Supporting User Privacy Preferences on Information Release in Open Scenarios

Claudio A. Ardagna[1] Sabrina De Capitani di Vimercati[1]
Sara Foresti[1] Stefano Paraboschi[2] Pierangela Samarati[1]

(1) DTI - Università degli Studi di Milano

(2) DIIMM - Università degli Studi di Bergamo

W3C Workshop on Privacy and Data Usage Control
October 5, 2010 – Cambridge, MA, USA

Starting scenario (1)

- Open scenarios where clients interact with remote parties and access remote resources
- Depart from the assumption that clients are authenticated before evaluating access requests
- The policy at the server refers to credentials/properties that the client must have (in contrast to client's identity)

$\Longrightarrow$ Attribute-based/credential-based access control

Starting scenario (2)

- Attribute-based access control requires re-thinking how access control process works
- Most proposals focus on the server side aspect of the problem
  - regulate how the server specifies policies
  - provide partial evaluation of the policy
  - define how to communicate policies to the client
  - they assume to adopt a symmetric approach at the client

Motivation

Access-control based specifications do not fit well the problem at the client side

+ they allow users to specify whether some information can be or cannot be released
− they do not allow users to express the fact that they might prefer to release some information over other when given choices

$\Longrightarrow$ Need to provide users with means to effectively regulate the release of their information

Goal of our work

Enable users to effectively regulate disclosure of their properties and credentials

- identify requirements and concepts that need to be captured
- organize of users properties and credentials in the user portfolio
- enable users to specify how much she values the disclosure of different components of the portfolio
- provide possible technical approaches for supporting user's preferences
- provide a basis for investigating user-friendly/user-understandable approaches for regulating release of user's properties

Client portfolio modeling

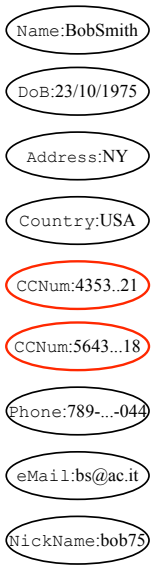
- The information of the client forms a client portfolio
- Credential: certificate issued and signed by a third party
  - certifies a set of properties
  - has a type, an identifier, and an issuer
- Declaration: property stored as a self-signed credential
- Hierarchy of abstractions of credential types $\mathscr{H}(\mathscr{T}, \preceq_{isa})$ (e.g., *id_card* $\preceq_{isa}$ *id*, *id* $\preceq_{isa}$ *credential*)

Client portfolio – Properties

- Credential-independent: the value depends only on the credential's owner (e.g., birth date)

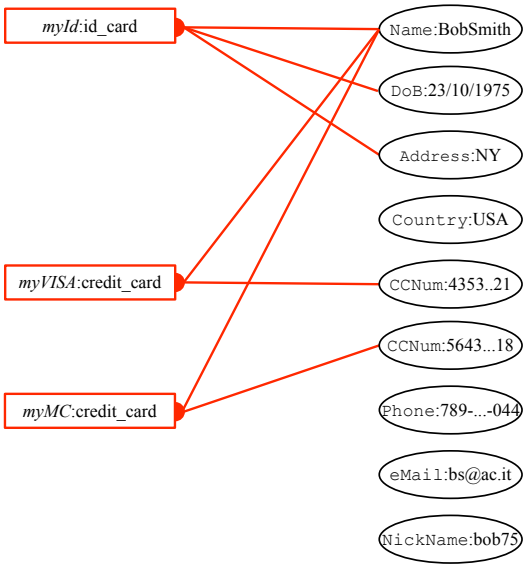
Client portfolio – Properties

- Credential-independent: the value depends only on the credential's owner (e.g., birth date)
- Credential-dependent: the value depends on the certifying credential (e.g., credit card number)



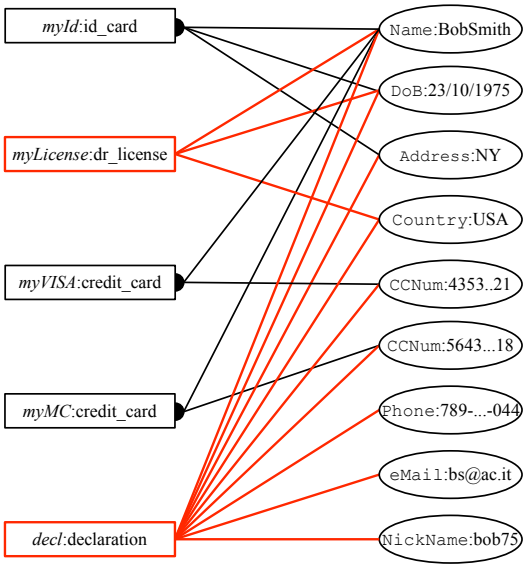
Client portfolio – Credentials

- Atomic: released as a whole (e.g., X.509)



