Detecting Inference Channels in Private Multimedia Data *via* Social Networks

Béchara Al Bouna[1] and Richard Chbeir[1]

[1] LE2I Laboratory UMR-CNRS, University of Bourgogne
21078 Dijon Cedex France
{bechara.albouna, richard.chbeir}@u-bourgogne.fr

Abstract. Indirect access to protected information has been one of the key challenges facing the international community for the last decade. Providing techniques to control direct access to sensitive information remain insufficient against inference channels established when legitimate data reveal classified facts hidden from unauthorized users. Several techniques have been proposed in the literature to meet indirect access prevention. However, those addressing the inference problem when involving multimedia objects (images, audio, video, etc.) remain few and hold several drawbacks. In essence, the complex structure of multimedia objects makes the fact of detecting indirect access a difficult task. In this paper, we propose a novel approach to detect possible inference channels established between multimedia objects representing persons by combining social network information with unmasked content of multimedia objects. Here, we present the techniques used to map the content of social networks to the set of multimedia objects at hand. We also provide an `MiD` function able to determine whether an unmasked multimedia object combined with data from the social network infers a sensitive multimedia object.

Keywords: Inference Channels, Multimedia, Access Control

1 Introduction

Providing appropriate techniques to protect sensitive information to be published and shared requires both 1) defining direct access and who has the right to perform a specified operation on confidential resources, and 2) preventing indirect access to information occurring when legitimate data reveal classified facts hidden from unauthorized users. On one hand, several access control models [9] [15] [18] have been proposed in the literature to meet direct access prevention requirements. Recently, with the increased use of multimedia objects (images, audio, video, etc.), the tradeoff between data availability and privacy has lead to the definition of adapted models [1] [4] [5] [6] [7] providing safe browsing and publishing of multimedia objects' contents. Particularly, masking out objects of interests representing persons is of great importance in several privacy scenarios (e.g. hiding the face of a popular person in a TV show). On the other hand, several studies [10] [12] [20] [27] [29] have focused on handling various forms of indirect access commonly known as the inference problem. They focus mainly on preventing inference channels in textual-based applications. However, none to our knowledge has explored the damage that might be caused by inference channels established in multimedia-based environments due to social networks or other common knowledge. In fact, social networks are becoming popular[1] and attracting lots of people and organizations who publish data, pictures, and share visions, ideas, hobbies, friendship, kinship, dislike, etc. Almost every user has an account on a social network with information containing pictures of him and a set of relations established with others (*friendOf, CollegeOf, inRelationshipWith*, etc.). In many situations, such information or common knowledge combined with unmasked content of multimedia objects make high the potential risk of uncovering the sensitive content of multimedia objects.

In this paper, we address privacy protection in multimedia objects representing persons by detecting inference channels thanks to knowledge gathered from social networks. Our study aims to detect whether a masked content of multimedia objects is endangered due to combining the social networks knowledge with unmasked (salient) objects. Here, we propose a two-phase approach to elaborate, on one hand, the social networks knowledge representation and multimedia objects mappings, and to provide, on the other hand, algorithms to detect possible inference channels established when protecting one or several sensitive multimedia objects. To the best of our knowledge, this is the first study to address multimedia-based inference problem using social networks.

The rest of this paper is organized as follows. In Section 2, we describe a motivation scenario to show the risks rose from social networks when protecting multimedia objects' content. In Section 3, we point out the set of techniques proposed in the literature to tackle inference channels. In Section 4, we present a set of definitions needed to fully understand our approach. In Section 5, we present our proposal holding a mapping module to map social network nodes and edges to multimedia objects. Finally, we conclude our paper and present some future directions.

2 Motivating Scenario

Let us consider a company holding a local image database accessible to all the staff members, visitors, trainees, clients, and collaborators. The images are categorized as follows:

- *Social dinner events*: containing all the photos of the staff members taken during social dinners with their husbands (or wives) and relatives.
- *Meetings*: containing all the images taken during meetings of staff members held in the research department.

To manage these images, a package containing a set of functions (distortion, face detection, movement detection, etc.) is provided with a search engine allowing to retrieve images using query-by-example techniques based-on low-level features

[1] For example, the Facebook social network holds more than one hundred million users

(colors, texture, shapes, etc.), similar object (i.e. sample image), or meta-data (keywords). A simple publication policy is defined in the company to preserve privacy ethics when publishing the content of the database; it states that *all staff members who are part of the head office of the research department should not be appear in social events' photos*. In order to apply this policy, the webmaster, after identifying (automatically and/or manually) the related images, uses a blur filter function to hide the related multimedia objects content. Fig. 1 shows both a photo of `Mr. Dupond`, head of the research department, with his wife and colleagues, taken in one of the social dinners, and the same photo after applying the publication policy where his face was blurred. `Mr. Dupond` is also active on the web and has an account on a known social network where he posts and shares several information with his friends, family and wife. In this situation, one can see that hiding the face of `Mr. Dupond` won't be enough here. That is, people who have access to information at the social network and aware of the identity of his wife, might easily recognise him *via* the presence of his wife sited nearby in the image to be secured.

Fig. 1. Social dinner photo with the head of the research department, `Mr. Dupond`, before and after applying the publication policy

This type of inference problem, that we call *inference by domain knowledge*, involving multimedia objects remains critical. In essence, combining unmasked information with the knowledge of the application domain might reveal interesting information which puts privacy at risks. However, as mentioned before, none has considered its influence when protecting sensitive content of multimedia objects. Our work here is dedicated to suggest a challenging solution.

