Toward the Foundation of Digital Identity Theory

Pierre Fobougong Saha[1], Mohamed Mejri[1] and Kamel Adi[2]
[1]*Laval University, Québec, Canada*
[2]*University of Quebec in Outaouais, Québec, Canada*

Keywords: Verifiable Credentials, Authentication, Privacy, Pruduct Family Algebra, Satisfiability Modulo Theories.

Abstract: Verifiable Credentials (VC) have become today a de facto digital credentials standard and play an increasingly important role in network exchanges. They often contain a large number of attributes that disclosure could have unfortunate consequences. Formally verifying whether the user can access the requested service and ensuring that their disclosed identity attributes generate the slightest risk, therefore, becomes very important. In this paper, using the product family algebra, we show how the consideration of verifiable credentials can help to easily and formally settle the question of whether a user can access a service and to respect the need-to-know principle. To this end, we propose a translation of product family algebra into first-order Boolean logic and vice versa. We then propose a Boolean equivalence of the product family algebra refinement operation. Using all these tools, we show how the problem of verifying a user's ability to authenticate, expressed using product family algebra, easily translates into an SMT problem. In order to guarantee the preservation of privacy and ensure the need-to-know principle, we associate VC attributes with a risk score and show how the question of disclosing the attributes generating the least risk can easily be resolved with Maximum Weighted SMT. So we can easily use the z3 solver to solve these problems in SMT form.

1 INTRODUCTION

In our daily life, we use various identity documents to access services provided by different providers. This may be a health insurance card, a national identity card, a driving license, a study certificate, a diploma, etc. The concept of verifiable credentials (VC) (Lim et al., 2021; Xu et al., 2023; Li et al., 2022; Manu et al., 2024), introduced by the W3C, tends to reproduce these identity documents in a digital form that can be easily stored in a digital wallet. Users, therefore, are in possession of a large number of VCs among which they must be able to prove their identity to different service providers while running as little risk as possible.

Although there is a growing body of scientific work on the issue of privacy, very little of them addresses the need for a user-centric approach. However, it is clear that with the emergence of decentralized technologies and self-sovereign identities, the user plays a key role in data protection. Even if cryptography, in particular zero-knowledge proof, have become good tools for preserving privacy, the fact remains that they are complex and cannot always prove everything. In this research, we postulate that assigning risk scores to a user's identity attributes can enhance their privacy protection by disclosing only those identity attributes necessary for accessing the requested service. Additionally, this reinforces adherence to the 'need-to-know' principle from the user's perspective.

We show in this paper how the verification of a user's ability to access a service and the determination of attributes whose disclosure is likely to generate the minimum risk can be easily solved using product family algebra (PFA) (Höfner et al., 2011) and Satisfiability Modulo Theories (SMT) (de Moura et al., 2007) or Max weighted SMT.

1.1 Contributions

In this paper, we use of formal tools such as the product family algebra and SMT (Satisfiability Modulo Theories) to answer two essential questions in accessing online services: 1 - Can a user's Verifiable Credentials be used to access a service ? 2 - What is the subset of VCs attributes available to the user that he can disclose while incurring the minimum risk ? Thus, our main contributions are as follows:

- A translation of product family algebra into first-order Boolean logic. Beyond helping us answer

Saha, P., Mejri, M. and Adi, K.
Toward the Foundation of Digital Identity Theory.
DOI: 10.5220/0012838100003767
Paper published under CC license (CC BY-NC-ND 4.0)
In *Proceedings of the 21st International Conference on Security and Cryptography (SECRYPT 2024)*, pages 812-819
ISBN: 978-989-758-709-2; ISSN: 2184-7711

the questions posed in this research, this translation can enable the rapid and reliable detection of errors or conflicts in product configurations, such as incompatible feature combinations or unfulfilled dependencies.

- A new methodological framework to enhance privacy and respect for the need-to-know principle in a user-centric approach.
- A formal specification of the question of whether a user can access a service and the set of attributes whose disclosure generates the minimum risk and the automatic resolution of these questions by the Z3 SMT tool.

1.2 Outline

In Section 2, we briefly present Verifiable Credentials, Product Family Algebra, Satisfiability Modulo Theories (SMT) and the motivation behind this research. Section 3 presents the proposed methodological framework. Section 4 summarizes how we translate the VC into a product family. Section 5 focuses on defining the mathematical tools we intend to use in our approach. Sections 6 and 7 deals with the implementation of our methodological framework. Section 8 summarises the existing literature on preserving privacy by taking risk into account in a user-centric approach. Finally, section 9 presents a discussion of our results.

2 BACKGROUND AND MOTIVATION

2.1 Verifiable Credentials and VC Product Family

Verifiable credentials, as defined by the (Manu et al., 2024), consist of tamper-evident claims about a subject made by an issuer. These credentials are pivotal for digital identity management and certification, leveraging W3C standards, blockchain, and decentralized identifiers (Manu et al., 2022). Issued by an identity provider or self-generated, these credentials are stored in a digital wallet, which can compile them into verifiable presentations for submission to service providers.

In this paper, we consider all VCs or VPs as product families, akin to real-life identity documents. Despite variations in document type and issuing authorities, common attributes like names and birth dates remain consistent for the same individual. Thus, multiple identity documents share both commonalities and variability, with each document regarded as a product family.