Client portfolio – Credentials

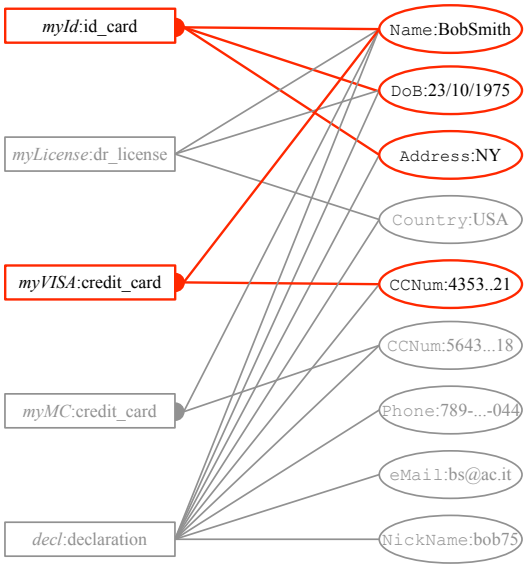
- Atomic: released as a whole (e.g., X.509)
- Non-atomic: properties can be selectively released, proof-of-possession can be certified (e.g., Idemix, U-Prove)



Disclosure

A disclosure is a subset of the client portfolio that satisfies:

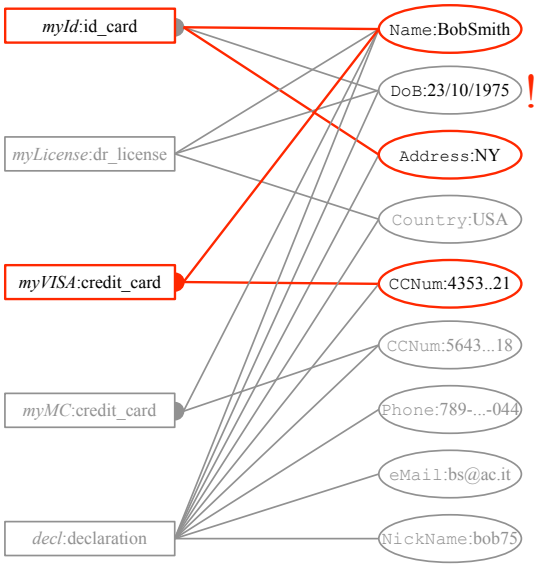
- certifiability: each property is certified by a credential
- atomicity: if a property of an atomic credential is disclosed, all its properties are disclosed



Disclosure

A disclosure is a subset of the client portfolio that satisfies:

- certifiability: each property is certified by a credential
- atomicity: if a property of an atomic credential is disclosed, all its properties are disclosed



Does not satisfy atomicity!

Privacy preferences – Requirements

- Clients may prefer to disclose some properties/credentials over others $\Longrightarrow$ different portfolio elements have different sensitivity
- Privacy preference specifications are needed to:
  - automatically regulate the disclosure of sensitive information
  - minimize the disclosure of sensitive information
- A solution to express privacy preferences must support:
  - fine-grained control on sensitive information
  - specifications on the sensitivity of associations
  - constraints on the disclosure of information

Portfolio sensitivity

- Privacy preferences expressed as sensitivity labels
- Sensitivity labels reflect how much a client values the disclosure of credentials/properties in the portfolio
- Sensitivity labels are characterized by:
  - partial order relationship $\succeq$
  - composition operator $\oplus$ for computing sensitivity of a set of elements, can be based on
    - additivity: the sensitivity of a combined disclosure is the sum of the sensitivities of the disclosed elements
    - maximum: the sensitivity of a combined disclosure is the upper bound of the sensitivities of the sensitivities of the disclosed elements