3 Related Work

In this section, we present an overview of the studies conducted in the literature to address inference detection and elimination techniques in three different areas: database, XML, and multimedia environments.

The inference problem in databases occurs when sensitive information can be disclosed from non sensitive data combined with either metadata/database constraints or external data related to the domain knowledge. Such an issue has been widely discussed in the database environment where users are able to establish inference channels based on the knowledge extracted from the domain. In [20], the authors use classical information theory to calculate the bandwidth of illegal information flow using an INFER function. An inference channel is established if there exists an item in the *sphere of influence* which is composed of entities, attributes, relationships and constraints with a classification higher than the classification of specified information. Research led by Hinke and Delugach in [12] led to the definition of Wizard [13], a system that takes a database schema as input and tests if it is possible to establish inference channels according to a set of predefined semantic graphs related to the domain knowledge. In this approach, domain knowledge data are acquired in a microanalysis to enrich database semantics. In [10], the authors present a Semantic Inference Model (SIM) based on the data contained in a given database, the schema of the database, and the semantic relations that might exist between the data. The authors use Bayesian Networks in order to calculate the possibility of inferring sensitive information.

Several studies have emerged to tackle indirect access caused by inference channels in XML environments. The work led by Yang and Li in [28] proposes an interesting approach in which it is possible to detect inference channels established when combining common knowledge with unclassified information along with others related to functional dependencies between different nodes in an XML document. Their approach is based on *conditions* $\rightarrow$ *facts* in order to represent XML constraints. The authors use an algorithm to construct an AND/OR graph which helps removing unnecessary links between unclassified information and the sensitive ones.

The described approaches are interesting and provide satisfactory results (depending on the application domain) when handling textual data. However, they cope badly with multimedia data. In fact, detecting inference channels in multimedia environments still complex for two main reasons: 1) the semantic gap between the low level features and the semantic meaning of a multimedia object, and 2) the complex structure of multimedia objects. Few studies have addressed so far the effect of inference when controlling multimedia objects. In [14], the authors define an interesting approach to replace salient objects of a video with virtual objects. Although, the approach looks efficient in preserving privacy and eliminating statistical inference, it puts however at risk data semantics where it becomes difficult to recognize some crucial content of the video.

4 Preliminaries and Definitions

In this section, we define the main concepts on which our approach relies. We first describe the basic concepts of multimedia objects, sensitive multimedia objects, multimedia relations, similarity functions and domain relations. After, we give the formal representation of social networks and show how it is possible to enrich social networks with new knowledge using explicit and user-defined rules.

Def. 1 - Multimedia Object (MO): represents any type of multimedia data such as text, image, video, or a salient object describing an object of interest (e.g. face of a person.). It is formally represented in our approach as:

$$\text{MO: } \langle \text{id, A, O} \rangle$$

where:

- `id:` is the identifier of the multimedia object
- `A`: represents the set of textual attributes describing the multimedia object. It is formally defined as: $\langle a_1:val_1, \ldots, a_n:val_n \rangle$ where each a_i represents an element in the of Dublin Core Metadata Element set[1] (source, description, date, contributor, format, etc.), MPEG-7 semantic set[2] (semantic place, concept, state, event, object, etc.), or any keywords
- `O`: contains the raw data, the link, or a representation that characterizes the multimedia object. It is formally defined as: $\langle o_1:val_1, \ldots, o_n:val_n \rangle$ where o_i can be a BFILE, an URL/URI, or an URL/URL augmented with a primitive to represent the object (e.g. Minimum Bounding Rectangle, Circle, etc.).

Fig. 2 shows an extract of the description of multimedia objects in Fig. 1 using our multimedia type representation. For the sake of simplicity, we represent in the following a multimedia object having an identifier i as mo_i.

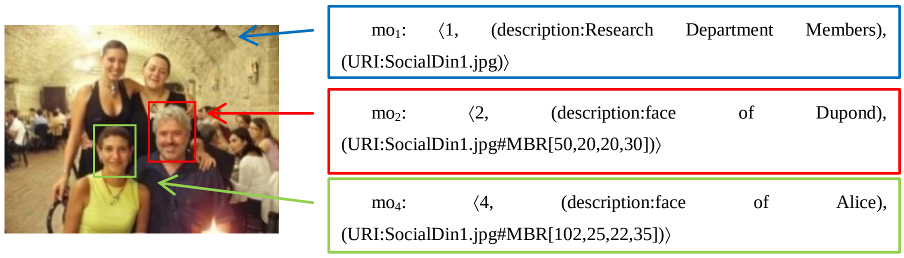

Fig. 2. Description of the Social Dinner photo with the head of the research department

Def. 2 – Sensitive Multimedia Object (SMo): is a multimedia object to be protected from unauthorized users. It is formally described as:

$$SMo: \langle Mo, C_f \rangle$$

where:

- `Mo` is (the identifier of) the multimedia object to be protected
- C_f refers to one or several multimedia protection function(s) (blur filter function, mosaic filter function, spiral filter function, substitution function, etc.). Each function can have its input parameters (e.g. the blur filter has a mosaic filter level varying from 1 to 10 (as defined in [8]). Details about C_f are omitted here due to the lack of space. For instance, hiding the identity of `Mr. Dupond` in Fig. 1 can be described as $smo_2: \langle mo_2, Blur(7) \rangle$