2.2 Product Family Algebra (PFA)

Introduced by Höfner et al. (Höfner et al., 2011), a product family algebra is a mathematical framework based on idempotent semiring that provides a formal way to represent and work with product families. This algebraic structure allows for the expression of common concepts used in the product family paradigm, such as how products are composed, the variability of features, and the relationships between different product families. For more information, we refer readers to (Höfner et al., 2011).

Since verifying that a user can access a service requires that he has the attributes required by the service provider, we assume that if the user has a product family VC and the service provider requires at least one product from a product family R, then the chosen subset of VC attributes contains all the attributes of some products of R.

2.3 Satisfiability Modulo Theories (SMT)

SMT solvers are pivotal in addressing verification problems due to their capability to efficiently manage complex logical constraints and theories (de Moura et al., 2007; De Moura and Bjørner, 2011). The process involves encoding the problem as a first-order formula incorporating operations from various theories, such as Boolean logic, bit-vectors, arithmetic, arrays, and recursive datatypes. The SMT solver then determines the formula's satisfiability using advanced algorithms for SAT solving and theory-specific decision procedures. In our approach, we integrate the theories of linear arithmetic and Boolean logic to verify user authentication possibilities. Employing the Z3 SMT solver, we obtain either a model or an explanation for the formula's satisfiability.

2.4 Motivation

Verifiable credentials simplify the replication of physical identity methods. Users can gather various identity documents from different providers and use their attributes to access services. This allows users to choose which identity to present, similar to real life. However, users can make mistakes and may unintentionally disclose sensitive information. This research aims to help users maintain control over their identity attributes, ensuring they can access services while minimizing the risk of unauthorized exploitation.

3 METHODOLOGICAL FRAMEWORK

The methodology followed in this research is illustrated in Fig. 1. We begin by translating verifiable credentials into product family algebra and then use an equivalent Boolean logic definition to convert this algebra into a Boolean formula. Our two questions are formally specified using the logical equivalent of the refinement operator, facilitating easy resolution. Finally, we translate our formulations into a format compatible with the SMT solver Z3 for automated resolution. When authentication is successful, the output is a model of identity attributes required for authentication. However, to enhance the approach, we might need to determine the necessary identity attributes if the solver returns ”UNSAT”.

4 TRANFORMING VC TO PFA EXPRESSION

One of the first steps is to translate the Verifiable Credentials (VC) or Verifiable Presentation (VP) into a product family algebra expression. To do this, we use the W3C specification (Manu et al., 2024) and represent the VC or VP as a Feature Tree.

Based on the transformation proposed in (Höfner et al., 2011), we can easily transform our Feature Tree into the corresponding Product Family Algebra expression.

5 PRELIMINARIES

Let us consider $\mathbb{F}$ a set of features and $\mathbb{P} =_{df} \mathcal{P}(\mathbb{F})$ a set of all possible product (knowing that a product is a set of features). A collection $\mathcal{P}(\mathbb{P})$ is called a product family.

Definition 1. *[$[\![\ ]\!]_{\mathbb{F}}$] $[\![\ ]\!]_{\mathbb{F}}$ is the transformation function of a product family into its Boolean equivalent in the Disjunctive Normal Form (DNF). More formally $[\![\ ]\!]_{\mathbb{F}}$ is defined as follow:*

$$[\![\ ]\!]_{\mathbb{F}} : \mathcal{P}(\mathbb{P}) \longrightarrow (\mathbb{B}, \wedge, \vee)$$

$$[\![T]\!]_{\mathbb{F}} = \begin{cases} False & if\ T = \emptyset = 0 \\ \bigwedge\limits_{f \in \mathbb{F}} \neg f & if\ T = \{\emptyset\} = 1 \\ \bigvee\limits_{t \in T} (\bigwedge\limits_{f \in t} f \bigwedge\limits_{f' \in \mathbb{F} \setminus t} \neg f') & else \end{cases}$$

Logically, we might be inclined to consider that for a service provider (verifier), requesting attribute a from a set of attributes $\{a, b\}$ does not mean that he doesn’t want attribute b. This consideration would lead us to slightly modify the proposed transformation. However, in the case of identity management, we rely on the *”Need To Know”* principle to consider that requesting attribute a does not authorize providing a and b.

subsequently, we introduce Definition 2 and Definition 3 for the reconstruction of the product family knowing the Boolean formula.

Definition 2. *[$\mathcal{A}(\varphi)$, $\mathcal{A}^+(\varphi)$, $\mathcal{A}^-(\varphi)$] Let φ be a boolean formula in DNF form. $\mathcal{A}(\varphi)$, $\mathcal{A}^+(\varphi)$ and $\mathcal{A}^-(\varphi)$ denote respectively the set of all atoms, positive atoms and negative atoms in φ. More formally $\mathcal{A}()$, $\mathcal{A}^+()$ and $\mathcal{A}^-()$ are defined as follows:*

$$\mathcal{A}^+$$

$$\begin{aligned} \mathcal{A}^+(a) &= \{a\} \\ \mathcal{A}^+(\neg a) &= \{\ \} \\ \mathcal{A}^+(\varphi_1 \vee \varphi_2) &= \mathcal{A}^+(\varphi_1) \cup \mathcal{A}^+(\varphi_2) \\ \mathcal{A}^+(\varphi_1 \wedge \varphi_2) &= \mathcal{A}^+(\varphi_1) \cup \mathcal{A}^+(\varphi_2) \end{aligned}$$