Sensitivity labels – Examples

- Sensitivity labels as integer values
  - $\succeq$ is the $\geq$ total order relationship
  - $\oplus$ is the sum $+$ of values (additivity)

  (e.g., $\lambda(\texttt{Name})$=1, $\lambda(\texttt{DoB})$=5, $\lambda(\texttt{Name})\oplus\lambda(\texttt{DoB})$=6)

- Sensitivity labels as multilevel security classifications
  - $\succeq$ is the total order relationship on security classes
  - $\oplus$ is the least upper bound (maximum)

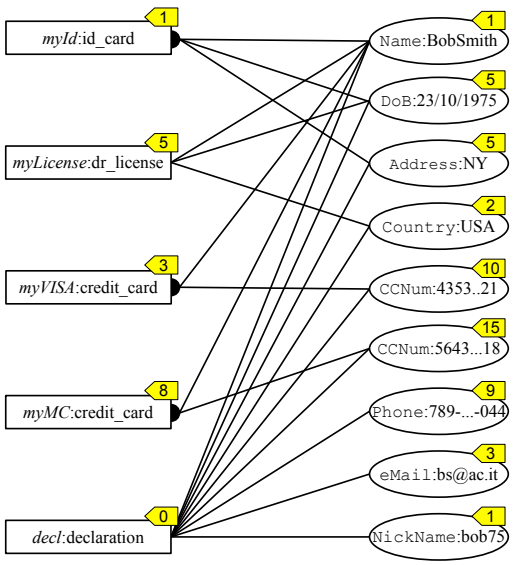
  (e.g., $\lambda(\texttt{Name})$=unclassified, $\lambda(\texttt{DoB})$=secret, $\lambda(\texttt{Name})\oplus\lambda(\texttt{DoB})$=secret)

For this talk we assume sensitivity labels as integer values

Sensitivity of properties and credentials

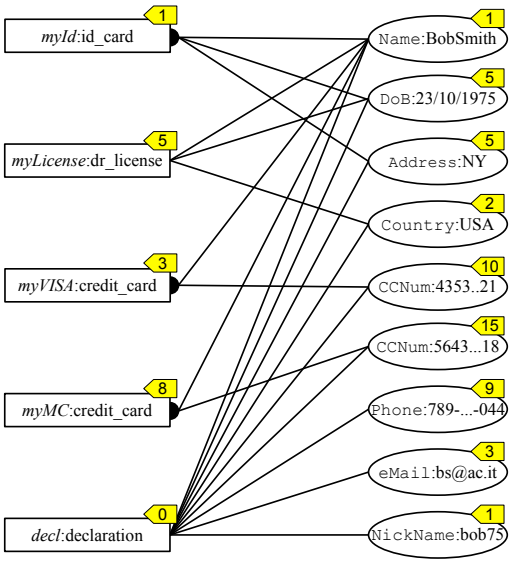
Specify how a client values information in her portfolio

- $\lambda(p)$: sensitivity of property p individually taken
- $\lambda(c)$: sensitivity of the existence of credential c



Sensitivity of associations

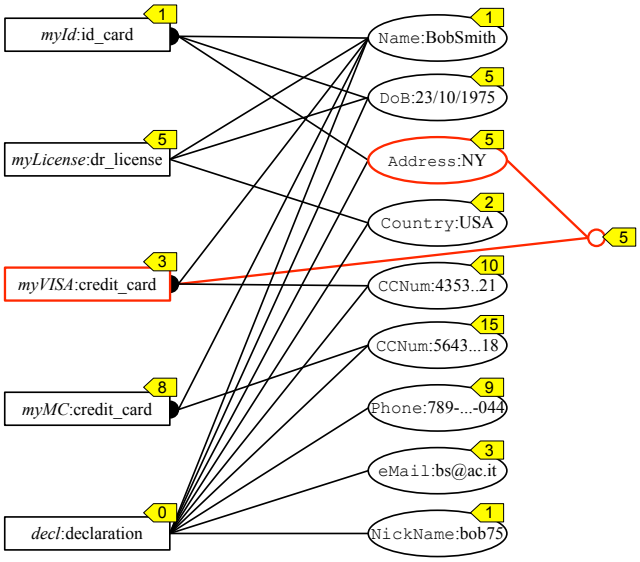
$\lambda(A)$: sensitivity of an association $A=\{p_i,\ldots,p_j,c_k,\ldots,c_n\}$, whose joint release carries:



Sensitivity of associations

$\lambda(A)$: sensitivity of an association $A=\{p_i,\ldots,p_j,c_k,\ldots,c_n\}$, whose joint release carries:

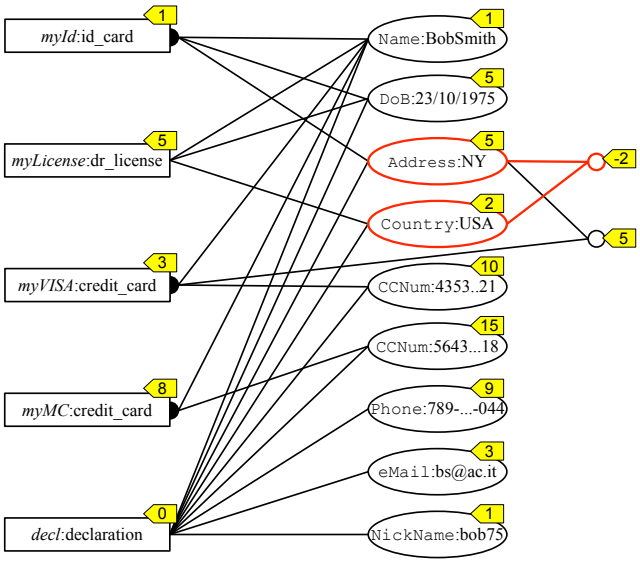
- more information than the release of each element in A
  $\Longrightarrow$ sensitive view



Sensitivity of associations

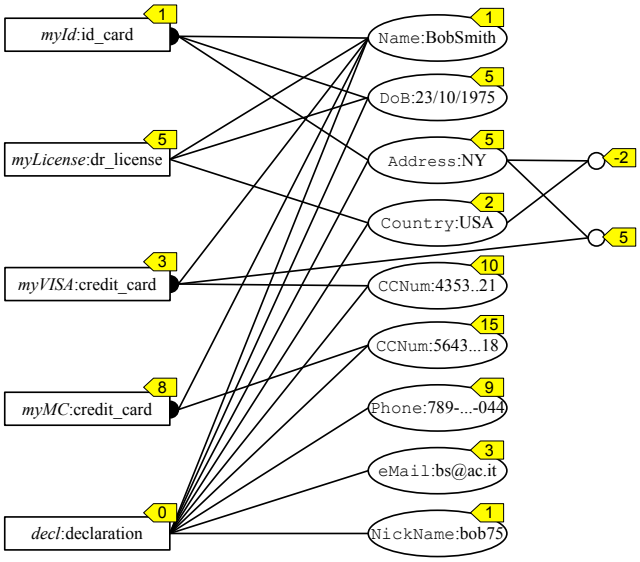
$\lambda(A)$: sensitivity of an association $A=\{p_i,\ldots,p_j,c_k,\ldots,c_n\}$, whose joint release carries:

- more information than the release of each element in A
  $\Longrightarrow$ sensitive view
- less information than the release of each element in A
  $\Longrightarrow$ dependency



Disclosure constraints

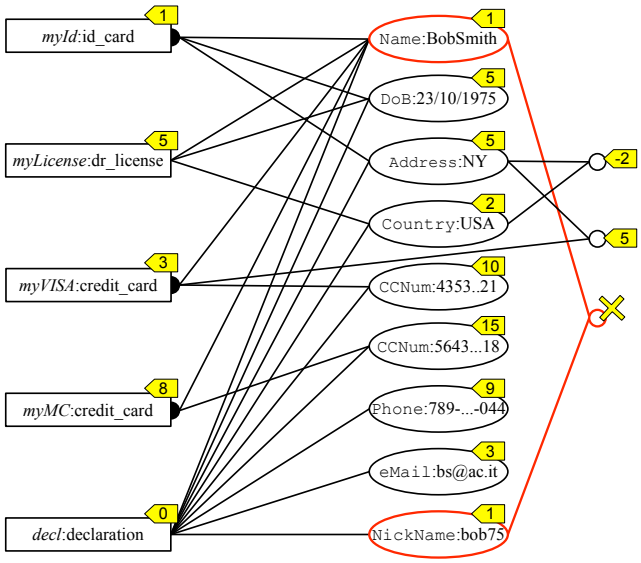
Set $A=\{p_i,\ldots,p_j,c_k,\ldots,c_n\}$ of elements whose release must be controlled