Def. 3 – Multimedia Relation (MR): represents a predefined multimedia relation that can link a set of multimedia objects and can be generated (automatically) using low-level features (shape, location, etc.). Each `MR` can be formally defined as:

$$MR: \langle name,\ type,\ P \rangle$$

where:

- `name` is the name used to identify the relation

[1] http://dublincore.org/documents/dcmi-terms/
[2] http://www.chiariglione.org/mpeg/standards/mpeg-7/mpeg-7.htm

- $\texttt{type} \in$ {co-occurrence, topologic[1], directional[2], temporal[3], metric[4], semantic[5]}
- $\texttt{P} \subseteq$ {reflexive, symmetric, transitive, associative} is a set of properties that characterize the multimedia relation

A MR is instantiated in our approach as a statement of the following form:

$$\texttt{MR.name}(mo_1, \ldots, mo_n)$$

For instance, the face of Mr. Dupond located to the left of Alice's in SocialDin1.jpg can be represented with Left(mo_2, mo_4).

Def. 4 - Similarity (S): is used to compare and measure the similarity between either textual descriptions or multimedia features. It is defined as:

$$S(X, Y) = \langle f_1(\langle x_1, \ldots, x_n\rangle, \langle y_1, \ldots, y_m\rangle), \ldots, f_k(\langle x_1, \ldots, x_n\rangle, \langle y_1, \ldots, y_m\rangle)\rangle$$
$$= \langle \delta_1, \ldots, \delta_k\rangle \;/\; n, m, k \in \mathbb{N}$$

where:

- $x_i \subseteq X$ and $y_j \subseteq Y$ represent the set of terms/expressions/features or multimedia objects to be compared depending on the similarity functions used
- f_i is either a set of textual similarity functions (edit distance, n-grams, etc.) or multimedia similarity functions[6]
- $\langle \delta_1, \ldots, \delta_n\rangle$ is the vector of scores returned by the similarity functions $\langle f_1, \ldots, f_n\rangle$ where $\delta_i \in [0, 1]$.

In the following, S_T and S_M will be used to designate Textual Similarity and Multimedia Similarity respectively.

Let us illustrate this, for instance, by computing the multimedia similarity between the picture of Mr. Dupond in Fig. 3 (represented as mo_{20}), and the photo of social event in Fig. 1 using the following multimedia functions f_1 and f_2:

- f_1 is related to the InterMedia Oracle module [21] based on color segments to calculate image similarity
- f_2 is based on color object recognition and SVM classifiers. It computes decisions based on a set of classes representing the trained images (See [26] for more details)

The obtained multimedia similarity has the following scoring set:

$$S_M(mo_{20}, mo_1): \langle f_1(mo_{20}, mo_1), f_2(mo_{20}, mo_1)\rangle = \langle 0.6, 0.8\rangle$$

---

[1] Such as Disjoint, Touch, Overlap, Equal, Contain, Inside, Cover, and isCoveredBy.

[2] Such as North, South, East, West, North-south, Northwest, Southeast, Southwest, Left, Right, High, Below, In front of, and Behind

[3] Such as Before, After, Touches, Overlaps, BeginsWith, EndsWith, Contains, During, and Equal

[4] Such as Far, Close, etc.

[5] It can represent what it is fixed by the nature (e.g. InLoveWith) or describe a social relationship (e.g. isMarriedTo), etc.

[6] Several multimedia functions are provided in the literature. For instance, some DBMSs such as Oracle and DB2 provide SQL-operators [16] while others are accessible via API functions [17] and web services [3]. Details on such functions and their applications are out of the scope of this paper.

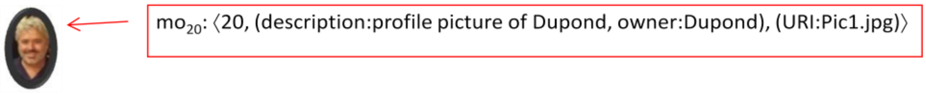

Fig. 3. Profile Picture of `Mr. Dupond`

Similarly, to compute textual similarity, two different functions f_3 and f_4 are used where:

- f_3 is a string similarity function based on the Levenshtein edit distance [19]
- f_4 is based on the number of different trigrams existing in the input text [19].

The obtained textual similarity has the following scoring set:

$$S_T(\text{"face of Dupond"}, \text{"Dupond"}): \langle f_3(\text{"face of Dupond"}, \text{"Dupond"}), f_4(\text{"face of Dupond"}, \text{"Dupond"})\rangle = \langle 0.11, 0.2\rangle^1$$

Def. 5 – Aggregation Function (μ): aggregates a set of values (returned by a similarity *S*) in order to select or compute one value to be considered. As several similarity functions can be used to compute the similarity between either textual-based or multimedia-based features, it is important to retrieve the most appropriate result for a given situation so to facilitate decision-making. An aggregation function can be defined by classical aggregation function (average, minimum, maximum, etc.) or any probabilistic function (the combination rule of Dempster and Shafer theory of evidence (DS) [22] [25], Bayesian Decision theory [23], Decision Trees [24], etc.). More details on aggregation functions can be found in [2]. It is formally written in our approach as:

$$\mu(S, \varepsilon) = \beta \in [0,1]$$

where:

- **S** is a textual or multimedia similarity (as defined in Def. 4)
- **ε** is an uncertainty threshold belonging to the interval [0,1]. It represents the percentage of uncertainty related to the combination of similarity functions used in the related similarity. In fact, ε can affect the overall scores returned by individual S_T or S_M. If omitted, ε = 0
- **β** is the (normalized) aggregated score.