$$\mathcal{A}^-$$

$$\begin{aligned} \mathcal{A}^-(a) &= \{\} \\ \mathcal{A}^-(\neg a) &= \{\ a\} \\ \mathcal{A}^-(\varphi_1 \vee \varphi_2) &= \mathcal{A}^-(\varphi_1) \cup \mathcal{A}^-(\varphi_2) \\ \mathcal{A}^-(\varphi_1 \wedge \varphi_2) &= \mathcal{A}^-(\varphi_1) \cup \mathcal{A}^-(\varphi_2) \end{aligned}$$

$$\mathcal{A}(\varphi) = \mathcal{A}^+(\varphi) \cup \mathcal{A}^-(\varphi)$$

Where a is an atomic formula, φ_1 and φ_2 are boolean formula in DNF form.

Definition 3. *[$[\![\ ]\!]_{\mathbb{F}}^{-1}$] $[\![\ ]\!]_{\mathbb{F}}^{-1}$ is the inverse function of $[\![\ ]\!]_{\mathbb{F}}$, which transforms a Boolean formula expressed in Disjunctive Normal Form (DNF) into an equivalent product family. More formally:*

$$[\![\ ]\!]_{\mathbb{F}}^{-1} : (\mathbb{B}, \wedge, \vee) \longrightarrow \mathcal{P}(\mathbb{P})$$

$$[\![\bigvee \varphi_i]\!]_{\mathbb{F}}^{-1} = \bigcup_{\varphi_i} [\![\varphi_i]\!]_{\mathbb{F}}^{-1}$$

$$[\![\varphi_i]\!]_{\mathbb{F}}^{-1} = \begin{cases} \emptyset & if\ \mathcal{A}^+(\varphi_i) \cap \\ & \mathcal{A}^-(\varphi_i) \neq \emptyset \\ \mathcal{A}^+(\varphi_i) \times 2^{\mathbb{F} \setminus \mathcal{A}(\varphi_i)} & else \end{cases}$$

With φ_i a conjunction of atomic Boolean variables.

The product family algebra operator of interest is the refinement operator (Definition 6). Checking whether a user can authenticate himself boils down to determining whether a product available to the user refines the product family claimed by the service provider. In the definitions that follow, we propose transformations that allow us to express this formulation with logical formulas. Since the refinement re-

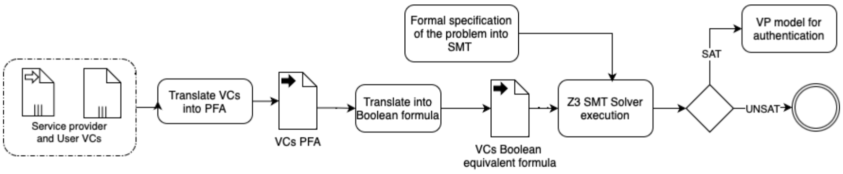

Service provider and User VCs
Translate VCs into PFA
VCs PFA
Translate into Boolean formula
VCs Boolean equivalent formula
Formal specification of the problem into SMT
Z3 SMT Solver execution
SAT
UNSAT
VP model for authentication

Figure 1: Methodological framework.

lation is a partial order, we take this into account by integrating the Definition 5.

Definition 4. *[$\langle\ \rangle^+$] Let P be a Boolean formula in DNF form. $\langle P\rangle^+$ is the transformation of P into another Boolean formula containing only its positive atoms. More formally $\langle\ \rangle^+$ is defined as follows:*

$$\underline{\underline{\langle\ \rangle^+ : (\mathbb{B},\wedge,\vee) \longrightarrow (\mathbb{B},\wedge,\vee)}}$$

$$\begin{array}{rcl} \langle a\rangle^+ & = & a \\ \langle\neg a\rangle^+ & = & True \\ \langle P\wedge Q\rangle^+ & = & \langle P\rangle^+ \wedge \langle Q\rangle^+ \\ \langle P\vee Q\rangle^+ & = & \langle P\rangle^+ \vee \langle Q\rangle^+ \end{array}$$

Where a is an atomic formula, P and Q are Boolean formulas.

Definition 5. *[$\preccurlyeq$] Let φ_1 and φ_2 be two Boolean formulas. We define a partial order relation $\preccurlyeq$ between φ_1 and φ_2 as following: $\varphi_1 \preccurlyeq \varphi_2$ if $\langle\varphi_1\rangle^+ \implies \langle\varphi_2\rangle^+$; Where $\implies$ is the Boolean implication.*

Definition 6. *[$\sqsubseteq$] Let P and Q be two product family in $\mathcal{P}(\mathbb{P})$. We define the refinement between P and Q, denoted by $P \sqsubseteq Q$, as following: $P \sqsubseteq Q$ if $[\![P]\!]_{\mathbb{F}} \preccurlyeq [\![Q]\!]_{\mathbb{F}}$.*

The refinement operator as defined by PFA ($P \sqsubseteq Q$) requires that each product in the product family (P) contains all features of some product in the product family (Q). In the context of authentication verification, this definition is not satisfactory, as the product family corresponding to the user's identity could contain a single identity that contains the set of attributes of some products in the product family claimed by the service provider. To solve this problem, we introduce, in Definition 7, the notion of partial refinement ($P \sqsubseteq_{\wp} Q$); understood here to mean that some products of P contain all the features of some products of Q.

