exceptions, in addition to her default privileges associated with her role. When A finally leaves this role and moves to another role, it is difficult for the policy implementers to identify and revoke all her other privileges that were granted as exceptions. The problem could become more complicated when many such examples of employee A are considered in an organisation. Both sets of policy professionals also face communicative and collaborative issues among them. It is difficult for policy implementers to keep track of and understand the changes made by policy makers. Similarly, policy makers also find it challenging to keep track of the changes made by other policy makers. Policy documentation to keep track of changes is also challenging. The lack of a common technical language between policy makers and implementers make it hard to keep track of changes to the policy. We content that a common technical language between the two sets of policy professionals can bridge the communication gap between them and facilitate better understanding of the implemented policy for the policy makers, while simultaneously making it easier for policy implementers to understand the rationale of the policy that they are implementing.

We surveyed several research-based access control tools that security professionals use for managing access to resources in organisations in order to investigate the extent to which these tools address the challenges of policy making, collaboration, and implementation described above. A total of 31 research-based access control tools have been reviewed. This helped us identify the pros and cons of the tools developed by these tools. We noted that these tools failed to acknowledge the AC policy collaboration problem. Recognising these limitations helped us explore visualizations as a potential solution to bridge the communication gap between policy makers and implementers. Of the 31 tools we reviewed, only one tool interviewed security administrators to understand their daily tasks before developing their tool. We also identified different resources that these tools protect with associated policies. We also compared the visualizations that these tools used and the user studies that were conducted to evaluate these tools.

The main contributions of this paper are as follows:

- *Policy Collaboration Framework*: A framework that identifies and frames access control policy issues that need to be addressed for *policy making*, *policy implementation*, and *effective collaboration* between policy makers and policy implementers.
- *Systematic Mapping of Policy Collaboration Issues*: A mapping of policy collaboration issues to common features of tools for the purpose of identifying the limitations of current access control tools toward addressing policy collaboration issues.
- *Evaluation*: An assessment of how well do current access control tool support activities for policy makers, policy implementers, and their collaboration in formulating and implementing access control policies.
- *Limitations Identification*: Identification of the limitations of current access control tools towards supporting policy collaboration activities and the paving of a research agenda for access control policy collaboration.

The rest of the paper is organized as follows. Section II describes the collaboration problems that arise during policy creation, editing and implementation. The scope of this work and the steps we followed to conduct the tool are discussed in Section III. Following this, we introduce a comprehensive set of operations that the tools support, along with a comparison of the operations in Section IV. In Sections V and VI we compare the reviewed tools based on their visualization techniques and the user studies they conducted to evaluate their tools, respectively. We then discuss the research trends and shortcoming of AC tools in Section VII. We finally conclude the paper in Section VIII.

II. ACCESS CONTROL POLICY COLLABORATION PROBLEM FRAMEWORK

A. POLICY MAKERS AND IMPLEMENTERS

Due to complex security policies, multiple stakeholders are responsible for creating, implementing and managing security policies. Bauer et al. [8] conducted a series of 11 interviews with 13 policy professionals in 5 organisations who worked on either controlling access to files or physical spaces. They recognise that it requires two sets of professionals for *creating, implementing* and *maintaining* policies. Two of the key findings from the Bauer et al. study include:

1) Two types of policy professionals are responsible for creating and implementing a policy, namely: *Policy Makers* and *Policy Implementers*.
    a) **Policy makers (PM) or architects** are responsible for creating and verifying the implementation of policies. They create several policies usually after discussing with stakeholders of the organisation. These policies are typically called *proposed policies* as they just capture the intentions of the policy makers and have not been implemented in the system yet. The policy makers typically belong to the management or the business side of the organisation and are not usually technically skilled. As a result, they don't have much knowledge on how to implement their policies on a system.
    b) **Policy implementers (PI)** are responsible for interpreting the policies created by the policy makers and then implementing them. They receive the *"proposed policies"* from the policy makers and implement them, thereby transforming the *proposed policies* to *operational policies*. They are technical experts in security. Implementers may also need to verify with policy makers if their implemented policies are consistent with their intentions. The general differences in the skill set and the tasks required of each set of these professionals are elaborated in Table 1.
2) Once a policy is implemented, they are maintained by multiple policy implementers. A policy might be edited by a policy implementer and a log of the changes is maintained.

PMs and PIs work together to create, edit and review policies. PMs usually initiate the creation of policies after either identifying a business need or because the permission to a resource needs to be altered. PMs usually record their intentions and send them to policy implementers. PIs then

TABLE 1. Differences between a policy maker and policy implementer.

PM	PI
Creates policy (intentions)	Implements the policy in the system
Has full knowledge of the policy	Only possesses working knowledge of the policy
Does not know how to view or edit the implemented policy	Can perform all the functionalities in the system
Knows WHAT to edit	Knows HOW to edit but does not know WHAT to edit
Responsible for verifying the implementations of the policy	Send policies periodically to the PM for verification

make them operational by transforming them into access control policy of the system. Tools can be used to check for policy conflicts or other constraints.

Policies are also edited time to time by the implementers to accommodate changes to the policy. Sometimes policy makers might not be aware of the changes that the policy has gone through. Policy implementers periodically consult with the policy makers to verify the consistency of the implemented changes. According to Bauer et al. [8], the periodicity of the verification vary from organisation to organisation. Some policy implementers consult with the policy makers regarding the edited policies on a monthly basis, while others may only do it once a year for approval.

B. THE POLICY COLLABORATION PROBLEM

The limitations of existing AC tools is the genesis of the *policy collaboration problem* - a set of problems/issues for policy collaboration. The problems arising due to collaboration between policy makers and implementers are shown in Fig. 1. The figure shows that a key source of the collaboration challenges between policy makers and implementers is misunderstanding which is a consequence of their use of different technical languages. Policy makers and implementers use PM_L and PI_L, respectively, as their technical languages. The inspiration for Fig. 1 came from three sources of knowledge: (a) literature, (b) informal interviews, and (c) personal experience. The challenges of collaboration between PMs and PIs are suggested in the literature [8], [9]. However, none of these works have analysed the challenges in detail or proposed a systematic approach towards addressing them. In order to get a better understanding of the practical challenges encountered by professionals, we conducted informal interviews with security professionals. This enriched our insight into the issues of collaboration between policy makers and implementers. The challenges between policy makers and implementers is akin to the issues faced in a software engineering process [10], where a client is responsible for stating their needs to a software engineer. These requirements are then translated into specifications of the system that is to be developed. The software developers eventually write the program code that will produce the behaviour stated in the specifications.

In this analogy, the formulations, definitions, and specifications of policies are equivalent to requirements for a software system. In the field of software engineering, the *elicitation* and *analysis* of software requirements is the responsibility of *Requirements Engineers* - software engineers are specialized in analysing and understanding users' needs. Requirements Engineers liaise with the *owners* of the *problem* to be solved by a software system in order to gain a better understanding of the *nature* and *context* of the problem. However, their understanding of the problem to be solved is often vague. One of the most common methods Requirements Engineers often use in eliciting requirements is to conduct interview with the problem owners to get a better understanding of the problem to be addressed. The understanding gives them insight that helps in better formulating the requirements of the software system. The same applies in the field of access control - the first step is to understand what security objective(s) of *assets* are to be protected against what *threats*? For example, in a payroll system, the salaries of employees in a company can be considered confidential. The formulation of an access control policy to protect the confidentiality (a security objective) of salaries (an asset) would require deciding who should have access to salary data and what level of access or privileges should they have? Can they *modify* the data or they can only *view* it?

We propose three categories of collaboration challenges as illustrated in Fig. 1, namely (1) *Policy Making*, (2) *Collaboration*, and (3) *Policy Implementation*. The policy issues are enclosed in the dotted round-corner rectangles. The dotted "document" icon represents artefacts such as documents where requirements are written or source code implementing policies. The cuboid represents a task to be performed by either policy makers or policy implementers. The dotted line with a diamond at the end represents the association of a policy issue with the entity connected at the end of the line. Security policies are initiated by policy makers who specify policy requirements such as *"A student should be able to see their quiz grade in an online course assessment system but they should not be able to modify the grade."* Such a requirement could be stated by the university's curriculum development committee.

C. POLICY MAKING ISSUES

Policy making issues include *Misunderstanding*, *Miscommunication*, *Requirement Update Documentation*, *Requirement Incompleteness*, *Requirement Redundancy* and *Requirement Inconsistency*. *Misunderstanding* and *Miscommunication* between policy makers may arise when policies requirements are being updated by different policy makers. Such issues of misunderstanding may result in from a lack of procedures to be followed when updating policy requires information on