For instance, by applying the average aggregation function on the result set obtained by f_1 and f_2 when comparing the picture of `Mr. Dupond` in Fig. 3 and the photo of social event in Fig. 1, we obtain the aggregated score of $\beta = (0.6 + 0.8)/(2+0.1) = 0.67$ (after having assigned 0.1 to ε related to S_M).

Def. 6 – Domain Entity (DE): represents a user, group, project, or organization in a social network. A **DE** can be formally described as:

`DE: ⟨id, name, CRED, MO⟩`

where:

- `id` is a unique identifier
- `name` is the name describing the domain entity (e.g. `Dupond`)

---

[1] Here we compare both sentences; however other techniques could be more precise to compute string similarity such as finding whether a sentence is contained in another, or comparing individual words, etc.

- CRED is a set of credentials which characterize **DE** in the social network. It is formally defined as: $\langle cred_1{:}val_1, \ldots, cred_n{:}val_n \rangle$ where each $\mathbf{cred_i}$ can represent common element in FOAF1 (name, phone, email, etc.), SIOC[2] (about, resource, etc.), or other social network descriptors. For instance, it could be location:France, University:Dijon, etc.
- MO represents a set of multimedia objects describing the **DE**.

For instance, the **DE** describing the profile of the user Dupond can be described as:

$$de_1: \langle 1, \text{Dupond}, \langle \text{location:France} \rangle, \langle mo_{20} \rangle \rangle.$$

In the following and for the sake of simplicity, a **DE** having *name* and an identifier *i* will be referenced as $name_i$.

Def. 7 – Domain Relation (DR): represents an application domain-related and/or semantic relation. A **DR** can be formally defined as:

$$DR: \langle name, type, P, Exp \rangle$$

where:

- name is the name used to identify the relation
- type $\in$ {ontologic[3], semantic}
- P $\subseteq$ {reflexive, symmetric, transitive, associative} is a set of properties that characterize the relation
- Exp is a Boolean expression used to represent (when possible) the designated relation throughout a set of **MR**. For instance, IsSittingNear.Exp = Left $\vee$ Right $\vee$ Above $\vee$ Below.

5 Proposal

The problem of determining the amount of information leakage in a set of unmasked multimedia objects MMDB is mainly related to both the representation of the application domain knowledge at hand, and the semantic gap existing between low-level features and the meaning of multimedia objects. In essence, in order to protect sensitive multimedia objects SMO contained in MMDB, one should be able to identify possible correspondences between SMO and some unmasked multimedia objects existing in MMDB through a common knowledge (to be extracted here from social networks).

To address these issues, we provide here an approach composed of two main levels holding, on one hand, information gathered from social networks and, on the other hand, the set of multimedia objects in MMDB. It includes:

1. A rich and flexible representation of social networks formally described as a Domain Knowledge (D_K) able to consider common features and multimedia descriptions and standards.

---

[1] http://www.w3.org/2001/sw/Europe/events/foaf-galway/

[2] http://www.w3.org/2008/09/msnws/papers/sioc.html

[3] isA, instanceOf, etc.

2. A framework dedicated to detect multimedia-based inference channels bearing three main modules:
   a. A *Mapping Module* (MpP): allowing to map the social networks content to the set of multimedia objects MMDB
   b. An *Inference Detection Module* (IDM): allowing to detect inference channels and to determine the amount of information leakage related to SMO
   c. An *Inference Elimination Module* (IEM): able to filter out all the inference channels detected.

In the following, we will detail our proposal components and discuss the process of detecting inference channels. The Inference Elimination Module will be detailed in another dedicated study.

5.1 Domain Knowledge (D_K)

In our approach, a Domain Knowledge (D_K) is used to organize the nodes and their relationships in social networks at hand into a semantic graph. It is formally defined as:

$$D_K: \langle N, E, W, \nu \rangle$$

where:

- N is the set of nodes representing users, groups, projects, and organizations in a social network. Each node $n \in DE$
- E is a set of edges interconnecting nodes of the social networks. An edge $e_i \in DR$. In the following, $e_i(n_1, n_2)$ and $e_i.name(n_1, n_2)$ are used interchangeably
- W is a set of values belonging to the interval $[0,1]$
- ν is a function assigning to each edge $e_i \in E$ a weight w_i, $\nu: E \rightarrow W$ so to reflect the importance of a corresponding relation on the social network.

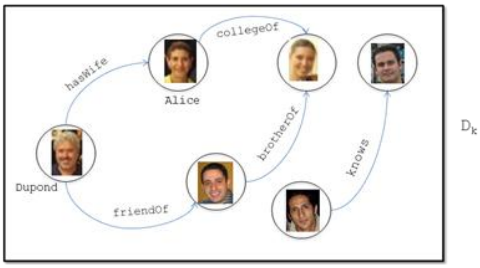

Fig. 4 Graphical representation of an extract of the social network of Mr. Dupond

In Fig. 4, we provide a graphical representation of an extract of the social network of Mr. Dupond in our running example. We do not detail here the process of transforming a social network into our representation D_K as it is straightforward and application-based.

In order to enrich the Domain Knowledge (D_K) with inferred semantics, we use a set of rules (Rg) representing derivation axioms. Each rule is defined as follows:

$$Rg: antecedent \rightarrow consequent$$

where:

- $\texttt{antecedent}$ is the body of the rule. It is formed by a set of conjunct atoms of DR relations between variable nodes written as $\texttt{antecedent= } dr_1(a, b)\ op\ \ldots op\ dr_n(x, y)$ where $op \in \{\wedge, \vee, \neg, \text{etc.}\}$ and a, b, x, y represent variables or instances of D_K
- $\texttt{consequent}$ is a DR relation representing the head of the rule.