Definition 7. *[$\sqsubseteq_{\wp}$] Let P and Q be two product family in $\mathcal{P}(\mathbb{P})$. We define the partial refinement between P and Q, denoted by $P \sqsubseteq_{\wp} Q$, as following: $P \sqsubseteq_{\wp} Q$ if there exist a product R in $\mathcal{P}(\mathbb{F})$ such that $R \leq P$ and $R \sqsubseteq Q$.*

Finally, we define (Definition 8) an extension of the refinement operation which consists, where possible, in finding a substitution of the values of different features such that the refinement is true. This definition is motivated by the fact that, in general, the service provider doesn't know the values of the attributes, but simply the attributes.

Definition 8. *[$\sqsubseteq_\Gamma$] Let P, Q be two product family in $\mathcal{P}(\mathbb{P})$ and Γ be a set of substitutions. We define the refinement modulo a substitution in Γ between P and Q, denoted by $P \sqsubseteq_\Gamma Q$, as following: $P \sqsubseteq_\Gamma Q$ if there exists a substitution σ in Γ such that $\sigma(P) \sqsubseteq \sigma(Q)$.*

6 AUTHENTICATION VERIFICATION

We consider two scenarios: a simple case where the service provider simply requests identity attributes and a more realistic case where the service provider requests a product family with attributes as variables.

6.1 Case Without Variables

6.1.1 Problem Description

The question is : *Given a user with a set of verifiable credentials and a service provider request, can the user authenticate to the service provider?*

6.1.2 Specification with PFA

Given the user's product family VC and the product family R claimed by the service provider, checking whether the user can authenticate is the same as checking whether VC partially refines R i.e. $VC \sqsubseteq_{\wp} R$.

6.1.3 Specification with SMT

Based on our above definitions, checking whether a user can authenticate is equivalent to checking if $\langle [\![VC]\!]_{\mathbb{F}}\rangle^+ \implies \langle [\![R]\!]_{\mathbb{F}}\rangle^+$ is satisfiable.

Example 1. *Let $\mathbb{F} = \{a,b,c,d,e,f,g,h\}$;*

$$\begin{array}{rcl} VC & = & a\cdot b\cdot c + d\cdot e\cdot f \\ R & = & a\cdot b + e\cdot f + g\cdot h \end{array}$$

Reasoning with PFA

Question : $VC \sqsubseteq_{\wp} R$?

*Based on the PFA definition of $\sqsubseteq_{\wp}$, the answer is **True** since there exists $U = d + c$; $U \in \mathcal{P}(2^{\mathbb{F}})$ such that*

$VC \leq R \cdot U$.

Reasoning with SMT

Translation into boolean formula:

$$\begin{aligned}\langle[\![VC]\!]_{\mathbb{F}}\rangle^+ &= (a \wedge b \wedge c) \vee (d \wedge e \wedge f)\\ \langle[\![R]\!]_{\mathbb{F}}\rangle^+ &= (a \wedge b) \vee (e \wedge f) \vee (g \wedge h)\end{aligned}$$

Equivalent question in SMT : Is $\langle[\![VC]\!]_{\mathbb{F}}\rangle^+ \implies \langle[\![R]\!]_{\mathbb{F}}\rangle^+$ satisfiable ?

Verification with Z3 shows that the Boolean formula is satisfiable and the user can authenticate.

6.2 Case with Variables

6.2.1 Problem

The question here is: *Given a user with verifiable credentials and a provider's request, what values must the user assign to the identity attributes to authenticate successfully?.*

6.2.2 Specification with PFA

Given the user's product family VC and the product family R claimed by the service provider, find the substitution σ such that $\sigma(VC) \sqsubseteq \sigma(R)$.

6.2.3 Specification with SMT

Given a $VC = \bigvee\limits_{t \in T} (\bigwedge\limits_{a_i \in t} a_i = v_i)$ and $R = \bigvee\limits_{t \in T'} (\bigwedge\limits_{a_i \in t} a_i = x_i)$ where T and T' are, respectively, boolean formula terms representing VC and R, find the model M of v_i values to assign to x_i such that $\langle[\![VC]\!]_{\mathbb{F}}\rangle^+ \implies \langle[\![R]\!]_{\mathbb{F}}\rangle^+$ is satisfiable.

Example 2. *Let $VC = (a_1 = v_1) \cdot (a_2 = v_2) \cdot (a_3 = v_3) \cdot (a_4 = v_4) + (a_4 = v_4) \cdot (a_5 = v_5) \cdot (a_6 = v_6)$,*
$R = (a_2 = x_1) \cdot (a_3 = x_2) + (a_4 = x_3) \cdot (a_6 = x_4) + (a_7 = x_5) \cdot (a_8 = x_6)$,
$\mathbb{F} = \{(a_1,v_1),(a_2,v_2),(a_3,v_3),(a_4,v_4),(a_5,v_5),(a_6,v_6),(a_2,x_1),(a_3,x_2),(a_4,x_3),(a_6,x_4),(a_7,x_5),(a_8,x_6)\}$.