Disclosure constraints

Set $A=\{p_i,\ldots,p_j,c_k,\ldots,c_n\}$ of elements whose release must be controlled

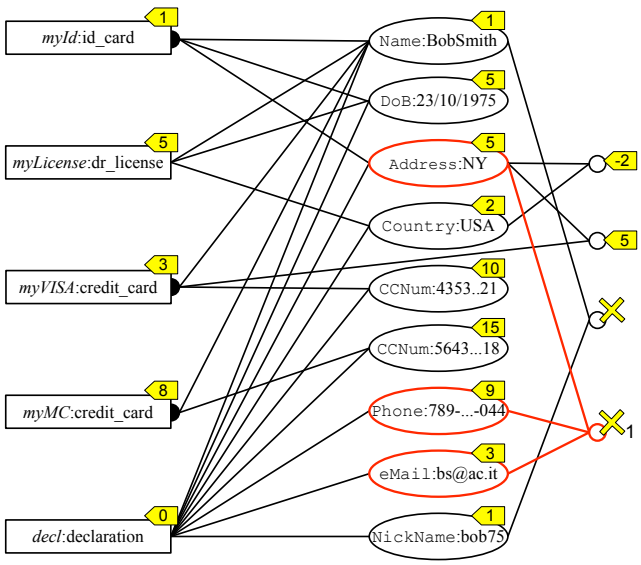
- forbidden view: the release of A is prohibited



Disclosure constraints

Set $A=\{p_i,\ldots,p_j,c_k,\ldots,c_n\}$ of elements whose release must be controlled

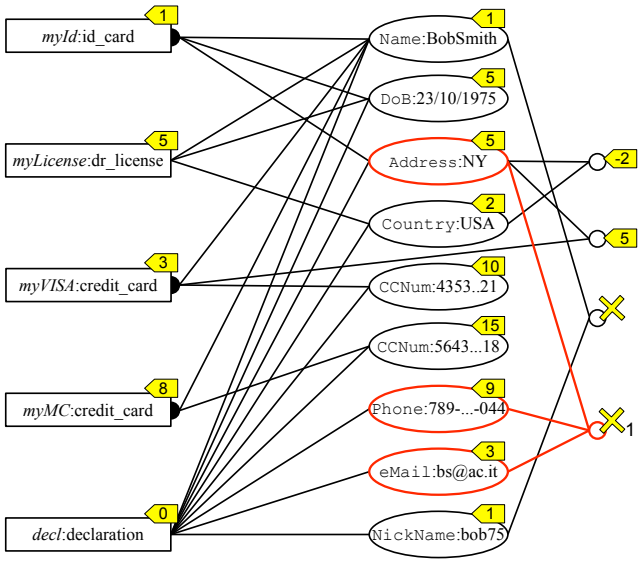
- forbidden view: the release of A is prohibited
- disclosure limitation: at most n elements in A can be released



Disclosure constraints

Set $A=\{p_i,\ldots,p_j,c_k,\ldots,c_n\}$ of elements whose release must be controlled

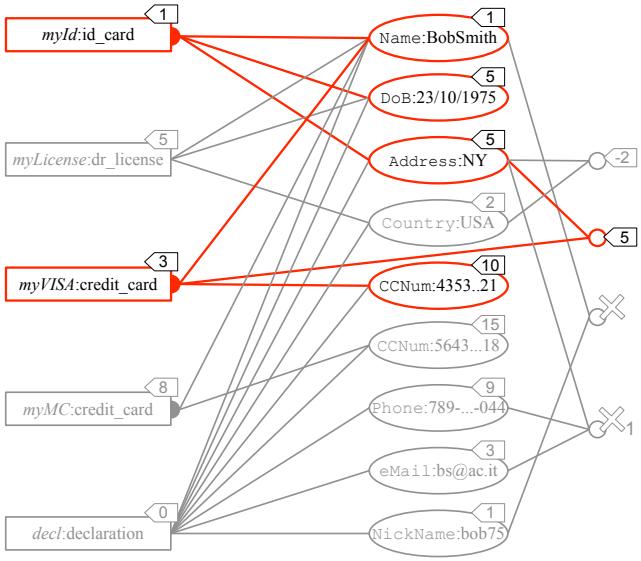
- forbidden view: the release of A is prohibited
- disclosure limitation: at most n elements in A can be released