For instance, Rg_1: $\texttt{FriendOf}(n_1,n_2) \wedge \texttt{MarriedTo}(n_2,n_3) \rightarrow \texttt{Knows}(n_1,n_3)$ states that if a node $n_1 \in N$ is a friend with a node $n_2 \in N$, then the former must know the spouse of the later.

5.2 Inference Framework

In this section we present our inference framework formed by a mapping module and an inference detection module.

5.2.1 Mapping Module (MpP)

In order to identify the correspondence between D_K content and multimedia objects in MMDB, two different but related mappings are used:

- *Node Mapping* (M_N): capable of identifying the correspondence between nodes of D_K and multimedia objects in MMDB
- *Edge Mapping* (M_E): represents the process of checking whether the edge is valid at the MMDB level according to the related DR.Exp defined.

5.2.1.1 Node Mapping (M_N)

In our approach, mapping nodes in D_K to MMDB considers the multi-criteria aspect of multimedia objects and descriptors by matching related low-level and textual features. We formally describe the node mapping M_N as:

$$M_N(mo, n) = \mu(S(X, Y), \varepsilon) \rightarrow \alpha$$

where:

- $mo \in MMDB$ is a multimedia object to be mapped
- $n \in N$ is a node of D_K
- S is the similarity between X and Y where $X \subseteq MO$ and $Y \subseteq N$
- μ represents an aggregation function[1] with its related uncertainty threshold ε used to aggregate the set of returned scores by S. The aggregated score returned by μ is compared to α in order to raise or not a mapping between the mo and n.
- $\alpha \in [0,1]$ is a the returned result of the node mapping process

For instance, in order to automatically compute the following mappings[2]:

[1] Different aggregation functions can be assigned according to the similarity functions used. That is, we could assign an aggregation function to compute values returned by textual similarity functions and Bayesian networks to compute values returned by multimedia similarity functions.

[2] Details about computation are omitted here.

M_N(smo$_2$, Dupond$_1$) = Max(Avg(S_T(smo$_2$.A, Dupond$_1$.n_name), 0.1), DS(S_M(smo$_2$.O, Dupond1.mo$_{20}$.O), 0.1))= Max(Avg(⟨0.2, 0.111⟩, 0.1), DS(⟨0.8, 0.6⟩, 0.1)) = Max(0.14, 0.78) = 0.78

M_N(mo$_4$, Alice$_2$)= Max(Avg(S_T(mo$_4$.A, Alice$_2$.n_name), 0.1), DS(S_M(mo$_4$.O, Alice$_2$.mo$_{40}$.O), 0.1))= Max(Avg(⟨0.2, 0.111⟩, 0.1), DS(⟨0.8, 0.7⟩, 0.1))= Max(0.14, 0.84) = 0.84

The algorithm of node mapping (M_N) to MMDB is given below:

Algorithm 1 [MO2N_Mapping]

Input: MMDB, D_N, DR /* MMDB *is* the set of multimedia objects, D_K is the domain knowledge DR is a specified set of domain relations */

Output: Map_Score(N, MMDB) // a matrix with score related to nodes mapped to MMDB

Begin

  For each *mo$_i$* **in** MMDB **Do**

    $n_0 = D_K(0)$ // represent a chosen node from the D_K

    Map_Score ← DFS(n_0, mo$_i$, *score*, *DR*)

  End For

  Return Map_Score

End

Algorithm 1 establishes a mapping between a set of multimedia objects MMDB and their nodes N interconnected using the set of edges DR of the domain knowledge. And so, a matrix holding mapping scores[1] of multimedia objects and nodes is retrieved. Algorithm 2 is used to search the D_K graph using the Depth First or DFS algorithm [11] to retrieve a vector of scores related to the mapping of a multimedia object mo$_i$ to the set of nodes of D_K using node mapping M_N.

Algorithm 2 [DFS]

Input:n, mo, score, DR /*n is the node to map, mo is a multimedia object ∈ MMDB, score represents the vector to hold the mapping scores between n and mo */

Output: score(n, mo)

Begin

    score ← M_N(n, mo)

    For each n_i **such that** (n,n_i) **is an edge** e_i **in DR Do**

      IF n_i **was not visited yet THEN**

        Dfs(n_i, mo, score, DR)

  End For

Return score

End

5.2.1.2 Edge Mapping (*M_E*)

Mapping edges refers to the process of finding whether any edge e_i defined at D_K level has a valid description at the MMDB level. Also, it is used to validate the

[1] Mapping nodes to their corresponding multimedia objects should be performed in the preprocessing phase due to the heavy computation time needed for processing semantic similarity between multimedia objects.

expression defined for each of the DRs at D_K level. We formally define the edge mapping M_E as follows:

$$M_E(e_i, mo_n, mo_m) = g(e_i.Exp, mo_n, mo_m)$$

where:

- e_i is an edge interconnecting two different nodes in D_K
- mo_n and mo_m are two different multimedia objects in MMDB
- *g* is a Boolean function able to evaluate the expression $e_i.Exp$ (i.e., the Boolean expression defined in e_i) with respect to (w.r.t.) mo_n and mo_m

In other words, an edge e_i, representing an DR in D_K, is mapped to MMDB if $\exists\ mo_n$ and mo_m that validate the set of multimedia relations MR contained in the Boolean expression Exp of e_i. For instance, the hasWife holding a Boolean expression hasWife.Exp = Left $\vee$ Right is mapped to MMDB if $Left(mo_i, mo_j) \vee Right(mo_i, mo_j)$ are valid for the existing multimedia objects mo_i and $mo_j \in$ MMDB. In the following, we describe our inference detection module defined to determine possible inference channels.