Reasoning with PFA

Question : Find σ such that $\sigma(VC) \sqsubseteq \sigma(R)$?
With $\sigma = \{x_1 \mapsto v_2,\ x_2 \mapsto v_3,\ x_3 \mapsto v_4\}$, we have $\sigma(VC) \sqsubseteq \sigma(R)$. However, it's not easy to find this sigma when reasoning with the product family.

Reasoning with SMT

Translation into Boolean formula:

$$\begin{aligned}\langle[\![VC]\!]_{\mathbb{F}}\rangle^+ &= (a_1 = v_1) \wedge (a_2 = v_2) \wedge (a_3 = v_3)\\ &\quad \wedge (a_4 = v_4) \wedge (a_4 = v_4) \wedge (a_5 = v_5)\\ &\quad \wedge (a_6 = v_6)\\ \langle[\![R]\!]_{\mathbb{F}}\rangle^+ &= (a_1 = v_1) \wedge (a_2 = v_2) \wedge (a_3 = v_3)\\ &\quad \wedge (a_4 = v_4) \wedge (a_4 = v_4) \wedge (a_5 = v_5)\\ &\quad \wedge (a_6 = v_6)\end{aligned}$$

Equivalent question in SMT : Find a model M such that $\langle[\![VC]\!]_{\mathbb{F}}\rangle^+ \implies \langle[\![R]\!]_{\mathbb{F}}\rangle^+$ is satisfiable.

The verification with Z3 shows that the formula is satisfiable with the model $M = \{x_1 = v_2,\ x_2 = v_3,\ x_3 = v_4,\ x_4 = v_6,\ x_5 ='',\ x_6 = B\}$ and with that, the user can authenticate.

7 AUTHENTICATION VERIFICATION WITH RISK

7.1 Problem

The question is : *Given a set of user-verifiable credentials, each of which has claimed properties whose disclosure incurs a defined risk, can we identify the verifiable presentation that minimises the user's risk upon disclosure ?* To address this concern, we introduce the Definition 9, which allows us to specify the comparison between two product families to which risk scores have been assigned to the attributes.

Definition 9. *[$\preccurlyeq_r$] Let P, Q be two product family in $\mathcal{P}(\mathbb{P})$. We define an order relation modulo a risk between P and Q, denoted by $P \preccurlyeq_r Q$, as following: $P \preccurlyeq_r Q$ if $r(P) \leq r(Q)$.*

Where $r(P)$ is a given function that computes the risk of the product family P. For the sake of simplicity, we consider the risk of a product family as the total sum of the risks associated with each of its different attributes.

Example 3. *Let $P = a + a \cdot b$ and $r(a) = 5,\ r(b) = 3$: $r(P) = r(a) + r(b) = 8$*

7.2 Case Without Variables

7.2.1 Specification with PFA

Given 2 product family VC and R whose attributes have been assigned risk scores, finding the product that enables the user to access the requested service and represents the minimum risk is the same as finding P such that the following three conditions are satisfied :

- $P \sqsubseteq VC$
- $P \sqsubseteq R$
- $\forall P', P' \sqsubseteq VC \wedge P' \sqsubseteq R, P \preccurlyeq_r P'$

7.2.2 Specification with MAX Weighted SMT

Given 2 product family VC and R whose attributes have been assigned risk scores, finding the product that enables the user to access the requested service and represents the minimum risk is the same as finding a model that satisfies the following objective and constraints.

Objective

$$Min \sum v(a_i) * r(a_i)$$

Where a_i is a VC attribute; $v(a_i)$ is set to 1 if the attribute is chosen and 0 otherwise; $r(a_i)$ represents the risk (here the weight) of the attribute a_i.

Constraints

- SAT(VC)
- $VC \implies R$

Example 4. *Let consider two product family VC and R such that:*

$$\begin{aligned} VC &= a\cdot b\cdot d + a\cdot b + b\cdot d \\ R &= a\cdot b + b\cdot c + b\cdot d \\ r(a) = 5, r(b) &= 3, r(c) = 6, r(d) = 2. \end{aligned}$$

Reasoning with PFA

Question : Find a product P such that the following three conditions are satisfied :

- $P \sqsubseteq VC$
- $P \sqsubseteq R$
- $\forall P', P' \sqsubseteq VC \wedge P' \sqsubseteq R, P \preccurlyeq_r P'$

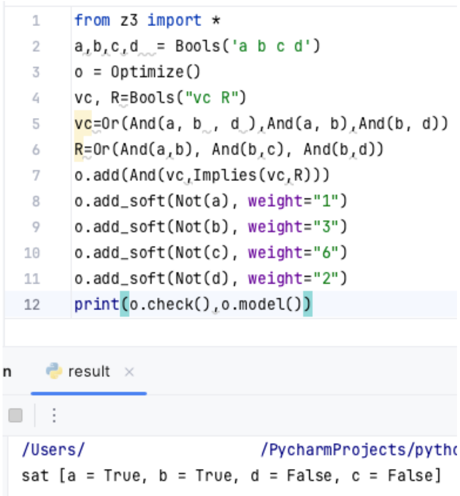
We have :

- $P_1 = a\cdot b,\ r(P_1) = 8$ *since* $a\cdot b \sqsubseteq a\cdot b\cdot d + a\cdot b + b\cdot d$ *and* $a\cdot b \sqsubseteq a\cdot b + b\cdot c + b\cdot d$
- $P_2 = b\cdot d,\ r(P_2) = 5$ *since* $b\cdot d \sqsubseteq a\cdot b\cdot d + a\cdot b + b\cdot d$ *and* $b\cdot d \sqsubseteq a\cdot b + b\cdot c + b\cdot d$
- $P_3 = a\cdot b\cdot d,\ r(P_3) = 10$ *since* $a\cdot b\cdot d \sqsubseteq a\cdot b\cdot d + a\cdot b + b\cdot d$ *and* $a\cdot b\cdot d \sqsubseteq a\cdot b + b\cdot c + b\cdot d$

Therefore, product P is $P = b\cdot d$ *with risk* $r(P) = 5$.