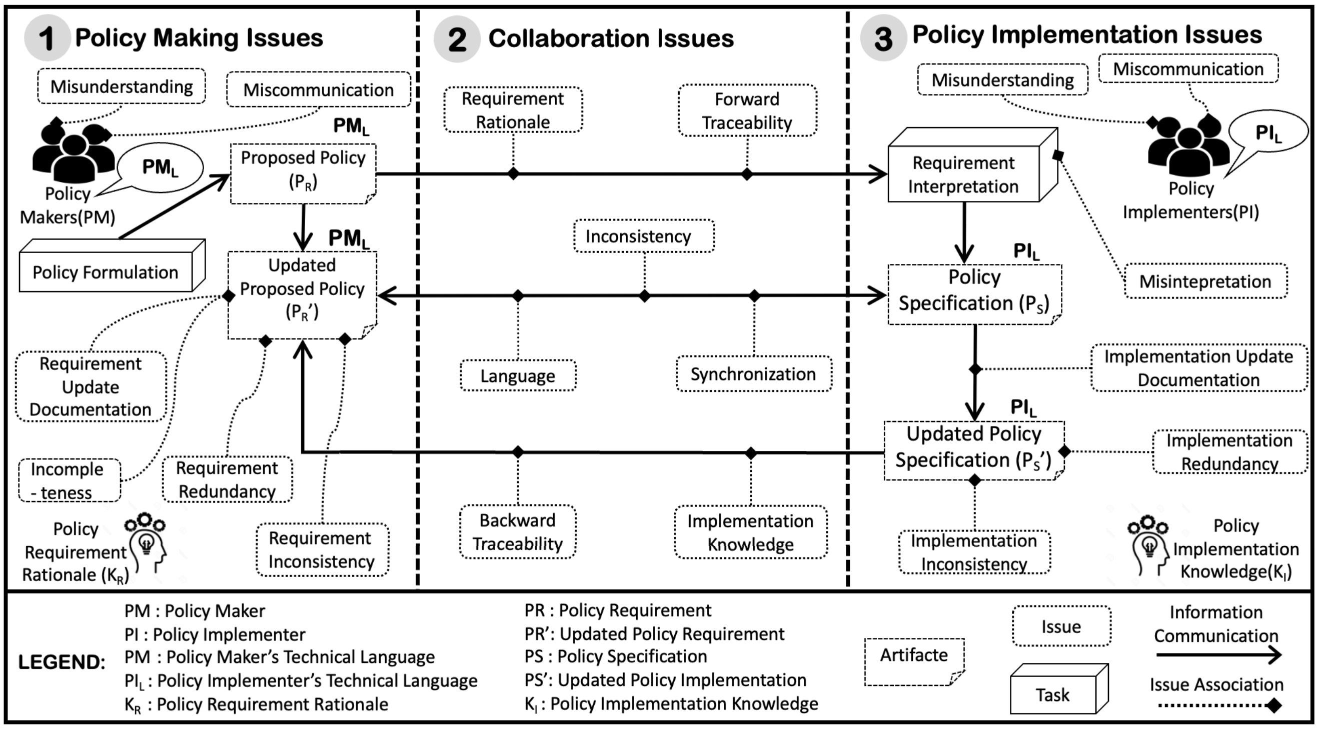

1 Policy Making Issues
2 Collaboration Issues
3 Policy Implementation Issues
Misunderstanding
Miscommunication
PM_L
Policy Makers(PM)
Proposed Policy (P_R)
Policy Formulation
Updated Proposed Policy (P_R')
Requirement Update Documentation
Incomple - teness
Requirement Redundancy
Policy Requirement Rationale (K_R)
Requirement Inconsistency
Requirement Rationale
Forward Traceability
Inconsistency
Language
Synchronization
Backward Traceability
Implementation Knowledge
Requirement Interpretation
PI_L
Policy Implementers(PI)
Policy Specification (P_S)
Misintepretation
Implementation Update Documentation
Updated Policy Specification (P_S')
Implementation Redundancy
Implementation Inconsistency
Policy Implementation Knowledge(K_I)
LEGEND:
PM : Policy Maker
PI : Policy Implementer
PM_L : Policy Maker's Technical Language
PI_L : Policy Implementer's Technical Language
K_R : Policy Requirement Rationale
PR : Policy Requirement
PR': Updated Policy Requirement
PS : Policy Specification
PS': Updated Policy Implementation
K_I : Policy Implementation Knowledge
Artifact
Issue
Task
Information Communication
Issue Association

FIGURE 1. Collaboration challenges between policy makers and implementers.

how updates to the requirement should be communicated to other members in the policy making teams.

The process of *Policy Formulation* results into the elicitation of policy requirements. The *Requirement Update Documentation* problem arises when a policy requirement, P_R, is being updated to a new requirement, P'_R, without proper and sufficient documentation of the changes made during the update. Such lack of documentation may result in different members of the policy implementers team making inconsistent updates to a policy without knowing what updates have been made by other team members. The *incompleteness* problem arises when some key assumptions have been either been omitted or misunderstood during the formulation of a policy requirement. The omission of these assumptions results in a requirement that is incorrect, and may eventually results in an incorrect policy not able to offer sufficient protection to assets. For example, an access control policy states that authentication to access sensitive information in a database should only be based on a password which is known only to the subject i.e. the owner of an account in the database. The verification of the identity of a subject in this case is based on entering a correct password. The basic assumption behind the policy in this example is wrong because passwords can be stolen or shared with others. Therefore, authentication based on only password does not guarantee correct verification of a subject.

As policy requirements get updated, new policies may overlap with existing policies. The overlap may sometimes mean that the functionality specified in a new policy is a super set of a policy that already exists. Such overlaps result in *Requirements Redundancies*. The problem with redundancies is that they lead to inconsistencies in the update of policies due to the existence of multiple versions. An update in one version of a policy while the other version with old policy can result in emergent and conflicting behaviours that are hard to be identified and fixed. The problem of *Requirements Inconsistency* between policies is realised when two or more policies are stated in such a way that their effect would be in conflict. The inconsistency problem often arises when there conflicting policies in a policy making team.

D. POLICY IMPLEMENTATION ISSUES

Issues specific to policy implementation include *Misunderstanding*, *Miscommunication*, *Inconsistency of Implementation Documentation*, *Implementation Redundancy*, *Implementation Update Documentation*, and *Misinterpretation*. For brevity, we will leave out the detailed discussion of these issues as they are similar in principle to their counterparts that have already been discussed in the policy making context. The problems of communication between the policy makers and implementers are well known to the Business Management community that is responsible for formulating policies [11], [12].

E. POLICY COLLABORATION ISSUES

Policy requirements specified by the policy maker need to be implemented as policy specification by the policy implementer. This requires collaboration between the two teams which results in the following issues. Collaboration issues include *Requirement Rationale, Forward Traceability, Inconsistency, Technical Language, Synchronization, Backward Traceability,* and *Implementation Knowledge*. A policy requirement is the outcome of a security policy requirements analysis process and as such, it is motivated by a *Requirement Rationale* which captures the reasoning behind the existence of the given policy requirement. In the context of security policies, a rationale is about the reasoning behind *"why we say what we say"* in the requirement. As an obvious example, why should students only be allowed to read their grades but should not be able to modify them? A simple motivation behind this requirement could be that as students are graded by their instructors, it should not be possible for any student to give themselves a grade as this will compromise the grading system and the quality of the course as well. The requirement rationale is not usually captured or documented as part the requirement shared with the policy implementers.

In the *Requirement Interpretation* task, policy implementers analyse the policy requirements in order to understand what is required, i.e. what is the expected behaviour of a system implementing the policies. From their understanding, they then come up with a *Policy Specification (P_S)*. During the update of the access control system, policy implementers need to be able to identify which policy requirement belongs to which policy specification. The ability to map a P_R to the corresponding P_S is called *Forward Traceability*. A lack of forward traceability would make it difficult to update the corresponding policy specification when there is a change in the policy requirement.

An *inconsistency* problem can be realised if the logic of the policy specification is not aligned to the policy requirement i.e. P_S is not designed properly to satisfy P_R. The root of the inconsistency between the P_R and P_S could result in from differing views/understanding of policy makers about a given policy requirement. The problem is further made worse, by the lack of transfer of *Requirement Rationale* to the policy implementers. Policy implementers get to know only the requirement without an understanding of the reasoning behind it, they often end-up with a faulty policy specification due to *misinterpretation* of the policy requirement. Such a specification would be inconsistent with the requirement. The differences in the policy making(PM_L) and the policy implementation(PI_L) *technical languages* which are used by the policy makers and policy implementers, respectively, leads to misunderstanding between the two teams. Policy makers may not have the *Policy Implementation Knowledge(K_I))* in order to have an appreciation of how the policy requirements they have proposed transformed into policy specifications(P_Is) by the policy implementation team.

When a change/update is made in the a policy requirement, such changes need to be also made to corresponding policy specifications for consistency. The reverse is also true. The policy *synchronization* problem occurs when changes/updates are only being made on one side without effecting the necessary changes in the other side. *Backward Traceability* is the ability to systematically identify a corresponding policy requirement, P_R, given a policy specification, P_S. Traceability is important for maintaining consistency in the update of policy requirements and specifications as it provides a systematic methods of tracing between requirements and their specifications.

We reviewed several AC tools to understand the policies they work with, the resources they protect and the level of support they offer for minimizing the AC collaboration problem. The review process followed to identify the tools included in this review paper is discussed in the next section.

III. REVIEW METHODOLOGY

In this section, we will first discuss the specific objectives and questions the review seeks to answer. After setting the scope of our research, we will discuss the process that we went through to identify the 31 access control tools that we have included in this review. This set of 31 access control tools are then compared to identify the operations that they support, the resources that they protect, the types of policies that they use, the visualizations that they utilize and the evaluations that were conducted on them.

A. SCOPE OF THE REVIEW

The main objective of the review is defined as follows:

Research Objective (RO): To identify the limitations of existing access control tools towards supporting policy making, collaboration, and implementation.

Specific Research Questions (RQs) arise from our RO. These include:

RQ1: How well do current access control policy authoring tools support *Policy Makers* towards eliciting policy requirements and ensuring that the policies are of good quality?

RQ2: How well do current tools for access control policy implementation support *Policy Implementers* towards implementing good quality policies?

RQ3: How well do current tools support the activities for *effective collaboration* between policy makers and policy implementers?

We picked these research questions to identify the support offered by currently available AC tools for creating, implementing and analysing policies. This also helped us to identify the shortcomings of the tools and that the tools failed to acknowledge the AC policy collaboration problem. Identifying these limitations helped us to explore visualizations as a potential solution to bridge the communication gap between the policy makers and implementers.

B. IDENTIFICATION OF THE LITERATURE

We searched for related papers by using search terms based on our RQs. The literature was identified using several combination of keywords. Some of the search terms are "access control policy" and "security policy" and "privacy

TABLE 2. 31 access control tools.

List of Access Control Tools reviewed in this paper	
Adage [7]	PeopleFinder [13]
ESCAPE [14]	Multi-level Grids [15]
Role Control Center RCC [16]	FB SNS [17]
HP Select Access Policy Builder [18]	FB SNS RPA [19]
Impromptu [20]	V3SPA [21]
Salmon [22]	PVA Tool: SELinux [23]
Grey [6]	SEGrapher [24]
IAM Wizard [1]	PVA Tool: RBAC [25]
RUBAVIZ [26]	SPARCLE [27]
Perspective [28]	MLSVisual [29]
Expandable Grids [22]	RBACVisual [30]
Expandable Grids P3P [31]	PRISIMOS [2]
UNIXVisual [32]	AudienceView [33]
DTEVisual [34]	SEFlowViz [35]
Sentential AC [36]	PACVIM [37]
PViz [38]	

policy". Using these search terms we collected the papers from IEEE, ACM, ScienceDirect, SpringerLink and Scopus. We then checked the title and abstract of each paper for relevancy. We then checked the related work of each of these papers to identify studies that may not have been gathered using our search terms. This set of tools is listed in Table 2.

IV. COMMON TASKS SUPPORTED BY CURRENT ACCESS CONTROL TOOLS

Though there has been various research to develop AC tools, these tools often lack support for many types of operations. We compiled a list of operations/tasks that were supported by each of the tools that we have reviewed in this survey. This list was then refined by grouping the tasks that were similar in functionality and eliminating some redundant tasks.

A. BASIC AC TOOL FEATURES

Each of the tools that we have reviewed was then compared against the list of tasks supported by all of the access control tools. This helped us to understand the commonly supported tasks and the lack of functionalities of the tools in more detail. Sets of similar tasks were grouped based on three components that would be responsible for executing each of these tasks. The three components are:

(C1) **Policy Designer:** The policy design component is responsible for creating and editing policies. Some tools supported policy authoring in which users can create their own policies using one or more of the following languages.

  a) Policy Creation: The policies can be created using two types of languages, namely:

    i) *Policy Language (PL)*: Policies can be created by users using a policy-specific language like XACML or by using an access control model-specific language like SELinux, RBAC etc. Some tools also supported policy creation using a language that the developers of the tools created. Often tools like ADAGE [7] created visual languages that help users to create their own policies. The tool is also capable of translating this specification into a policy structured language that the tool then analyses.

    ii) *Natural Language specification*: Users can create policies by specifying them in natural language or by using some interactive drop-down elements.

  b) Policy Editing: The policies can be updated after they have been implemented in the system.