5.2.2 Inference Detection Module (IDM)

To detect inference channels, we define a *Multimedia based inference detection function* called MiD to detect the possible risk of inferring a sensitive multimedia object smo_m from a given multimedia object mo_n according to their corresponding mapped nodes n_1 and n_2 at D_K level. Our MiD can detect if the multimedia object mo_n infers the sensitive multimedia object smo_m w.r.t. the nodes n_1 and n_2 as:

$$MiD(mo_n \rightarrow smo_m)_{(n1,n2)} = \mu(\langle M_N(mo_n, n_1), M_N(smo_m, n_2)\rangle, \varepsilon) \times \psi((n_1, n_2))_{(mo_n, smo_m)} > \gamma$$

where:

- $mo_n, smo_m \in MMDB, n_1, n_2 \in N$, and $e_1 \in E$ linking n_1 to n_2
- $M_N(mo_n, n_1)$ and $M_N(smo_m, n_2)$ represent the scores related to the mapping between n_1 and n_2 and their corresponding multimedia objects mo_n and sensitive multimedia objects smo_m respectively.
- $\psi((n_1, n_2))_{(mo_n, smo_m)}$ is a function that returns a value representing the maximum computed weight of the set of DR between n_1 and n_2, i.e. $\psi((n_1, n_2))_{(mo_n, smo_m)} = Max\ (\cup^n_{l=1} M_E(e_l, mo_n, smo_m) \times e_l.w)$. Max could be replaced by any other aggregation function (see Def. 5) w.r.t. the domain of application. l represents the number of relations existing between the nodes n_1 and n_2. These set of relations are either directly related or inferred (w.r.t. both the set of properties P, such as symmetric, transitivity, predefined for each relation), and the predefined explicit rules Rg used to enrich the D_K. We consider that, an edge e_1 between two nodes n_1 and n_2 could provide potential inference at the MMDB level independently from the mapping direction and its symmetric property. That is, if both mo_n and smo_m are mapped to n_1 and n_2 respectively, an edge e_1 between n_1 and n_2, could be considered as possible threat whether it is defined as $e_1(n_1, n_2)$ or $e_1(n_2, n_1)$ unless the edge mapping M_E with mo_n and smo_m has computed a false

value. For example, the relation $\texttt{hasWife}(\texttt{Dupond}_1, \texttt{Alice}_2)$ between the nodes $\texttt{Dupond}_1$ and $\texttt{Alice}_2$ provides potential inference knowing that, on one hand, $\texttt{mo}_4$ (the multimedia object representing Alice) is mapped to the node $\texttt{Alice}_2$ and the $\texttt{smo}_2$ (the multimedia object representing Mr. Dupond) is mapped to the instance $\texttt{Dupond}_1$, and, on the other hand, the relation mapping $M_E(\texttt{mo}_4, \texttt{smo}_2)_{\texttt{hasWife}}$ is valid.

- γ represents a predefined threshold varying between [0, 1] on which MiD is based to determine whether the multimedia object $\texttt{mo}_n$ infers the sensitive multimedia object $\texttt{smo}_m$

An $\texttt{smo}_m$ is considered *safe* if its corresponding mapped nodes at D_K level have no upward and downward edges that could be discovered at the MMDB level leading consequently to its identification. Formally:

$\forall\ \texttt{mo}_n, \texttt{smo}_m \in \texttt{MMDB}, n_1, n_2 \in N, \text{ and } e \in E, \texttt{smo}_m \text{ is safe} \Rightarrow \nexists\ M_E(\texttt{mo}_j, \texttt{mo}_n, e) = 1, M_N(\texttt{mo}_j, n_1), M_N(\texttt{smo}_m, n_2)$ and $\texttt{MiD}(\texttt{mo}_n \rightarrow \texttt{smo}_m)_{(n_1, n_2)} > \gamma$

which means that an inference channel could be established in a multimedia environment between $\texttt{mo}_n, \texttt{smo}_m \in \texttt{MMDB}$ if their mapped nodes n_1 and n_2 respectively, are related at the social network level and the social network dependent relation e_k between n_1 and n_2 is mapped to the set MMDB.

In order to illustrate the use of MiD function, we will refer to our motivating scenario. To hide the face of Mr. Dupond (described using $\texttt{smo}_2$), one should check to see if the DR hasWife between $\texttt{Dupond}_1$ and $\texttt{Alice}_2$ could lead to the identification of Mr. Dupond. Both nodes $\texttt{Dupond}_1$ and $\texttt{Alice}_2$ are mapped to the MMDB and there exists an edge mapping that could return true for the mapped multimedia objects where the edge e_k is defined as $\langle \texttt{hasWife}, 0.7\rangle$. $w=0.7$ represents a weight reflecting the relevance of the DR at D_K level. If we wish to protect the multimedia object $\texttt{smo}_2$ representing the face of Mr. Dupond, let us determine now the possible threat due to the multimedia object $\texttt{mo}_4$. In this case, the MiD function is defined as follows:

$$\texttt{MiD}(\texttt{mo}_4 \rightarrow \texttt{smo}_2)_{(\texttt{Alice}_2, \texttt{Dupond}_1)}$$

$$= \text{Avg}\ (\langle M_N(\texttt{mo}_4, \texttt{Alice}_2), M_N(\texttt{smo}_2, \texttt{Dupond}_1)\rangle, 0.0) \times \psi(\texttt{Alice}_2, \texttt{Dupond}_1)_{(\texttt{mo}_4, \texttt{smo}_2)} > 0.5$$

The mappings are as follows:

$M_N(\texttt{smo}_2, \texttt{Dupond}_1) = 0.78$ and $M_N(\texttt{mo}_4, \texttt{Alice}_2) = 0.84$.