Reasoning with SMT

By programming the specification with Z3 SMT as shown in Fig. 2, we determine that the identity attributes to be provided for authentication are b and d. Thus, the verifiable presentation to be provided to the service provider is $P = b.d$ *with risk* $r(P) = 5$.



```
from z3 import *
a,b,c,d = Bools('a b c d')
o = Optimize()
vc, R=Bools("vc R")
vc=Or(And(a, b , d ),And(a, b),And(b, d))
R=Or(And(a,b), And(b,c), And(b,d))
o.add(And(vc,Implies(vc,R)))
o.add_soft(Not(a), weight="1")
o.add_soft(Not(b), weight="3")
o.add_soft(Not(c), weight="6")
o.add_soft(Not(d), weight="2")
print(o.check(),o.model())
```

```
/Users/                    /PycharmProjects/pytho
sat [a = True, b = True, d = False, c = False]
```

Figure 2: Z3 SMT without variables.

7.3 Case with Variables

7.3.1 Specification with PFA

Given 2 product family VC and R whose attributes have been assigned risk scores, finding the product that enables the user to access the requested service and represents the minimum risk is the same as finding a substitution σ and a product P such that the following three conditions are satisfied :

- $\sigma(P) \sqsubseteq \sigma(VC)$
- $\sigma(P) \sqsubseteq \sigma(R)$
- $\forall P', \sigma(P') \sqsubseteq \sigma(VC) \wedge \sigma(P') \sqsubseteq \sigma(R), \sigma(P) \preccurlyeq_r \sigma(P')$

7.3.2 Specification with MAX Weighted SMT

Given 2 product family VC and R whose attributes have been assigned risk scores, finding the product that enables the user to access the requested service and represents the minimum risk is the same as finding a substitution σ and a model that satisfies the following objective and constraints.

Objective

$$Min \sum (a_i = v_i) * r(a_i = v_i)$$

Where $r(a_i = v_i)$ represent the risk (here the weight) of attribute a_i with value v_i.

Constraints

- SAT($\sigma(VC)$)
- $\sigma(VC) \implies \sigma(R)$

Example 5. *Let consider two product family VC and R such that:*

$$\begin{aligned} VC &= (a_1 = v_1)\cdot(a_2 = v_2)\cdot(a_4 = v_4) \\ &\quad +(a_1 = v_1)\cdot(a_2 = v_2) + (a_2 = v_2)\cdot(a_4 = v_4) \\ Q &= (a_1 = x_1)\cdot(a_2 = x_2) + (a_2 = x_2)\cdot(a_3 = x_3) + \\ &\quad (a_2 = x_2)\cdot(a_4 = x_4) \end{aligned}$$

$r(a_1 = v_1) = 5,\ r(a_2 = v_2) = 3,\ r(a_3 = v_3) = 6,\ r(a_4 = v_4) = 2.$

Reasoning with PFA

Question : Find a substitution σ *based on which we can obtain a product P such that the following three conditions are satisfied:*

- $\sigma(P) \sqsubseteq \sigma(VC)$
- $\sigma(P) \sqsubseteq \sigma(R)$
- $\forall P', \sigma(P') \sqsubseteq \sigma(VC) \wedge \sigma(P') \sqsubseteq \sigma(R),\ \sigma(P) \preccurlyeq_r \sigma(P')$.

We have :

- $\sigma_1 = \{x_1 \mapsto v_1,\ x_2 \mapsto v_2,\ x_3 \mapsto \text{""""},\ x_4 \mapsto \text{""""}\},\ P_1 = (a_1 = v_1)\cdot(a_2 = v_2),\ r(P_1) = 8$ *since* $(a_1 = v_1)\cdot(a_2 = v_2) \sqsubseteq (a_1 = v_1)\cdot(a_2 = v_2)\cdot(a_4 = v_4) +$

$(a_1 = v_1) \cdot (a_2 = v_2) + (a_2 = v_2) \cdot (a_4 = v_4)$ *and* $(a_1 = v_1) \cdot (a_2 = v_2) \sqsubseteq (a_1 = v_1) \cdot (a_2 = v_2) + (a_2 = v_2) + (a_2 = v_2)$