(C2) **Policy Viewer:** This component supports two main functions for access control tools.

  a) *View one rule of a policy:* View one rule of a policy at a time, which indicates the permissions that are set by this policy for a single object or user.

  b) *View multiple rules of a policy:* View an overview of the policy, which displays multiple rules that indicate all the objects and users that this policy affects.

(C3) **Policy Evaluator:** This component makes the tool capable of analysing the policies and for offering solutions to resolve the errors that it identifies. Some of the analysis that can be performed on the policies include:

  a) Policy Analyser: This sub-component is responsible for analysing policies to identify any inconsistencies.

    i) *Conflict:* Two types of conflict detection is possible. The first type checks if a new policy conflicts with an existing policy. The other type makes sure that conflicting roles cannot be assigned to the same user.

    ii) *Similarity:* Check if two policies are similar to eliminate redundant policies.

    iii) *Compare versions:* Compare two versions of a policy to check how different the current version of the policy is from the last.

    iv) *Policy specific analysis:* Every access control model has it's own unique set of constraints and rules that are enforced using the policy language. The tool should be capable of analysing these policy specific rules and constraints.

  b) Policy Solver: The policy solver sub-component offers intelligent tips to solve the errors that it finds after analysing the policies. For example, consider the case where a policy $P1$ states that printers should be used between 9 am and 1 pm and policy $P2$ states that printers should be used between 10 am and 1 pm. In this case, the tool would identify that policy $P1$ dominates $P2$. The intelligent tips the tool offers could be to either delete $P2$ from the system as $P1$ already covers

*P*2, or to delete *P*1 if the objective is to only let the employees use the printers between 10-1.

After identifying the three components and their tasks, we compared them against each of the reviewed tools in Table 3. We examined the tools to identify the resources that the tool protects - files, data, Facebook settings or access to a location. An imbalance lies in the literature with more research focused on file AC. We also reviewed the tools by the functionality that they support. A functionality supported by the tool is represented by a green cell.

While there is certainly support for the Policy Design component, it's clear that only some tools offered support to create tasks in Natural Language which would help even the non-technical policy makers to create their own policies. More support for policy creation in a language that is both easy to understand and that can handle all the complexities would be able to bridge the communication gap between the policy makers and implementers. It is also noticeable that several tools offer no support for editing a policy. This lack of support for a simple yet fundamental operation that is performed can be attributed to the fact that some tools were developed for merely visualizing the policies available in a system. The objective of these tools were to simply inform the user's of the security implications of their access control settings. Another drawback that some tools face is their inability to view a single policy (which is made up of multiple rules) as a whole. Some tools only allow the user to view a policy, rule-by-rule which makes it harder for policy makers and implementers to understand the overall scope and functionality of the policy. These tools can facilitate better understanding and easier analysis if they offer an overview of the policy which summarizes all of the rules of the policy.

Evaluating a policy is the work of the policy implementer and these tasks are often cognitively demanding. From Table 2, it is clear that the most of the tools lack support for evaluating a policy. The PVA Tool-RBAC [25] offers an extra policy analysis task that allows policy professionals to integrate different RBAC policies. If the analysis operations are not available to the policy implementer, they might lead to errors during implementation or editing of the policy in the system. These errors would then have to be resolved before the policy is successfully implemented in the system. Finding the cause of the errors and then resolving them is a complex task. Only two tools in the table offered solutions to resolve some of these issues and errors. The IAM tool [1] identified the errors that arise after analysing the policy and offered tips to resolve these errors. The FB SNS tool [17] was also capable of identifying the differences between the intentions of the users while sharing a resource and the actual permissions that they set for the resource. User intention might be to share the resource only with college friends while the actual permissions that were set might have shared the resource to both college and work friends of the user. The tool after identifying the potential violations was able to offer tips on how the settings can be changed to align with the user intentions for that resource. The lack of support for such difficult tasks which are crucial to the working of the implementer lay a lot of cognitive stress on the users.

B. RATIONALE FOR MAPPING OF COLLABORATION ISSUES TO TOOL FEATURES

Table 3 shows an evaluation of which AC tools support common access control operations. The key question we aim to address in this paper is to what extent do the AC tools address the policy collaboration issues shown in Fig. 1. In order to answer this question we performed two tasks: (1) mapping collaboration issues to the common features of AC tools; and (2) evaluating the support for collocation issues. The mapping of collaboration issues to the AC tool features is an intermediate step towards answering our key research question. It helps in relating tool features to collaboration issues. As the support of common features by different tools is already documented in Table 3, this intermediate step helps to relate AC tools to the policy collaboration issues. The mapping between common features of AC tools and policy collaboration issues is shown in Table 4. The columns in this table are the common features of AC tools and the rows are the policy collaboration issues. A tick in a cell of the table indicates that the given AC feature (column) can contribute towards addressing the given collaboration (row) issue. The rationale for mapping each of the collaboration issues to the tool features depicted in Table 4 is explained in this section.

The *misunderstanding* problem is addressed if the tool has the ability to express policies in natural language (NL), provides a user interface (UI) for authoring policies, provides the ability to view both single and multiple policies. When policies are expressed in natural language, misunderstanding for policy makers is reduced since policy requirements are initially expressed in natural language. The provision of a user interface for displaying the contents of policies (either as single or multiple rules) is a fundamental requirement for reducing misunderstanding of policies.

The policy *update* problem is mapped to the ability of a tool to enable the user to *edit* and *compare the versions of policies*. A prerequisite for updating a policy is the availability of functionality in a tool that makes it possible to make edits, and once edits are made it is important to be able to compare the new version of a policy with the previous version of other existing policies. This helps in identifying commonalities and overlaps between policies helps with the process of factoring out those commonalities. *Incompleteness* of a policy requirements can be determined if an AC tool supports two main features: *constraint analysis* and *incompleteness detection*. When some constraints necessary for the satisfaction of policy requirements are not taken into account in the policy requirements, it is like that the policy would not be feasible to enforce. For example, the policy that users accessing sensitive information in database should use strong passwords is not enforceable if it is not made clear what is meant by a strong password. For example, the minimum password length should be 10 characters, the

TABLE 3. Comparison of the operations of a tool.

OPERATIONS / AC TOOLS	ADAGE [7]	ESCAPE [14]	Role Control Center RCC [16]	HP Select Access Policy Builder [18]	Impromptu [20]	Salmon [22]	IAM [1]	Perspective [28]	Expandable Grids [22]	Multi-level Grids [15]	V3SPA [21]	PVA Tool - SELinux [23]	SEGrapher [24]	SEFlowViz [35]	PVA Tool - RBAC [25]	DTEVisual [34]	MLSVisual [29]	RBACVisual [30]	UNIXVisual [32]	Rubaviz [26]	SPARCLE [27]	PACVIM [37]	Expandable Grids for P3P [31]	Prisimos [2]	PeopleFinder [13]	Grey [6]	Sentential AC [36]	FB SNS [17]	FB SNS RPA [19]	Audience View [33]	PViz [38]
Resource Type	FAC	FAC	FAC	FAC	FAC	FAC	FAC	FAC	FAC	FAC	FAC	FAC	FAC	FAC	FAC	FAC	FAC	FAC	FAC	DAC	DAC	DAC	DAC	Viz P3P	Loc AC	Loc AC	Loc AC	FB Policy	FB Policy	FB Policy	FB Policy
C1: POLICY DESIGNER																															
Policy Creator																															
Create policy in Natural Language	🟩				🟩		🟩														🟩										
Create policy using interface or PL	🟩	🟩	🟩	🟩	🟩	🟩	🟩		🟩	🟩		🟩				🟩	🟩	🟩				🟩	🟩		🟩	🟩	🟩		🟩	🟩	
Policy Editor																															
Edit Policy	🟩	🟩	🟩	🟩	🟩	🟩	🟩		🟩		🟩	🟩		🟩		🟩	🟩	🟩			🟩	🟩			🟩	🟩	🟩			🟩	
C2: POLICY VIEWER																															
View a single rule	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩
View multiple rules	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩		🟩	🟩		🟩	🟩					🟩	
C3: POLICY EVALUATOR																															
Policy Analyser																															
Conflict																															
– View conflicting roles			🟩				🟩							🟩																	
– View conflicting policies							🟩								🟩									🟩							
Similarity / Dominance											🟩													🟩							
Compare version										🟩																					
Policy Specific Constraints		🟩	🟩	🟩							🟩	🟩	🟩							🟩	🟩										
Policy Solver																															
Resolve							🟩																					🟩			

FAC - File AC P3P - Platform for Privacy Preferences Project DAC - Data AC Loc - Location FB - Facebook

inclusion of numbers, capital and small letters, and the special characters, are useful in ensuring the strength of a password can be enforceable. Without these constraints the password policy is incomplete. Incompleteness detection depends on the ability to identify missing constraints from the vagueness of a policy requirement.

The problem of *redundancy* during policy making can be detected and resolved if the AC tool has the ability to *detect similarities* and *compare versions* between policy requirements and rules. Through similarity comparison it is possible to identify those policies that are closely related or have commonalities. In the same vein, version comparison can detect if a new policy is repeating a policy that already exists. Therefore, these two features of AC tools are necessary for detecting policy redundancies. AC tools can contribute towards addressing *inconsistencies* if they have the following features: multiple rule viewing, role conflicts detection, policy conflicts detection, constraints analysis, and conflict resolution. From an inconsistency perspective, the ability to view, analyse, and compare multiple policy rules is necessary in order to determine whether a conflict exists between them. In addition to the ability to detect role and policy conflicts, the resolution of such conflicts is necessary for a complete solution towards addressing the policy requirements inconsistency problem. AC tools with features for documenting *requirements-to-implementation mapping* can address the

TABLE 4. Mapping collaboration issues to tool features.

			Policy Design					Policy View		Policy Evaluation									
			Policy Creation				Policy Update			Policy Analysis								Resolution	
			Author Policy																
			NL	PL	UI for Policy Authoring	Rationale Documentation (Justification + Assumptions)	Edit Policy	View Single Rule	View Multiple Rules	Role Conflicts	Policy Conflicts	Similarity/Dominance	Compare Version	Constraints Analysis	Incompleteness Detection	Requirements-to-Implementation Mapping	Implementation-to-Requirements Mapping	Resolver	N_f
Policy Collaboration Problem	Policy Making	Misunderstanding	✓		✓			✓	✓										4
		Update					✓						✓						2
		Incompleteness												✓	✓				2
		Redundancy										✓	✓					✓	3
		Inconsistency							✓	✓	✓			✓				✓	5
		Forward Traceability														✓			1
	Policy Collaboration	Rationale				✓													1
		Inconsistency	✓						✓	✓	✓								4
		Language	✓																1
		Synchronization											✓						1
		Implementation Knowledge					✓												1
	Policy Implementation	Misunderstanding		✓	✓														2
		Inconsistency							✓	✓	✓			✓				✓	5
		Backward Traceability															✓		1
		Redundancy										✓	✓					✓	3

Forward Traceability issue. This is because requirements-to-implementation mapping makes it possible to trace policy requirements to their policy implementations - a necessary feature for ease of maintenance of policies. On the contrary, addressing the *Backward Traceability* issue requires the documentation of implementation-to-requirements mapping.