$\psi(\texttt{Alice}_2, \texttt{Dupond}_1)_{(\texttt{mo}_4, \texttt{smo}_2)} = M_E(\texttt{mo}_4, \texttt{smo}_2)_{\texttt{hasWife}} \times 0.7$.

Both multimedia objects $\texttt{mo}_4$ and $\texttt{smo}_2$ are mapped to nodes in D_K. Furthermore, the nodes $\texttt{Dupond}_1$ and $\texttt{Alice}_2$ are related with the edge e_k representing the hasWife DR. As $\texttt{mo}_4$ is located to the left of $\texttt{smo}_2$ in the same image SocialDin1.jpg $\in$ MMDB, the edge mapping of hasWife defined as $M_E(\texttt{mo}_4, \texttt{smo}_2)_{\texttt{hasWife}}$ is satisfied. Finally, $\psi(\texttt{Alice}_2, \texttt{Dupond}_1)_{(\texttt{mo}_4, \texttt{smo}_2)} = 0.7$ as there are no other edges between nodes $\texttt{Dupond}_1$ and $\texttt{Alice}_2$ in this example. Thus, the final result computed by MiD is: $\texttt{MiD}(\texttt{mo}_4 \rightarrow \texttt{smo}_2)_{(\texttt{Alice2}, \texttt{Dupond1})} = \text{Avg}(\langle 0.78, 0.84\rangle, 0.0) \times 0.7 = 0.567$. This means that the multimedia object $\texttt{mo}_4$ infers the sensitive multimedia object $\texttt{smo}_2$ as the result returned by the MiD is greater than the predefined threshold `0.5`. Algorithm 3 is used to highlight threatening multimedia objects that might lead to the identification of a sensitive multimedia object `smo`.

Algorithm 3 [Inference_Detection]

Input: smo, γ, Map_Score(N, MMDB), E /*smo represents the sensitive mo, γ is the inference threshold, Map_Score is the mapping matrix between nodes and MMDB, E is the predefined set of edge to consider while detecting relations between nodes */

Output: TMO // a set of threatening multimedia objects

Begin

N = retrieveNodes (smo, Map_Score (N, MMDB))
// retrieve the nodes mapped to smo from the Map_Score matrix

For each n_i **In** N

AdjN$_D$ = getDirectlyRelatedNodes (n_i, E)
//retrieve all nodes directly related to the n_i according to the edges in E

AdjN$_I$ = getInferredNodes(n_j) //retrieve all nodes inferred from either an explicit rules
// (user defined) or implicit rules (according to predefined relation properties)

AdjN = AdjN$_D$ ∪ AdjN$_I$

For each n_j **In** AdjN

MO = retrieveMO (n_j, Map_Score (N, MO)) // retrieve the multimedia objects
// mapped to n_j from the Map_Score matrix

For each mo_k **In** MO

t = MiD($mo_k \rightarrow$ smo)$_{(ni,\, nj)}$

If (t > γ)

TMO ← mo_k

End If

End For

End For

End For

Return TMO

End

The Inference Detection algorithm works as follows. First, we retrieve the set of nodes mapped to the sensitive multimedia object `smo` from the matrix `Map_Score(N, MMDB)` computed previously. For each node n_i within the retrieved set, we get its adjacent nodes according to the specified edges E at D_K level. We consider two different types of related nodes: *directly* related nodes (i.e. `hasWife(Dupond`$_1$`, Alice`$_2$`)`), and *inferred* nodes using either D_K rules or implicit relation-based rules (based on their properties i.e. `isRelatedTo(a,c)`, `isRelatedTo(c,b)` ⇒ `isRelatedTo(a,b)` when `isRelatedTo` is transitive). For each node in the set of adjacent nodes related to node n_i, we retrieve the corresponding multimedia objects using the `retrieveMO` function, according to the set of mappings already computed. We determine consequently whether a multimedia object mo_k related to `smo` is a possible threat which is determined using the predefined `MiD` function. That is, a multimedia object is considered threatening if the value returned by the `MiD` function is greater than the input value γ. The final computed result represents a set of Threatening Multimedia Objects `TMO`.

6 Conclusion and future work

In this paper, we proposed a technique to protect privacy from inference channels established in a multimedia environment by combining social networks information with unmasked multimedia objects content. Our approach is based on a generic domain knowledge in which we describe nodes and edges representing the social network data. We also proposed techniques to map these data to the set of multimedia objects to be protected. A MiD function is used to detect whether a multimedia object mo_i infers a multimedia object mo_j according to the mapped nodes and relations.

In the future work, we intent to test the efficiency of our MiD function w.r.t. different multimedia and textual mapping techniques. We further wish to tackle inference related to the returned result from multiple queries which could lead to uncovering sensitive multimedia objects.