- $\sigma_2 = \{x_1 \mapsto \text{""},\ x_2 \mapsto v_2,\ x_3 \mapsto \text{""},\ x_4 \mapsto v_4\}$, $P_2 = (a_2 = v_2) \cdot (a_4 = v_4)$, $r(P_2) = 5$ *since* $(a_2 = v_2) \cdot (a_4 = v_4) \sqsubseteq (a_1 = v_1) \cdot (a_2 = v_2) \cdot (a_4 = v_4) + (a_1 = v_1) \cdot (a_2 = v_2) + (a_2 = v_2) \cdot (a_4 = v_4)$ *and* $(a_2 = v_2) \cdot (a_4 = v_4) \sqsubseteq (a_2 = v_2) + (a_2 = v_2) + (a_2 = v_2) \cdot (a_4 = v_4)$
- $\sigma_3 = \{x_1 \mapsto v_1,\ x_2 \mapsto v_2,\ x_3 \mapsto \text{""},\ x_4 \mapsto v_4\}$, $P_3 = (a_1 = v_1) \cdot (a_2 = v_2) \cdot (a_4 = v_4)$, $r(P_4) = 10$ *since* $(a_1 = v_1) \cdot (a_2 = v_2) \cdot (a_4 = v_4) \sqsubseteq (a_1 = v_1) \cdot (a_2 = v_2) \cdot (a_4 = v_4) + (a_1 = v_1) \cdot (a_2 = v_2) + (a_2 = v_2) \cdot (a_4 = v_4)$ *and* $(a_1 = v_1) \cdot (a_2 = v_2) \cdot (a_4 = v_4) \sqsubseteq (a_1 = v_1) \cdot (a_2 = v_2) + (a_2 = v_2) + (a_2 = v_2) \cdot (a_4 = v_4)$.

Therefore, $\sigma = \{x_1 \mapsto \text{""}, x_2 \mapsto v_2, x_3 \mapsto \text{""}, x_4 \mapsto v_4,\}$ *and the product P is* $P = (a_2 = v_2) \cdot (a_4 = v_4)$ *With risk of* $r(P) = 5$.

Reasoning with SMT

By programming the specification with Z3 SMT as shown in Fig. 3, we determine that the model with the minimum risk is the one with $\sigma = \{x_1 \mapsto \text{""},\ x_2 \mapsto v_2,\ x_3 \mapsto \text{""},\ x_4 \mapsto v_4\}$. *Therefore, the verifiable presentation to be provided to the service provider is* $P = (a_2 = v_2) \cdot (a_4 = v_4)$ *With risk* $r(P) = 5$.

8 RELATED WORK

Despite the proliferation of decentralized identity systems, there is a dearth of research on privacy preservation in user-centric identity management through formal approaches that consider risks directly associated with identity attributes. Some studies have addressed user-centric privacy by focusing on context-related risks.

Jafari et *al.* (Jafari-Lafti et al., 2009) introduced P2F, a recommendation tool that analyzes a user's transaction history and privacy preferences, along with real-world privacy guidelines, to prevent unintended disclosure of personal information. This tool uses a risk assessment model based on service providers' properties, potential collusion between providers, sensitivity of disclosed information, and the risk of linking to undesirable transactions. Similarly, (Ahn and Sekar, 2011) proposed a risk-based approach to help users evaluate the risk level of disclosing their identity in user-centric identity management, employing ontology-based evaluation and privacy preference assessment. However, these approaches focus on contextual elements without directly considering the value of identity attributes.

Zaeem et *al.* (Chang et al., 2018) proposed two methods, using the Identity Threat Assessment and Prediction (ITAP) database, to aid in evaluating and identifying the optimal set of Personally Identifiable Information (PII) that meets authentication goals while minimizing risk exposure. The static approach assigns risk and uniqueness scores to identity attributes, combining these to determine a final score. The dynamic approach uses Bayesian networks to infer the risk score of an attribute, considering its accessibility and the potential impact of inappropriate disclosure. Zaeem et *al.* (Zaeem et al., 2016) work provides a useful foundation for assigning risk scores to identity attributes.

9 DISCUSSION

A notable contribution of this paper is the translation of product family algebra into Boolean logic, enabling the automatic verification of constraints associated with product families using SAT or SMT solvers. This advancement facilitates the rapid and reliable detection of errors or conflicts in product configurations.

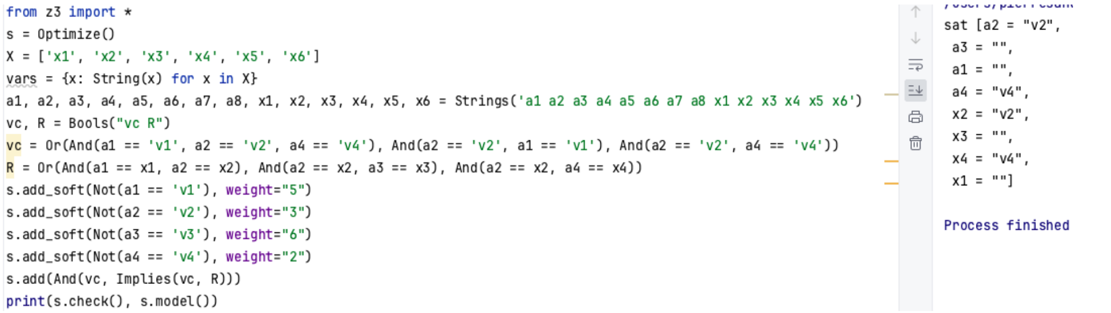
In the authentication protocol, the service provider's request sets the context and defines specific constraints for user authentication. By interpreting verifiable credentials and the service provider's request as product families and translating them into Boolean formulas, we simplify the verification process. Our research demonstrates that determining a user's ability to authenticate becomes straightforward when framed as a satisfiability problem.