AC tools with features for *documenting requirement rationale* can contribute towards addressing the requirements *rationale* issue. Every requirement needs to have its rationale documented in order to ascertain the origins and reasons behind the proposal of the policy requirement. In order to document policy *Implementation Knowledge*, AC tools need to have the ability to *edit* them. Without editing features, capturing implementation knowledge would not be possible. The *synchronization* of policy requirements and their specifications between policy makers and policy implementers requires AC tools to have the ability to compare versions of policies and a documentation of the mapping between requirements to their implementations (and vice versa). In this way, changes happening on the requirements side can be synchronised to the implementation. In the same way, changes happening in the implementation can be synchronised to the corresponding requirements.

For each collaboration issues there are a finite number, N_f, of features that can support that issue. For example, the issues of *incompleteness* of policy requirements during *policy making* can be addressed if the AC tools have features for *constraint analysis* and *incompleteness detection*. In this case, $N_f = 2$.

C. EVALUATION OF HOW WELL AC TOOLS ADDRESS THE COLLABORATION ISSUES

Table 5 shows an evaluation of the extent to which each AC tool supports the collaboration issues. This table is derived through a combination of the information in Tables 3 and 4. The rows are the collaboration challenges and the columns are the AC tools. For each AC tool, t, we identify the set of features, $\Phi(t)$ it supports from Table 3. For each collaboration issue, using Table 4, we determine how many features from $\Phi(t)$ can support the collaboration issue. We call this number N_c. The extent to which a given AC tool supports a collaboration issue is given by the ratio of N_c/N_f. If $N_c = N_f$ that means the given tool has all the features necessary to support the collaboration issue. For example, in Table 5 the ADAGE tool has the ratios $4/4$ for misunderstanding of policy requirements, $1/5$ for inconsistency of policy requirements during policy making. The ratios help in comparing the tools against each other on the extent to which they address a given policy collaboration issue. For example the Rubaviz and FB SNS tools are the poorest in addressing the *misunderstanding* issue of policy requirements during policy making because they both support $1/4$ features required to address it.

TABLE 5. Evaluation of how well ac tools support collaboration issues. colors indicate level of support. 🟥 $0 \leq$RED$\leq \frac{1}{4}$, 🟧 $\frac{1}{4}<$ORANGE$\leq \frac{1}{2}$, 🟨 $\frac{1}{2}<$YELLOW$\leq \frac{3}{4}$, 🟩 $\frac{3}{4}<$GREEN≤ 1 .

ISSUES / AC TOOLS	ADAGE [7]	ESCAPE [14]	Role Control Center RCC [16]	HP Select Access Policy Builder [18]	Impromptu [20]	Salmon [22]	IAM [1]	Perspective [28]	Expandable Grids [22]	Multi-level Grids [15]	V3SPA [21]	PVA Tool - SELinux [23]	SEGrapher [24]	SEFlowViz [35]	PACVIM [37]	PViz [38]	Sentential Access Control [36]	PVA Tool - RBAC [25]	DTEVisual [34]	MLSVisual [29]	RBACVisual [30]	UNIXVisual [32]	Rubaviz [26]	SPARCLE [27]	Expandable Grids for P3P [31]	Prisimos [2]	PeopleFinder [13]	Grey [6]	FB SNS [17]	FB SNS RPA [19]	Audience View [33]
Policy Making Issues																															
Misunderstanding	$\frac{4}{4}$	$\frac{3}{4}$	$\frac{3}{4}$	$\frac{3}{4}$	$\frac{4}{4}$	$\frac{3}{4}$	$\frac{4}{4}$	$\frac{2}{4}$	$\frac{3}{4}$	$\frac{3}{4}$	$\frac{2}{4}$	$\frac{3}{4}$	$\frac{2}{4}$	$\frac{2}{4}$	$\frac{3}{4}$	$\frac{1}{4}$	$\frac{2}{4}$	$\frac{2}{4}$	$\frac{3}{4}$	$\frac{3}{4}$	$\frac{3}{4}$	$\frac{2}{4}$	$\frac{1}{4}$	$\frac{3}{4}$	$\frac{2}{4}$	$\frac{2}{4}$	$\frac{2}{4}$	$\frac{2}{4}$	$\frac{1}{4}$	$\frac{2}{4}$	$\frac{3}{4}$
Update	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$		$\frac{1}{2}$	$\frac{1}{2}$		$\frac{1}{2}$	$\frac{1}{2}$		$\frac{1}{2}$		$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$			$\frac{1}{2}$			$\frac{1}{2}$	$\frac{1}{2}$			$\frac{1}{2}$
Incompleteness		$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$							$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$					$\frac{1}{2}$												
Redundancy							$\frac{1}{3}$			$\frac{1}{3}$	$\frac{1}{3}$							$\frac{1}{3}$								$\frac{1}{3}$			$\frac{1}{3}$		
Inconsistency	$\frac{1}{5}$	$\frac{2}{5}$	$\frac{3}{5}$	$\frac{2}{5}$	$\frac{1}{5}$	$\frac{1}{5}$	$\frac{4}{5}$	$\frac{1}{5}$	$\frac{1}{5}$	$\frac{1}{5}$	$\frac{2}{5}$	$\frac{2}{5}$	$\frac{2}{5}$	$\frac{2}{5}$	$\frac{1}{5}$			$\frac{2}{5}$	$\frac{1}{5}$	$\frac{1}{5}$	$\frac{1}{5}$	$\frac{1}{5}$	$\frac{1}{5}$	$\frac{2}{5}$		$\frac{2}{5}$	$\frac{1}{5}$		$\frac{1}{5}$		$\frac{1}{5}$
Forward Traceability																															
Policy Collaboration Issues																															
Rationale																															
Inconsistency	$\frac{3}{5}$	$\frac{2}{5}$	$\frac{3}{5}$	$\frac{2}{5}$	$\frac{3}{5}$	$\frac{2}{5}$	$\frac{5}{5}$	$\frac{1}{5}$	$\frac{2}{5}$	$\frac{2}{5}$	$\frac{1}{5}$	$\frac{2}{5}$	$\frac{1}{5}$	$\frac{2}{4}$	$\frac{1}{4}$			$\frac{2}{5}$	$\frac{2}{5}$	$\frac{2}{5}$	$\frac{2}{5}$	$\frac{1}{5}$	$\frac{1}{5}$	$\frac{2}{5}$	$\frac{1}{5}$	$\frac{2}{5}$	$\frac{2}{5}$	$\frac{1}{5}$		$\frac{1}{5}$	$\frac{2}{5}$
Language	$\frac{2}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{2}{2}$	$\frac{1}{2}$	$\frac{2}{2}$	$\frac{1}{2}$	$\frac{1}{2}$			$\frac{1}{2}$							$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$			$\frac{1}{2}$	$\frac{1}{2}$		$\frac{1}{2}$	$\frac{1}{2}$		$\frac{1}{2}$	$\frac{1}{2}$
Synchronisation		$\frac{1}{1}$								$\frac{1}{1}$																					
Implementation Knowledge	$\frac{1}{1}$	$\frac{1}{1}$	$\frac{1}{1}$	$\frac{1}{1}$	$\frac{1}{1}$	$\frac{1}{1}$	$\frac{1}{1}$	$\frac{1}{1}$			$\frac{1}{1}$	$\frac{1}{1}$		$\frac{1}{1}$	$\frac{1}{1}$		$\frac{1}{1}$		$\frac{1}{1}$	$\frac{1}{1}$	$\frac{1}{1}$			$\frac{1}{1}$			$\frac{1}{1}$	$\frac{1}{1}$			$\frac{1}{1}$
Policy Implementation Issues																															
Misunderstanding	$\frac{2}{2}$	$\frac{2}{2}$	$\frac{2}{2}$	$\frac{2}{2}$	$\frac{2}{2}$	$\frac{2}{2}$	$\frac{2}{2}$	$\frac{2}{2}$	$\frac{2}{2}$			$\frac{2}{2}$			$\frac{2}{2}$		$\frac{2}{2}$		$\frac{2}{2}$	$\frac{2}{2}$	$\frac{2}{2}$	$\frac{2}{2}$	$\frac{2}{2}$				$\frac{2}{2}$	$\frac{2}{2}$		$\frac{2}{2}$	$\frac{2}{2}$
Inconsistency	$\frac{1}{5}$	$\frac{2}{5}$	$\frac{3}{5}$	$\frac{2}{5}$	$\frac{1}{5}$	$\frac{1}{5}$	$\frac{4}{5}$	$\frac{1}{5}$	$\frac{1}{5}$	$\frac{1}{5}$	$\frac{2}{5}$	$\frac{2}{5}$	$\frac{2}{5}$	$\frac{2}{5}$	$\frac{1}{5}$			$\frac{2}{5}$	$\frac{1}{5}$	$\frac{1}{5}$	$\frac{1}{5}$	$\frac{1}{5}$		$\frac{2}{5}$		$\frac{2}{5}$	$\frac{1}{5}$		$\frac{1}{5}$		$\frac{1}{5}$
Backward Traceability																															
Redundancy							$\frac{1}{3}$			$\frac{1}{3}$	$\frac{1}{3}$															$\frac{1}{3}$			$\frac{1}{3}$		

The ratios in Table 5 are color coded with 4 colours: green yellow, orange, and red. Green means the ratio of support is in the range $>\frac{3}{4}$ to 1. The percentage support ranges $>\frac{1}{2}$ to $\frac{3}{4}$, $>\frac{1}{4}$ to $\frac{1}{2}$, and 0 to $\frac{1}{4}$, correspond to yellow, orange, and red, respectively. For example, we can easily see by the green color that most of the tools address *misunderstanding* of policies during *policy implementation*. We can also see that detection and resolution of inconsistency of policy requirements during *policy making* is not supported by most of the tools. A white space implies that the AC tools does not have any support for the given collaboration issue. For example, *Forward Traceability, Rationale,* and *Backward Traceability* are not supported by any of the tools. Table 5, also shows that, overall, the existing AC tools have more support for *policy implementation* issues compared to *policy making* issues.