A disclosure is valid if no disclosure constraint is violated

Disclosure sensitivity

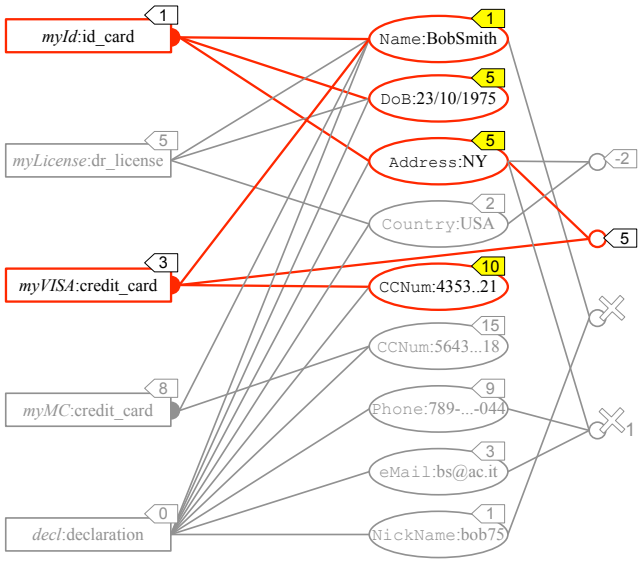
The sensitivity $\lambda(\mathscr{D})$ of a disclosure $\mathscr{D}$ is the sum of the sensitivity labels of released:



Disclosure sensitivity

The sensitivity $\lambda(\mathscr{D})$ of a disclosure $\mathscr{D}$ is the sum of the sensitivity labels of released:

- properties

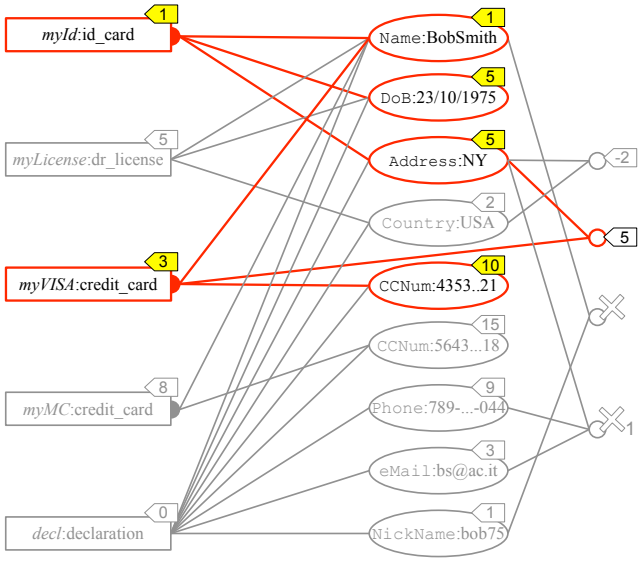


$\lambda(\mathscr{D})$ = 1+5+5+10

Disclosure sensitivity

The sensitivity $\lambda(\mathscr{D})$ of a disclosure $\mathscr{D}$ is the sum of the sensitivity labels of released:

- properties
- credentials

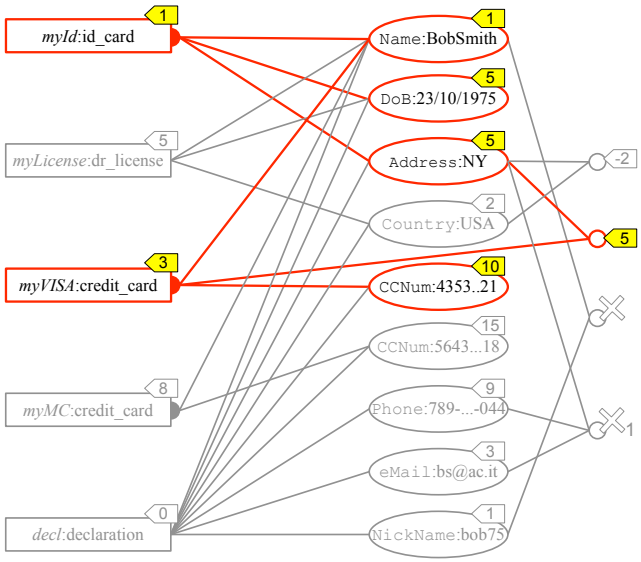


$\lambda(\mathscr{D})$ = 1+5+5+10+1+3

Disclosure sensitivity

The sensitivity $\lambda(\mathscr{D})$ of a disclosure $\mathscr{D}$ is the sum of the sensitivity labels of released:

- properties
- credentials
- associations



$\lambda(\mathscr{D})$ = 1+5+5+10+1+3+5 = 30

Server request

Request $\mathscr{R}$: disjunction of simple requests

- Simple request R: conjunction of terms
  - term r=*type*.$\{p_1,\ldots,p_m\}$: disclosure of $\{p_1,\ldots,p_m\}$ from c s.t. *type*$(c) \preceq_{isa}$ *type*

    $\Longrightarrow$ *type* is an abstraction of credential type *type*(c) in $\mathscr{H}$

Example

$\mathscr{R} = r_1 \wedge r_2$

r_1 = *id*.{Name,Address}

r_2 = *cc*.{Name,CCNum}

Min-disclosure problem

A disclosure $\mathscr{D}$:

- satisfies $\mathscr{R}$ if it satisfies at least a R in $\mathscr{R}$
- satisfies R if, $\forall$ r=*type*.$\{p_1,\ldots,p_m\}$ in R, it includes c s.t.:
  - c certifies $\{p_1,\ldots,p_m\}$
  - *type*$(c)\preceq_{isa}$*type*

Min-disclosure problem

$\mathscr{R}$ = *id*.{Name,Address} $\wedge$ *cc*.{Name,CCNum}

A disclosure $\mathscr{D}$:

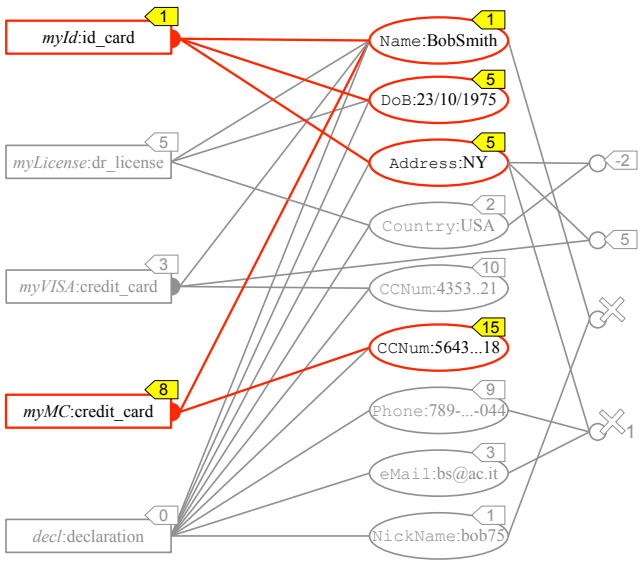
- satisfies $\mathscr{R}$ if it satisfies at least a R in $\mathscr{R}$
- satisfies R if, $\forall$ r=*type*.$\{p_1,\ldots,p_m\}$ in R, it includes c s.t.:
  - c certifies $\{p_1,\ldots,p_m\}$
  - *type*$(c)\preceq_{isa}$*type*

Min-disclosure problem

$\mathscr{R}$ = *id*.{Name,Address} ∧ *cc*.{Name,CCNum}

A disclosure $\mathscr{D}$:

- satisfies $\mathscr{R}$ if it satisfies at least a R in $\mathscr{R}$
- satisfies R if, $\forall$ r=*type*.$\{p_1,\ldots,p_m\}$ in R, it includes c s.t.:
  - c certifies $\{p_1,\ldots,p_m\}$
  - *type*(c)$\preceq_{isa}$*type*