Our findings also indicate that integrating an SMT solver into a user's digital wallet could lead to a new generation of intelligent wallets. These wallets would not only secure assets but also assist users in making informed decisions, personalizing their experience, and navigating the decentralized identity ecosystem. They could transparently disclose identity attributes with minimal harm, provided the attributes are correctly encoded and risks properly characterized.

Verifiable credentials are inherently more complex than our current research scope, involving metadata, claims, and proof. Even if a user possesses all the attributes required by a service provider, authentication may fail if the service provider cannot verify the identity's origin.

10 CONCLUSION

The increasing use of verifiable credentials brings us closer to the conventional practice of physical identity, allowing users to select among multiple identities for service access, provided the service provider accepts the chosen identity. Our research demonstrates



```
from z3 import *
s = Optimize()
X = ['x1', 'x2', 'x3', 'x4', 'x5', 'x6']
vars = {x: String(x) for x in X}
a1, a2, a3, a4, a5, a6, a7, a8, x1, x2, x3, x4, x5, x6 = Strings('a1 a2 a3 a4 a5 a6 a7 a8 x1 x2 x3 x4 x5 x6')
vc, R = Bools("vc R")
vc = Or(And(a1 == 'v1', a2 == 'v2', a4 == 'v4'), And(a2 == 'v2', a1 == 'v1'), And(a2 == 'v2', a4 == 'v4'))
R = Or(And(a1 == x1, a2 == x2), And(a2 == x2, a3 == x3), And(a2 == x2, a4 == x4))
s.add_soft(Not(a1 == 'v1'), weight="5")
s.add_soft(Not(a2 == 'v2'), weight="3")
s.add_soft(Not(a3 == 'v3'), weight="6")
s.add_soft(Not(a4 == 'v4'), weight="2")
s.add(And(vc, Implies(vc, R)))
print(s.check(), s.model())
```

```
sat [a2 = "v2",
 a3 = "",
 a1 = "",
 a4 = "v4",
 x2 = "v2",
 x3 = "",
 x4 = "v4",
 x1 = ""]

Process finished
```

Figure 3: Z3 specification with variables and result (on the right).

that formal methods can enhance information security by leveraging product family algebra and Satisfiability Modulo Theories (SMT) to verify user access. Additionally, we address the risk associated with disclosing specific identity attributes, thereby enhancing privacy protection and adhering to the need-to-know principle. We show that Max weighted SMT automates and resolves the disclosure of identity attributes with minimal risk, facilitated by the Z3 solver.

This approach can be integrated into a decentralized identity management system, which will be the next step in our research. We also plan to refine this approach by considering the creation of verifiable presentations from verifiable credentials, including attribute aggregation where possible.

REFERENCES

Ahn, G.-J. and Sekar, P. (2011). Ontology-based risk evaluation in user-centric identity management. In *2011 IEEE International Conference on Communications (ICC)*, pages 1–5. IEEE.

Chang, K. C., Zaeem, R. N., and Barber, K. S. (2018). Enhancing and evaluating identity privacy and authentication strength by utilizing the identity ecosystem. In *Proceedings of the 2018 Workshop on Privacy in the Electronic Society*, pages 114–120.

De Moura, L. and Bjørner, N. (2011). Satisfiability modulo theories: introduction and applications. volume 54, pages 69–77. ACM New York, NY, USA.

de Moura, L., Dutertre, B., and Shankar, N. (2007). A tutorial on satisfiability modulo theories: (invited tutorial). In *International conference on computer aided verification*, pages 20–36. Springer.

Höfner, P., Khedri, R., and Möller, B. (2011). An algebra of product families. *Software & Systems Modeling*, 10:161–182.

Jafari-Lafti, M., Huang, C.-T., and Farkas, C. (2009). P2f: A user-centric privacy protection framework. In *2009 International Conference on Availability, Reliability and Security*, pages 386–391. IEEE.

Li, Y., Fu, Y., Du, Z., and Cai, Z. (2022). An access control scheme based on decentralized identifiers and verifiable credentials in iot. In *2022 3rd International Conference on Computer Science and Management Technology (ICCSMT)*, pages 279–283. IEEE.

Lim, S., Rhie, M.-H., Hwang, D., and Kim, K.-H. (2021). A subject-centric credential management method based on the verifiable credentials. In *2021 International Conference on Information Networking (ICOIN)*, pages 508–510. IEEE.

Manu, S., Dave, L., David, C., and Orie, S. (2024). Verifiable credentials data model v2.0. *Draft Community Group Report*.

Manu, S., Dave, L., Markus, S., Drummond, R., Orie, S., and Christopher, A. (2022). Decentralized identifiers (dids) v1.0 : Core architecture, data model, and representations.

Xu, L., Li, T., and Erkin, Z. (2023). Verifiable credentials with privacy-preserving tamper-evident revocation mechanism. In *2023 Fifth International Conference on Blockchain Computing and Applications (BCCA)*, pages 266–273. IEEE.

Zaeem, R. N., Budalakoti, S., Barber, K. S., Rasheed, M., and Bajaj, C. (2016). Predicting and explaining identity risk, exposure and cost using the ecosystem of identity attributes. In *2016 IEEE International Carnahan Conference on Security Technology (ICCST)*, pages 1–8. IEEE.