V. COMPARISON OF AC TOOLS BASED ON VISUALIZATION TECHNIQUE

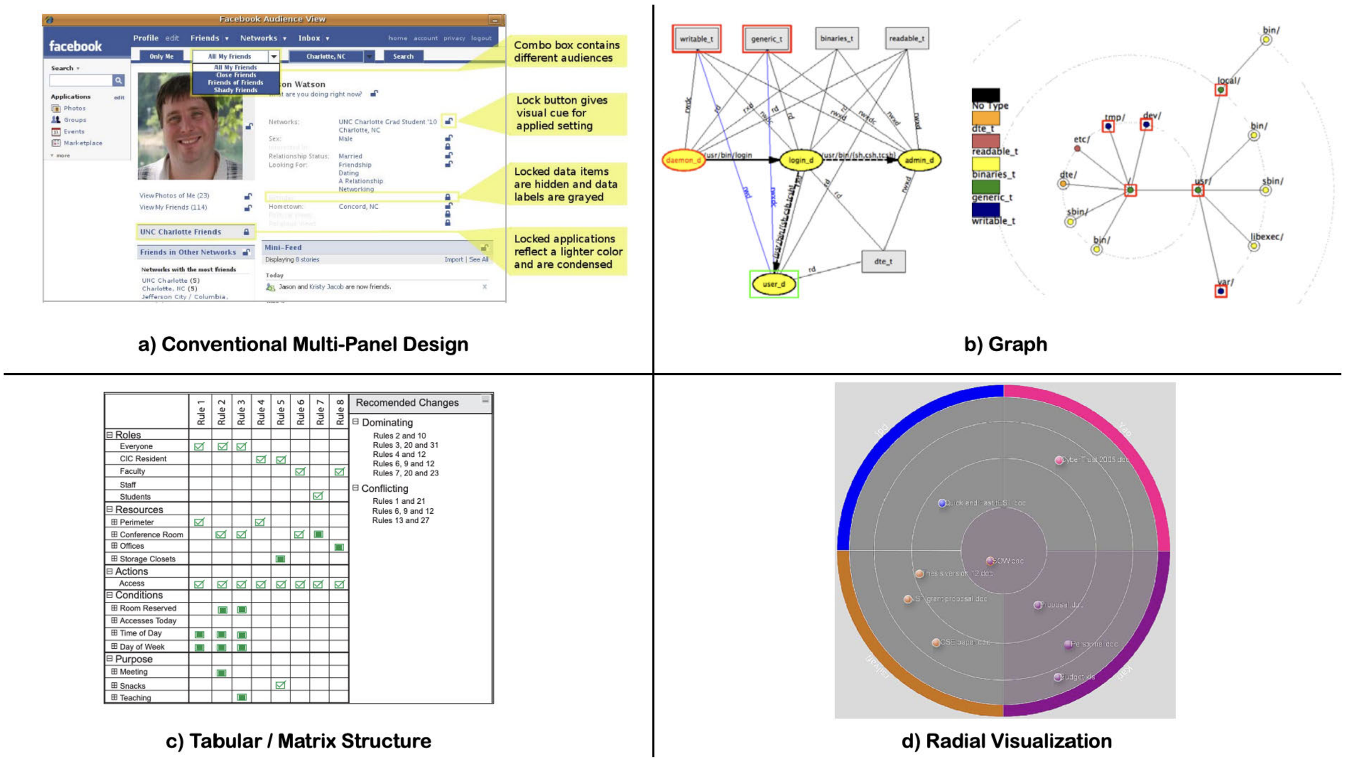
"A picture is worth a thousand words" is an adage which means that it is often much easier to express complex ideas with a simple visualisation than a lot of descriptive words. Using the same line of thought, we came up with the hypothesis that the issues related to misunderstanding and miscommunication in the collaboration problem can be addressed through visualisation features in the AC tool. This section compares different types of visualisations used in the tools we reviewed. We observed that several AC tools used simple visualizations to reduce the cognition demands that are required to understand an AC policy [23] by applying visualizing techniques to visualize the policy components. These tools can even be used by users who do not possess AC expertise for policy creation, implementation and management [39]. These tools tackle the problem of policy complexity by visualizing the policies. In addition to reducing the cognitive levels, the visualizations are an excellent solution to bridge the gap between the technical policy implementers and non-technical policy makers. The visual language helps policy makers to understand the implemented policy while allowing the policy implementers to understand the rationale behind the formulation of the policies.

In this section, we compare different tools based on techniques used for visualizing the policies. In addition to tackling the problems of policy complexity, the policy visualization must also be simple enough so that it can be understood by a non-technical user. An effective visualization tool must help both Policy Makers and Policy Implementers and should address several design goals. It should be able to support the following tasks:

1) Extract semantic information from policy attributes to visualize policy details, so that the policy specifications can be clearly understood by both policy makers and implementers.
2) Support policy editing.
3) Provide an overview of the policy.
4) Support policy analysis.
5) Offer solutions to identify and resolve policy inconsistencies.

Different visualization techniques help in solving different facets of policy complexities while some types of visualizations are more equipped to support a wider range of functionality. Fig. 2 shows different types of visualizations used by AC tools.

1) **Conventional multi-panel design** The tools based on this type of design use a simple interface design with multiple panes, which are good for creating/editing the policies by dragging components from another pane. These tools look simple but unfortunately they don't seem to support a variety of operations. Due to their simplistic interfaces, these tools help policy makers to easily create their own policies without relying on the policy implementers. The Adage [7] and the Audience View [33] tools use simple conventional multi-panelled design. The Adage tool used it for creating policies with simple drag-and-drop components. On the other hand, the Audience View tool used it for displaying the privacy settings in a multi-panel display that resembled Facebook's profile information and layout.
   Four tools [7], [14], [17], [33] used a simple interface with images or other minimal visualizations. The Adage tool [7] uses a two-panel interface to display a visual policy builder which consists of components in one panel that can be dragged and dropped into another panel to build policies. Similarly the ESCAPE [14] tool uses a 3 panelled interface with the resources in one panel, the names of all the users in one panel from which you can drag and drop users into another panel to grant them to access to the selected resource in the third panel. On the other hand, the [17], [33] tools used simple interfaces with images and minimal interactions to display ways to resolve policy violations and for editing policy settings respectively. In addition, the Windows File Permissions tool also follows a simple multi-panelled interface. From the working of these tools, it's clear that these tools are simplistic in nature which makes it easier for less technically skilled individuals to easily create, edit and analyse policies. Hence, these advantages can potentially help non-technical policy makers to understand different phases of policy implementation.
2) **Graphs** Graphs are commonly found in topology-based relations as specified in SNS style policies. They are also used to represent hierarchical role structure in RBAC policies. These are a common form of visualization with seven tools [16], [19], [23], [24], [26], [29], [32], [34] using this in their access control tools. The graph visualizations are especially used for visualizing network style graphs of SNS. They are good at conveying group memberships or roles to which multiple users belong by drawing edges between the various users, groups and resources in the visualization. They also clearly represent the structure of the hierarchy in the organisations by using directed edges and also help in identifying outliers in policies. For example, a user who is assigned a managerial role is visualized by drawing a directed edge from the user node to the manager node. This makes it easy to identify when a user is given access to a resource that is meant only for managers, as there is no edge between the user and manager nodes. Though graphs are adept in visualizing multiple relationships between the components while maintaining the hierarchical structure, an overload of information often make them look cluttered. The DTEVisual Tool [34] shown in Fig. 2 b) uses graphs to denote the relationship between domain and type for visualizing the resources in the form of a radial tree in which files at the same level are visualized on the same circle. Graph visualizations have the ability to easily convey the hierarchical structure of the organisation. They help policy makers to understand the organisational hierarchy while creating policies and also help policy implementers to understand the implications of the implemented policies.
3) **Tabular/Matrix/Grid structure** Tabular visualizations can be easily understood by everyone due to their familiar way of representing data in tables. This type of visualization is mostly used for visualizing the permissions that are used assigned to objects. These visualizations are often interactive in nature and helped to view and edit permissions simultaneously. They are also a common type of visualization with 7 tools [1], [2], [18], [22], [26], [27], [31] using it in their tools. The IAM Wizard [1] and Expandable Grids [22] Tools used grid visualizations to visualize the resources and the levels of access in a matrix format. On the other hand, the PRISIMOS tool [2] shown in Fig. 2 c) uses a tabular visualization to visualize the policies that grant access to the resources.
   Four tools [6], [13], [15], [21] used grids for visualizing the permissions for different objects. These visualizations support group memberships by collapsing rows or columns to denote members who belonged to a particular group or role. Though these visualizations are good for indicating whether a user or a group of users have access to a resource by using colors or icons, they do not offer an overview of all the users who have access to a particular resource like graph


Facebook Audience View
Combo box contains different audiences
Lock button gives visual cue for applied setting
Locked data items are hidden and data labels are grayed
Locked applications reflect a lighter color and are condensed
a) Conventional Multi-Panel Design
writable_t
generic_t
binaries_t
readable_t
daemon_d
login_d
admin_d
user_d
dte_t
No Type
dte_t
readable_t
binaries_t
generic_t
writable_t
b) Graph
c) Tabular / Matrix Structure
d) Radial Visualization

	Rule 1	Rule 2	Rule 3	Rule 4	Rule 5	Rule 6	Rule 7	Rule 8
⊟ Roles								
Everyone	✓	✓	✓					
CIC Resident				✓	✓			
Faculty						✓		✓
Staff								
Students							✓	
⊟ Resources								
⊞ Perimeter	✓			✓				
⊞ Conference Room		✓	✓			✓	🟩	
⊞ Offices								🟩
⊞ Storage Closets					🟩			
⊟ Actions								
Access	✓	✓	✓	✓	✓	✓	✓	✓
⊟ Conditions								
⊞ Room Reserved		🟩	🟩					
⊞ Accesses Today								
⊞ Time of Day	🟩	🟩	🟩					
⊞ Day of Week	🟩	🟩	🟩					
⊟ Purpose								
⊞ Meeting		🟩						
⊞ Snacks					✓			
⊞ Teaching			🟩					

Recomended Changes
⊟ Dominating
Rules 2 and 10
Rules 3, 20 and 31
Rules 4 and 12
Rules 6, 9 and 12
Rules 7, 20 and 23
⊟ Conflicting
Rules 1 and 21
Rules 6, 9 and 12
Rules 13 and 27

FIGURE 2. **Common visualization techniques: a) Conventional Multi-Panel Design: [33], b) Graph: [34], c) Matrix/Tabular structure: [2], d) Radial Visualization: [20] (Images a) and c) show the top parts of scrollable interfaces. Images b) and d) appear in low resolution in their respective papers).**

visualizations do. In the case of graph visualizations, a policy is displayed in its entirety, whereas tabular displays are limited by space and only display portions of the policy at a time. This disadvantage limits their ability to act as good representations for policy analysis that are performed by implementers. However, due to their simple and familiar way of representing data, they are a good choice for non-technical users and policy makers to understand the implications of the policies.

4) **Radial Visualizations** The Impromptu tool [20] uses concentric circles to visualize the files that are available to users with different levels of access. The main objective of the tool is to only identify how the files are being used which could be easily indicated using this simple visualization. However, it would not be able to visualize more complex functions and relationships. The Impromptu tool [20] shown in Fig. 2 d) uses concentric circles to visualize the available files and the levels of access associated with them.