References

1. Adam, N. R., Atluri, V., Bertino, E., & Ferrari, E. A Content Based Authorization Model for Digital Libraries. IEEE Transaction on Knowledge And Data Engineering, 296-315 (2002)
2. AL Bouna, B., Chbeir, R., & Miteran, J. MCA2CM: Multimedia Context-Aware Access Control Model. Intelligence and Security Informatics (pp. 115-123). New Brunswick, New Jersey: IEEE. (2007)
3. Alliance, ASP. (s.d.) Visited: 02 16, 2008, http://aspalliance.com/404_Image_Web_Service
4. Atluri, V., & Chun, S. A.. An Authorization Model for Geospatial Data. IEEE Tansaction on Dependable And Secure Computing , 1 (4), 238-254 (2004)
5. Bertino, E., Fan, J., Ferrari, E., Hacid, M.-S., Elmagarmid, K. A., & Zhu, X. A Hierarchical Access Control Model for Video Database Systems. ACM Trans. Inf. Syst. , 155-191 (2003)
6. Bertino, E., Ferrari, E., & Perego, A. Max: An Access Control System for Digital Libraries and the Web . COMPSAC, pp. 945-950 (2002).
7. Bertino, E., Hammad, M. A., Aref, W. G., & Elmagarmid, A. K. Access Control Model for Video Databases. 9th International Conference on Information Knowledge Management, CIKM, pp. 336-343 (2000).
8. Boyle, M., Edwards, C., & Greenberg, S. The effects of filtered video on awareness and privacy. CSCW (pp. 1-10). Philadelphia, Pennsylvania: ACM (2000).
9. Chamberlin, D. D., Gray, J., & Irving L., T. Views, Authorization, and Locking in a Relational Database System. (pp. 425-430). ACM National Computer Conference (1975)
10. Chen, Y., & Chu, W. W. Protection of Database Security via Collaborative Inference Detection. IEEE Transactions on Knowledge and Data Engineering (TKDE), Special Issue on "Knowledge and Data Management and Engineering in Intelligence and Security Informatics (2007).
11. Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. Depth-first search. Dans Introduction to Algorithms (pp. 540–549). MIT Press and McGraw-Hill (2001).

12. Delugach, H. S., & Hinke, T. H. Using Conceptual Graphs To Represent Database Inference Security Analysis. Jour. Computing and Info. Tech., vol. 2, no. 4 , 291-307 (1994).
13. Delugach, H. S., & Hinke, T. H. Wizard: A Database Inference Analysis and Detection System. IEEE Transactions on Knowledge and Data Engineering, Volume 8 , 56-66 (1996).
14. Fan, J., Luo, H., Hacid, M.-S., & Bertino, E. A novel approach for privacy-preserving video sharing. CIKM (pp. 609-616). Bremen, Germany: ACM (2005).
15. Ferraiolo, D. F., Barkley, J. F., & Khun, D. R. A Role-Based Access Control Model and Reference Implementation within a Corporate Intranet. ACM Transactions on Information and System Security (TISSEC), (2) , 34-64 (1999).
16. IBM. (s.d.). QBIC - DB2 Image Extenders. Visited : 02 16, 2008, http://wwwqbic.almaden.ibm.com
17. Lab, e. C. (s.d.). Image Processing. Visited : 01 02, 2008, http://www.efg2.com/Lab/Library/ImageProcessing/SoftwarePackages.htm
18. Landwehr, C. Formal Models of Computer Security. ACM Computer Survey, Volume 13, 247-278 (1981).
19. Lin, D. An Information-Theoretic Definition of Similarity. Madison, Wisconsin: International Machine Learning Society (1998)
20. Morgenstern, M. Controlling logical inference in multilevel database systems. IEEE Symp. on Security and Privacy (pp. 245-256). Oakland, CA, USA: IEEE (1988).
21. Network, O. T. (s.d.). Oracle Multimedia. Visited : 11 09, 2007, http://www.oracle.com/technology/products/intermedia/index.html
22. P. Dempster, A. A Generalization of the Bayesian Inference. Journal of Royal Statistical, 205-447 (1968).
23. Poole, D. Logic, Knowledge Representation, and Bayesian Decision Theory. Computational Logic (pp. 70-86). London, UK: Springer. (2000)
24. Quinlan, J. R. Induction of Decision Trees. Machine Learning, 1 (1) (1986).
25. Shafer, G. A Mathematical Theory of Evidence. Princeton University Press. (1976).
26. Smach, F., Lemaitre, C., Miteran, J., Gauthier, J. P., & Abid, M. Colour Object recognition combining Motion Descriptors, Zernike Moments and Support Vector Machine. IEEE Industrial Electronics, IECON (pp. 3238-3242). Paris - France: IEEE (2006).
27. Staddon, J. Dynamic Inference Control. Workshop on Research Issues on Data Mining and Knowledge Discovery (DMKD), (pp. 94-100). San Diego, California, USA. (2003)
28. Yang, X., & Li, C. Secure XML Publishing without Information Leakage in the Presence of Data Inference. Proceedings of the Thirtieth International Conference on Very Large Data Bases (VLDB) (pp. 96-107). Torronto, Canada: Morgan Kaufmann (2004).
29. Yip, R. W., & Levitt, K. N. Data Level Inference Detection in Database Systems. IEEE Computer Security Foundations Workshop (pp. 179-189). Rockport, Massachusetts, USA: IEEE. (1998).