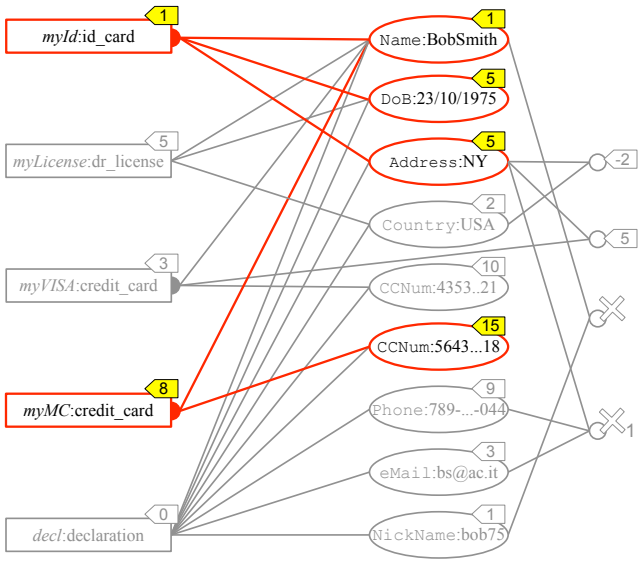
$\lambda(\mathscr{D})$ = 1+8+1+5+5+15 = 35

Min-disclosure problem

$\mathscr{R}$ = *id*.{Name,Address} ∧ *cc*.{Name,CCNum}

A disclosure $\mathscr{D}$:

- satisfies $\mathscr{R}$ if it satisfies at least a R in $\mathscr{R}$
- satisfies R if, $\forall$ r=*type*.$\{p_1,\ldots,p_m\}$ in R, it includes c s.t.:
  - c certifies $\{p_1,\ldots,p_m\}$
  - *type*(c)$\preceq_{isa}$*type*
- is minimum if $\nexists$ a valid disclosure $\mathscr{D}'$ s.t. $\mathscr{D}'$ satisfies $\mathscr{R}$ and $\lambda(\mathscr{D}')<\lambda(\mathscr{D})$



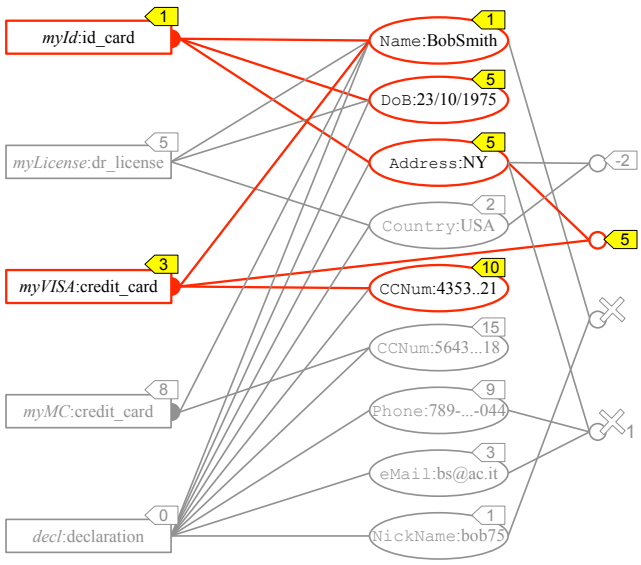
$\lambda(\mathscr{D})$ = 35 $\Longrightarrow$ $\mathscr{D}$ is not minimum

Min-disclosure problem

$\mathscr{R}$ = *id*.{Name,Address} $\wedge$ *cc*.{Name,CCNum}

A disclosure $\mathscr{D}$:

- satisfies $\mathscr{R}$ if it satisfies at least a R in $\mathscr{R}$
- satisfies R if, $\forall$ r=*type*.$\{p_1,\ldots,p_m\}$ in R, it includes c s.t.:
  - c certifies $\{p_1,\ldots,p_m\}$
  - *type*(c)$\preceq_{isa}$*type*
- is minimum if $\nexists$ a valid disclosure $\mathscr{D}'$ s.t. $\mathscr{D}'$ satisfies $\mathscr{R}$ and $\lambda(\mathscr{D}')<\lambda(\mathscr{D})$



$\lambda(\mathscr{D}')$ = 30 $\Longrightarrow$ $\mathscr{D}'$ is minimum

Computing a minimal disclosure

The problem of computing a disclosure that minimizes release of information is NP-hard

- exploit graph-based representation of portfolio and requests, providing heuristics based on graph-matching [PASSAT'10]
- exploit Max-SAT representation of the problem and existing SAT solver [WPES'10]

Work to be investigated

- Sensitivity labels assigned to proofs (provided by non-atomic credentials)
- Sensitivity labels based on context
- Integration with server-side solutions and more expressive server requests
- User-intuitive approaches for expressing preferences (and possibly translate them to sensitivity labels)
- Consideration of previous disclosures