Lipford et al. [40] compared two tools namely, AudienceView [33] which used a simple interface design and Expandable Grids [22]. They conducted user studies to check if users preferred the more visual tool like AudienceView which used a simple interface with a layout similar to Facebook or the Expandable Grids tool which used a more compact visualization like grids to visualize the privacy settings. They concluded that users liked the visual feedback of AudienceView but not the multiple page visits. The users also liked the compact overview of Expandable Grids which had all the settings in one page even though it lacked the visual feedback that AudienceView provides. A common sentiment among participants was that they preferred a mix of the two approaches, with one participant stating that they would use Expandable Grids for its privacy settings and then using AudienceView for its visual feedback to tweak the settings.

While Fig. 2 shows screenshots of the commonly used visualization techniques, Fig. 3 shows how these visualization techniques visualize the same set of data. This will help us to identify the ease with which each technique can convey the required information. Fig. 3 shows how the different visualizations represent a simple scenario in which the Manager, Employee 1 and Employee 2 are assigned resources r1, r2, r3 and r4 with varying degrees of permissions. Fig. 3 a) shows the process of building this policy visually. Fig. 3 b) shows the inheritance of the resources. While employee 1 has access to resources r2 and r3, employee 2 has access to r4. The manager inherits the access to all of these three resources in addition to the resource r1 that the manager has exclusive access to. However, this does not show the level of access permissions that the users have on these resources. Fig. 3 c) shows the levels of permissions that a user has for different resources. Fig. 3 d) shows the resource matrix that represents the users, resources and permissions. It is clear that the some visualizations do not show all of the information but are better for viewing the structures of the role hierarchy, or for viewing the varying degrees of permissions easily. Other visualizations might be better for showing all of the information in a compact manner. Based on the pros and cons of each visualization, here are some recommendations on which visualization to choose based on the scope of the access control tool.

1) Simple multi-panelled interfaces do not easily support policy analysis as they only contain simple drag-and-drop elements which are better suited for policy creation and for displaying images that can be used to resolve policy violations. They do not use advanced visualizations which help in effective policy analysis. This makes them a good choice for policy makers to understand the implications of the policies that they create.
2) Graphs are good visualizations to consider if the structure of the hierarchy of permissions, visualizations of group/role memberships and spotting outliers of a policy are important. It is also quite capable of visualizing complex policy details. Hence, they are a good choice for policy implementers to analyse the policies.
3) Tabular forms of visualizations are good if the tool requires a compact visualization, and supports easy policy editing. It is good for visualizing simple policies, which makes them a good choice for non-technical policy makers.

The visualizations used by the tools are created either for the purpose of creating policies or for analysing and editing the policies. The need for different representations of a policy emphasizes the need for stronger notations in visualizations that capture all the concepts of the access control model for effective policy creation. In contrast, graph visualizations are mostly designed for policy implementers to analyse policies without any support for creating or editing policies. On the other hand, the grid/matrix structure visualizations are both simple enough to be used by policy makers to create their own policies and for facilitating easy policy editing and simple policy analysis by policy implementers. Policy analysis using the matrix structure is not very comprehensive owing to the simple structure of the matrix visualizations. The Expandable Grids tool which uses a matrix visualization, grouped the users by roles who are assigned a particular resource. This allows for easy visualization of the users who have different permission settings than what is normally assigned to other users who are assigned the same role. However, owing to the simple tabular structure of the visualization, it would require a great deal of scrolling to see all the users who are assigned to a group. On the other hand, policy editing is extremely simple and only required clicking on a cell in the grid to grant/revoke privileges. Heitzmann et al. [41] used a simple TreeMap visualization to visualize the file system of a single user or a groups of user. This visualization in which each node represents a file in the computer system was colored according to the permission settings for that file.

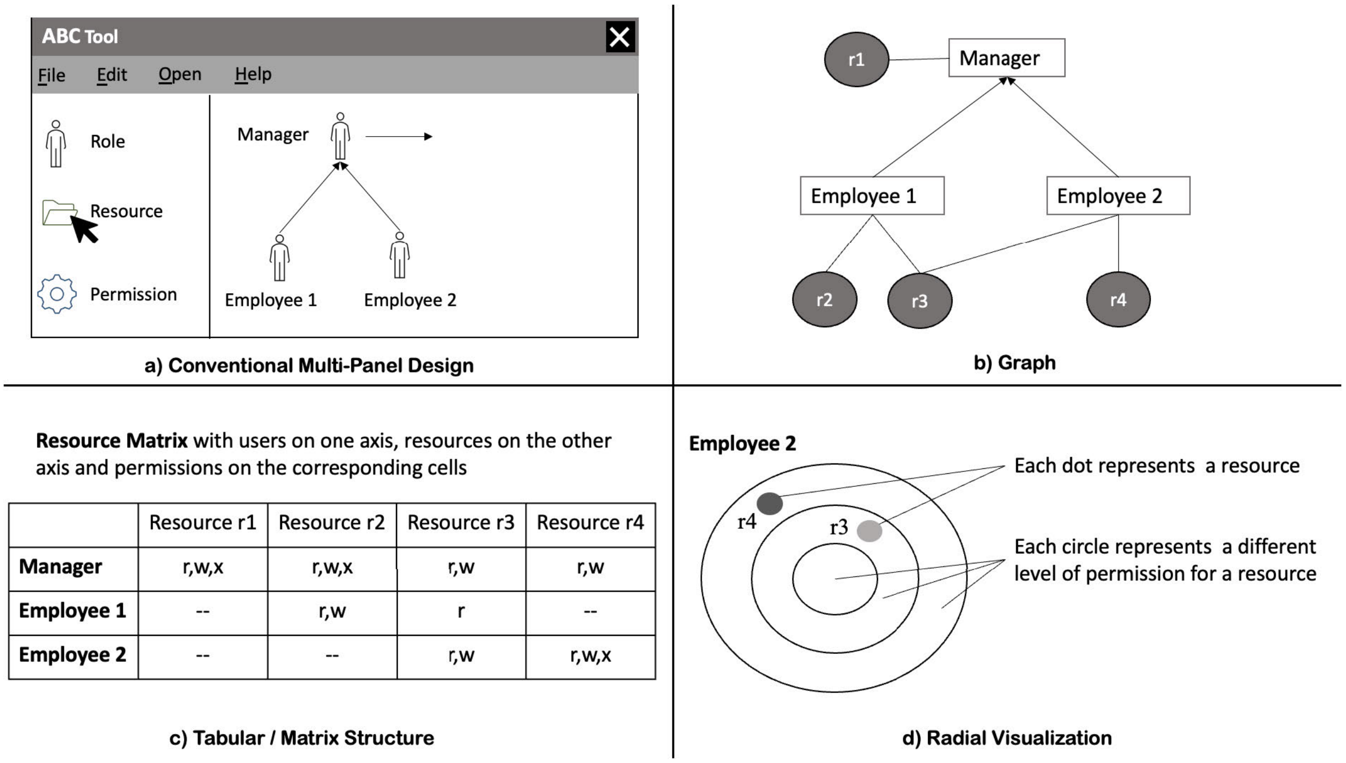
Hence, visualizations may help to bridge the communication gap between policy makers and implementers. They help policy makers to create their own policies without relying on the policy implementers. This in turn controls the errors of interpretation that the implementers face when they have to implement a policy that the policy makers create. They also help policy implementers to analyse the existing policies and serve as a medium using which implementers can point out any inconsistencies to policy makers.

VI. COMPARISON OF THE USER STUDIES OF THE TOOLS

Another common drawback of the research-based AC tools was that most of the evaluations conducted on the tools was limited. Of the 31 papers that were reviewed for this paper, it was found that approximately 40% of the tools did not conduct user studies to evaluate the tools. Although some of the other tools did conduct user study evaluations, these evaluations were not conducted with real policy professionals who would actually use these tools. As a result, none of the tools recognised that two sets of policy professionals (policy makers and policy implementers) are involved in the creation, manipulation and implementation of the policies. Hence, none of the tools attempted to address the collaboration problem between the policy makers and implementers. Of the 60% of tools that actually conducted user studies, only 3 studies evaluated their tools using system administrators or working professionals, while the other studies only recruited students to evaluate their tools. Student evaluations are suitable for for pedagogical tools as the tools were created for teaching students about the different AC models and policies. However, for other tools that were designed for working professionals, better feedback could have been achieved by involving working professionals in the user study.

The tools that conducted user studies followed different modes of evaluation:

1) **Evaluation is focused on a specific component/capability of the tool:** Tools like SPARCLE [27] and Expandable Grids for P3P [31] that dealt with parsing policies and presenting them in NL to the users are included in this type. These tools focused on the NL parsing ability of the tool. While the SPARCLE tool also evaluated the tool's ability to help the user with authoring policies, the Expandable Grids for P3P tool only focused on the NL parser.
2) **Evaluation of the tool - Visual or Interactive analysis:** The participants were assigned tasks that could be performed by visually verifying the content on the screen with minimal interaction, while other studies required the participants to engage and interact actively with the tool to successfully complete the tasks. The user study for the social network tools FB SNS [17] and FB SNS RPA [19] mostly required the users to visually or mentally analyse the information on the screen. The FB SNS tool did not require the users to interact with the tool. The users merely had to check if the tool had


ABC Tool
File Edit Open Help
Role
Resource
Permission
Manager
Employee 1
Employee 2
a) Conventional Multi-Panel Design
r1
Manager
Employee 1
Employee 2
r2
r3
r4
b) Graph
Resource Matrix with users on one axis, resources on the other axis and permissions on the corresponding cells
c) Tabular / Matrix Structure
Employee 2
r4
r3
Each dot represents a resource
Each circle represents a different level of permission for a resource
d) Radial Visualization

	Resource r1	Resource r2	Resource r3	Resource r4
Manager	r,w,x	r,w,x	r,w	r,w
Employee 1	--	r,w	r	--
Employee 2	--	--	r,w	r,w,x

FIGURE 3. Commonly used visualization techniques visualizing an example scenario involving 3 users/subjects and 4 resources. a) Conventional Multi-panel Interface showing a user creating the policy using drag-and-drop elements b) A graph representation of the users and resources c) Tabular/Matrix representation showing all of the users, resources, and permissions. d) Radial Visualization showing the resources that can be accessed and their permissions for a single employee *Employee 2*.

correctly identified potential violations between their intentions for a post/status/image and the actual setting that they had applied on it. On the other hand, the FB SNS RPA tool required the users to just mentally analyse the policies in one setting while interacting with the tool in another setting.

3) **Comparison with a baseline tool:** Some tools took the user study a further step by comparing their tool with a related/baseline tool.

A comprehensive comparison of the user studies of the tools based on the above modes of evaluation is shown in Table 6.

The studies used a variety of tasks to evaluate the user's understanding of the tools. These tasks tested the user's ability to author policies or analyse the implemented policies using the tools. The ADAGE, SPARCLE and FB SNS RPA tools were the only tools that tested the users ability to create their own policies in the tool. The ADAGE tool in particular was the only tool to delegate tasks to check policy authoring, analysing and editing. The other tools predominantly focused on the user's ability to analyse the policies and check if they are correctly implemented. Some tools went a step further with their tasks, and required their users to edit the policies according to a policy description. For example, some tasks involved checking if users understood the access control permissions for files. The users verified if the permission settings for a file were correctly implemented. If it was not correctly implemented then the users had to edit the permission settings to match the policy description. The tasks were clearly biased towards understanding if the policy implementers found the tool easy to analyse and edit policies and were not designed for understanding the user's ability to create policies. If the tasks were also designed to check the ease of use for authoring new policies then the tools might have encouraged policy makers to use the tool to create their own policies.

Though the tools catered to and supported a large range of policies and functions, they all shared a common drawback - lack of involvement with the end-users. Most of the tools were built on the assumption that users of the tool would be expert users who can understand complex policy analysis with minimal support. However this seems to be contrary to Botta et al.'s study of IT professionals who are involved in security management [42]. The study found that security professionals preferred tools that exhibit easy to understand correlations between the data and the tool's decisions. All tools except two had no contact with system administrators to identify their needs before developing the tool. Only the V3SPA tool by Robert Gove [21] conducted a small survey with policy developers to identify their experience with SELinux policies with six questions. Only one other tool outlined the workings of policy administrators and the

TABLE 6. Comparison of the user studies to evaluate the tool.

OPERATIONS / AC TOOLS	ADAGE [7]	Impromptu [20]	Salmon [22]	Grey [6]	IAM [1]	SPARCLE [27]	Expandable Grids [22]	Expandable Grids for P3P [31]	FB SNS [17]	FB SNS RPA [19]	V3SPA [21]	PVA Tool - SELinux [23]	DTEVisual [34]	MLSVisual [29]	RBACVisual [30]	UNIXVisual [32]	Audience View [33]	PViz [38]
Survey with Policy Architects	🟩										🟩							
User Study	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩		🟩	🟩	🟩	🟩	🟩	🟩	🟩
PARTICIPANT DETAILS																		
Total Sample Size	5		24		10	26	36	532	65	36				22		21	28	20
Participant Profile																		
System administrators	🟩											🟩						
Working professionals					🟩													
Students or Staff		🟩				🟩		🟩	🟩		🟩	🟩	🟩	🟩	🟩	🟩		🟩
USER STUDY DETAILS																		
Rounds of study		3	1		1	1	1	2		1		1	1	1	1	1	1	1
Purpose of User Study																		
Eval. a component of the tool						🟩		🟩										
Visual Analysis									🟩	🟩								
Interactive Analysis	🟩	🟩	🟩		🟩	🟩	🟩	🟩				🟩	🟩	🟩	🟩	🟩	🟩	🟩
Compare tool with a related tool			🟩				🟩	🟩				🟩						🟩

cognitive load that they deal with. But they had no formal interviews with policy professionals. Another drawback was that the tools were not built using actual data collected from organisations. This can be attributed to the fact that actual access control policy data is hard to obtain due to the their sensitive nature. Hence, the data used by the tools was generated to mimic specific scenarios that are found in access control policies. However, these generated policies are not as complete and complex as original access control policies and these makes the tools somewhat ineffective. Finally, most systems lacked a user study to evaluate their tool. Even if a user study was conducted, most tools were evaluated using students and not actual system professionals who would be using the tools.

It's clear that the tools are built on assumptions of the user's capabilities and the functions that they would need to perform their tasks. This minimal involvement with the end-users is one of the reasons why none of the tools recognised that there are two types of policy professionals involved in the creation and implementation of policies. The recognition of the policy makers and implementers would've helped researchers to build role-specific functionalities for the policy makers and implementers, and the collaboration problems that they face.

VII. DISCUSSION

In this work, we set out to investigate the problem of how well do current Access Control tools provide support for Policy Makers, Policy Implementers, and effective collaboration between Policy Makers and Policy Implementers. This problem is articulated in greater depth in the research questions in section III. Briefly, the research questions we are addressing in this work are: (1) How well do current access control policy authoring tools support Policy Makers towards eliciting good quality policy requirements?; (2) How well do current tools for access control policy implementation support Policy Implementers towards implementation of good quality policies?; and (3) How well do current tools support the activities for effective collaboration between policy makers and policy implementers?

The discussion in section IV-C has shown that current AC policy tools have limitations toward supporting effective policy collaboration. In particular, the results in Table 5 indicates that policy inconsistency is a common limitation of AC tools in all three aspects of the policy collaboration problem (policy making, policy collaboration, and policy implementation). For policy making, Incompleteness and Requirements Update are the additional issues that have the least support from current AC tools. The challenges of policy collaboration discussed in the preceding sections suggests that the design of current AC tools needs significant improvement in order to make these tools useful at facilitating effective collaboration between policy makers and policy implementers. To this end, we propose some pointers for further development of the capabilities of AC tools towards supporting effective collaboration.

1) *Policy Requirement Update Rationale Capture*: As discussed in the introduction, the rationale for the proposed specific policy requirements by the policy making team is often not relayed to the policy implementation team. Communicating such rationale details

would ensure that the policy implementation does not only receive required updates to the specifications of policies but also understand the rationale behind the updates or proposal of new policy requirements. AC tools need to have the capability to capture rationales for updates occurring in both policy requirements and policy specifications. This will address the problem of communication within and across policy making and policy implementation teams. The idea is that if a member of the policy making team makes an update to a policy requirement, in addition to the update being systematically captured in a log by the tool, it should possible for the professional making the update record the rationale for the update along with the update. This is so because another member of the same team who wants to make further updates should be warned about previous updates that he/she may not be aware of and the rationales documenting why the updates were made. This should minimize inconsistencies in the policy updates. The information about what updates have been made, to which policy and the reasoning on why the updates were made, should be readily available, and the tool must have a mechanism to automatically notify the relevant professionals in the policy making team about the updates and their rationales. Furthermore, notifications about such policy updates to policy requirements should be relayed to the policy implementation team so that they can make the necessary updates to the relevant policy specifications based on the rationales captured in the requirements update. The same should apply to the policy implementation team.

2) *Policy Implementation Knowledge Capture*: Policy makers lack knowledge about how a particular policy is implemented and whether it was implemented as intended in the policy requirements. This knowledge is only available to the policy implementation team. A mechanism for systematically capturing the implementation knowledge and sharing with the policy making team would be useful towards addressing this challenge. Having the implementation knowledge would help policy makers in avoiding policy duplication - where a policy that was already implemented is proposed for implementation again.
3) *Technical Language Interfacing Notation*: The root of the collaboration problems between policy makers and policy implementers is because they use different technical languages. This leads to misunderstandings and miscommunication between them. In order to address this challenge, there is need for a technical language translation interface which translates between PI_L and PM_L. This would ensure that any information being communicated can be translated to the technical language that can be understood by the recipient of the message - thus avoiding miscommunication. An alternative approach to address this challenge is by formulating a language based on a notation that can be easily understood by both policy makers and implementers. Regardless of the approach taken, the key idea is to ensure that the semantics of messages relayed during collaboration between policy makers and implementers is not lost - thus minimizing the possibility of miscommunication.
4) *Policy Conflict Detection and Resolution*: There is need for support in detection of conflicts between policies at the policy making, policy implementation, and policy collaboration stages. As stated earlier, at the policy making stage, conflict could arise because of inconsistency between policy requirements. At the implementation stage, conflict could arise between different implementations of the same policy or implementations of different policies. Conflicts arise at the collaboration stage when there is an inconsistency between policy requirements and the implementation of a policy.
5) *Forward and Backward Traceability Capture*: Current tools do not have a systematic approach for tracing/mapping between policy requirements and specifications (i.e.) given a policy requirement P_{Rx} it is not easy to answer the question of which policy specification P_{Sy} it corresponds to - *Forward Traceability*. The lack of traceability information makes maintaining policies a challenge. Similarly, in the reverse direction, if a policy specification, P_{Sy} is updated, there is no systematic way to know which set of policy requirements are affected by the update - *Backward Traceability*. Having a policy *traceability database* can help in systematically mapping between policy requirements and specifications for both *forward* and *backward* traceability.
6) *Policy Update Synchronization*: When an update is made to a policy requirement, i.e., P_R changes to P'_R, there is a need to have the ability to make the same update in the corresponding policy specification, i.e., P_S changes to P'_S. The ability to make synchronised updates would ensure that changes in policy making are always synchronized with the corresponding changes in policy implementation. The information captured in the traceability database would be useful in identifying the policies that need to be updated in order to keep requirements synchronised to their specifications.

In the context of access control, in order to get a better appreciation of the practicalities of the capabilities identified above, future studies should focus on interviewing policy professionals (makers and implementers) in order to gain a better understand of the tasks that they perform before developing the functionalities of the required AC tools.

VIII. CONCLUSION

In this paper, we conducted a systematic review of existing AC tools and identified their limitations. We argued that these limitations are the genesis of the policy collaboration problem - challenges that hinder effective collaboration between policy makers and policy implementers. We articulated the challenges and identified directions for future research in the development of AC tools. We also identified the minimal set of operations that should be supported by the access control tool. We have aimed to set a roadmap that will be helpful towards informing researchers in the field of access control policies on the key challenges in collaboration for policy making and implementation. A common drawback that most of the tools faced was the failure to identify the different types of policy professionals who are involved in the process of policy creation and implementation. Hence, future research should focus on identifying the tasks that each policy professionals is responsible for before designing tools. The tools should focus on bridging the communication gap between the non-technical policy makers and the technical policy implementers. While some tools capitalized on simple visualizations to convey the structure of the policies, the potential of visualizations remain unexplored to a large extent. Though the tools utilized different types of visualizations to reduce the cognitive load on the users while analyzing the policies, the potential of the visualizations lies in their ability to address the AC policy collaboration problem. Hence, future researches on AC tools should aim to exploit the capabilities of the visualizations to bridge the gap between the non-technical policy makers and the technical policy implementers.

ACKNOWLEDGMENT

This research is made possible by National Priorities Research Program (NPRP) grant No. 12S-0115-190002 from the Qatar National Research Fund (a member of Qatar Foundation). The publication of this article was funded by Qatar National Library. The statements made herein are solely the responsibility of the authors.

REFERENCES

[1] X. Cao and L. Iverson, "Intentional access management: Making access control usable for end-users," in *Proc. 2nd Symp. Usable Privacy Secur.*, 2006, pp. 20–31.

[2] K. Vaniea, Q. Ni, L. Cranor, and E. Bertino, "Access control policy analysis and visualization tools for security professionals," in *Proc. SOUPS Workshop (USM)*, 2008, pp. 7–15.

[3] D. K. Smetters and N. Good, "How users use access control," in *Proc. 5th Symp. Usable Privacy Secur.*, Jul. 2009, pp. 1–12.

[4] S. K. Lam and E. Churchill, "The social web: Global village or private cliques?" in *Proc. Conf. Designing User eXperiences*, Nov. 2007, pp. 1–7.

[5] S. Ahern, D. Eckles, N. S. Good, S. King, M. Naaman, and R. Nair, "Over-exposed?: Privacy patterns and considerations in online and mobile photo sharing," in *Proc. SIGCHI Conf. Hum. Factors Comput. Syst.*, Apr. 2007, pp. 357–366.

[6] L. Bauer, S. Garriss, J. M. McCune, M. K. Reiter, J. Rouse, and P. Rutenbar, "Device-enabled authorization in the grey system," in *Proc. Int. Conf. Inf. Secur.* Cham, Switzerland: Springer, 2005, pp. 431–445.

[7] E. C. Center, "Adage system overview," Tech. Rep.,.

[8] L. Bauer, L. F. Cranor, R. W. Reeder, M. K. Reiter, and K. Vaniea, "Real life challenges in access-control management," in *Proc. SIGCHI Conf. Hum. Factors Comput. Syst.*, Apr. 2009, pp. 899–908.

[9] T. Whalen, D. Smetters, and E. F. Churchill, "User experiences with sharing and access control," in *Proc. CHI Extended Abstr. Hum. Factors Comput. Syst.*, Apr. 2006, pp. 1517–1522.

[10] J. H. Manley, "Software engineering provisioning process," in *Proc. 8th Int. Conf. Softw. Eng.*, Washington, DC, USA, 1985, pp. 273–284.

[11] S. Maynard, A. Ruighaver, and A. Ahmad, "Stakeholders in security policy development," in *Proc. 9th Austral. Inf. Secur. Manag. Conf.*, 2011, pp. 1–8.

[12] S. V. Flowerday and T. Tuyikeze, "Information security policy development and implementation: The what, how and who," *Comput. Secur.*, vol. 61, pp. 169–183, Aug. 2016.

[13] N. Sadeh, "Understanding and capturing people's privacy policies in a mobile social networking application," *Pers. Ubiquitous Comput.*, vol. 13, no. 6, pp. 401–412, Aug. 2009.

[14] D. Balfanz, "Usable access control for the world wide web," in *Proc. 19th Annu. Comput. Secur. Appl. Conf.*, 2003, pp. 406–415.

[15] P. Rao, G. Ghinita, E. Bertino, and J. Lobo, "Visualization for access control policy analysis results using multi-level grids," in *Proc. IEEE Int. Symp. Policies Distrib. Syst. Netw.*, Jul. 2009, pp. 25–28.

[16] D. F. Ferraiolo, R. Chandramouli, G.-J. Ahn, and S. I. Gavrila, "The role control center: Features and case studies," in *Proc. 8th ACM Symp. Access Control Models Technol.*, Jun. 2003, pp. 12–20.

[17] M. L. Johnson, "Toward usable access control for end-users: A case study of Facebook privacy settings," Ph.D. dissertation, Columbia Univ., 2012.

[18] M. C. Mont, R. Thyne, and P. Bramhall, "Privacy enforcement with hp select access for regulatory compliance," HP Laboratories Bristol, Bristol, U.K., Tech. Rep., HPL-2005-10, 2005.

[19] M. Anwar and P. W. L. Fong, "A visualization tool for evaluating access control policies in Facebook-style social network systems," in *Proc. 27th Annu. ACM Symp. Appl. Comput.*, Mar. 2012, pp. 1443–1450.

[20] R. De Paula, X. Ding, P. Dourish, K. Nies, B. Pillet, D. F. Redmiles, J. Ren, J. A. Rode, and R. S. Filho, "In the eye of the beholder: A visualization-based approach to information system security," *Int. J. Hum.-Comput. Stud.*, vol. 63, nos. 1–2, pp. 5–24, Jul. 2005.

[21] R. Gove, "V3SPA: A visual analysis, exploration, and diffing tool for SELinux and SEAndroid security policies," in *Proc. IEEE Symp. Visualizat. Cyber Secur. (VizSec)*, Oct. 2016, pp. 1–8.

[22] R. W. Reeder, L. Bauer, L. F. Cranor, M. K. Reiter, K. Bacon, K. How, and H. Strong, "Expandable grids for visualizing and authoring computer security policies," in *Proc. SIGCHI Conf. Hum. Factors Comput. Syst.*, Apr. 2008, pp. 1473–1482.

[23] W. Xu, M. Shehab, and G.-J. Ahn, "Visualization-based policy analysis for SELinux: Framework and user study," *Int. J. Inf. Secur.*, vol. 12, no. 3, pp. 155–171, Jun. 2013.

[24] S. Marouf and M. Shehab, "SEGrapher: Visualization-based SELinux policy analysis," in *Proc. 4th Symp. Configuration Anal. Autom. (SAFECONFIG)*, Oct. 2011, pp. 1–8.

[25] P. Li, L. Ning, and Z. Xiaochao, "Visualization framework for inter-domain access control policy integration," *Commun., China*, vol. 10, no. 3, pp. 67–75, Mar. 2013.

[26] J. Montemayor, A. Freeman, J. Gersh, T. Llanso, and D. Patrone, "Information visualization for rule-based resource access control," in *Proc. Int. Symp. Usable Privacy Secur. (SOUPS)*, 2006, p. 24.

[27] C. A. Brodie, C.-M. Karat, and J. Karat, "An empirical study of natural language parsing of privacy policy rules using the SPARCLE policy workbench," in *Proc. 2nd Symp. Usable Privacy Secur.*, 2006, pp. 8–19.

[28] B. Salmon, S. W. Schlosser, L. B. Mummert, and G. R. Ganger, "Putting home storage management into perspective," Carnegie Mellon Univ. Parallel Data Lab, Pittsburgh, PA, USA, Tech. Rep., 2006.

[29] M. Wang, S. Carr, J. Mayo, C.-K. Shene, and C. Wang, "MLSvisual: A visualization tool for teaching access control using multi-level security," in *Proc. Conf. Innov. Technol. Comput. Sci. Educ.*, 2014, pp. 93–98.

[30] M. Wang, J. Mayo, C.-K. Shene, T. Lake, S. Carr, and C. Wang, "RBACvisual: A visualization tool for teaching access control using role-based access control," in *Proc. ACM Conf. Innov. Technol. Comput. Sci. Educ.*, Jun. 2015, pp. 141–146.

[31] R. W. Reeder, P. G. Kelley, A. M. McDonald, and L. F. Cranor, "A user study of the expandable grid applied to P3P privacy policy visualization," in *Proc. 5th Symp. Usable Privacy Secur.*, Jul. 2009, pp. 45–54.

[32] M. Wang, J. Mayo, C.-K. Shene, S. Carr, and C. Wang, "UNIXvisual: A visualization tool for teaching UNIX permissions," in *Proc. ACM Conf. Innov. Technol. Comput. Sci. Educ.*, Jun. 2017, pp. 194–199.

[33] J. Watson, M. Whitney, and H. R. Lipford, "Configuring audience-oriented privacy policies," in *Proc. 2nd ACM Workshop Assurable Usable Secur. Configuration*, Nov. 2009, pp. 71–78.

[34] Y. Li, S. Carr, J. Mayo, C.-K. Shene, and C. Wang, "DTEvisual: A visualization system for teaching access control using domain type enforcement," *J. Comput. Sci. Colleges*, vol. 28, no. 1, pp. 125–132, 2012.

[35] K. K. Singh, R. B. Radhika, and R. K. Shyamasundar, "SEFlowViz: A visualization tool for SELinux policy analysis," in *Proc. 12th Int. Conf. Inf. Commun. Syst. (ICICS)*, May 2021, pp. 439–444.

[36] C. Lawson and F. Zhu, "Sentential access control," in *Proc. 50th Annu. Southeast Regional Conf.*, Mar. 2012, pp. 303–308.

[37] A. Ferreira, G. Lenzini, C. Santos-Pereira, A. B. Augusto, and M. E. Correia, "Envisioning secure and usable access control for patients," in *Proc. IEEE 3rd Int. Conf. Serious Games Appl. Health (SeGAH)*, May 2014, pp. 1–8.

[38] A. Mazzia, K. LeFevre, and E. Adar, "The PViz comprehension tool for social network privacy settings," in *Proc. 8th Symp. Usable Privacy Secur.*, Jul. 2012, pp. 1–12.

[39] M. Wang, "Accessible access control: A visualization system for access control policy management," Ph.D. dissertation, 2019.

[40] H. R. Lipford, J. Watson, M. Whitney, K. Froiland, and R. W. Reeder, "Visual vs. compact: A comparison of privacy policy interfaces," in *Proc. SIGCHI Conf. Hum. Factors Comput. Syst.*, Apr. 2010, pp. 1111–1114.

[41] A. Heitzmann, B. Palazzi, C. Papamanthou, and R. Tamassia, "Effective visualization of file system access-control," in *Proc. Int. Workshop Vis. Comput. Secur.* Cham, Switzerland: Springer, 2008, pp. 18–25.

[42] D. Botta, R. Werlinger, A. Gagné, K. Beznosov, L. Iverson, S. Fels, and B. Fisher, "Towards understanding IT security professionals and their tools," in *Proc. 3rd Symp. Usable Privacy Secur.*, Jul. 2007, pp. 100–111.

RACHAEL FERNANDEZ is currently pursuing the Ph.D. degree with the University of Sussex. She is currently a Researcher with Qatar University. She works in the domain of Information Visualization and has created several visualizations based on cybersecurity data.

PETER C.-H. CHENG is currently a Professor of cognitive science with the University of Sussex. He heads the Representational Systems Laboratory, Creative Technology Research Group. His research interests include the cognitive science of representational systems, complex problem solving, conceptual learning, processes of writing and drawing, competence measurement, and cognitive profiling. He was formerly the Chair of the Cognitive Science Society.

ARMSTRONG NHLABATSI received the Ph.D. degree in computer science from Open University, U.K. He is currently a Research Associate with the KINDI Computing Research Centre, Qatar University. He has contributed to security research and innovation in adaptive security for the cloud, threat-specific security risk evaluation, quantification of satisfaction of security requirements, and information security for sports accreditation systems. He is currently working on approaches to analyzing emergent security properties of systems of socio-technical systems and approaches to visualizing security policies. His research interests include security requirements engineering, security risk evaluation, requirements traceability, and the feature interaction problem for information security.

KHALED MD. KHAN (Senior Member, IEEE) received the B.S. and M.S. degrees from the Norwegian University of Science and Technology, and the Ph.D. degree from Monash University. He is currently an Associate Professor of computer science and engineering with Qatar University. Prior to these, he served at Western Sydney University, Australia, and the Head of the graduate programs.

NOORA FETAIS (Senior Member, IEEE) is currently an Associate Professor with the Department of Computer Science and Engineering, Qatar University. She is a certified Innovation Leader and a fellow with The Institute of Innovation and Knowledge Exchange, U.K. She is the founding Secretary-General of the Arab Association for Cyber Security, a member of the U.K.–Gulf Women in Cyber Fellowship Program, and a Mentor at the ITU (UN) Women in Cyber Mentorship Program 2022. She is a Senior Member of ACM and a member of other different international professional associations such as INSTICC, ASEE, among others. She was awarded the NATO-SPS Grant to support her "CyberWomen" initiative in Qatar for NATO and partner countries.