A Semantic Context-Aware Privacy Model for FaceBlock

Primal Pappachan[1], Roberto Yus[2], Prajit Kumar Das[1], Tim Finin[1], Eduardo Mena[2], and Anupam Joshi[1]

[1] University of Maryland, Baltimore County, Baltimore, USA
{primal1,prajit1,finin,joshi}@cs.umbc.edu,
[2] University of Zaragoza, Zaragoza, Spain
{ryus,emena}@unizar.es

Abstract. Wearable computing devices like Google Glass are at the forefront of technological evolution in smart devices. The ubiquitous and oblivious nature of photography using these devices has made people concerned about their privacy in private and public settings. The FaceBlock[3] project protects the privacy of people around Glass users by making pictures taken by the latter, *Privacy-Aware*. Through sharing of privacy policies, users can choose whether or not to be included in pictures. However, the current privacy model of FaceBlock only permits simple constraints such as allow versus disallow pictures. In this paper, we present an extended context-aware privacy model represented using OWL ontologies and SWRL rules. We also describe use cases of how this model can help FaceBlock to generate *Privacy-Aware Pictures* depending on context and privacy needs of the user.

Keywords: Privacy, Google Glass, Semantic Web, context-aware

1 Introduction

Google Glass is a wearable device with an optical head-mounted display that enable users to take pictures just by saying "OK Glass, take a picture" or winking. Therefore, the device has raised privacy concerns because it has a readily available camera which can take pictures without anyone noticing. We developed FaceBlock [10] as a solution to preserve the privacy of users in this new scenario[4].

FaceBlock allows users to state their policy about being photographed (i.e., "I don't want my picture to be taken") by other people. To start with, FaceBlock generates an *eigenface* [8], a mathematical representation which we call a *face*

[3] http://face-block.me/

[4] Notice that FaceBlock can be used to preserve the privacy of users from pictures taken by any smart device (e.g., eyewear, smartphone, tablet, camera). However, we emphasize Google Glass, as eyewear devices are raising privacy concerns among the general public.

identifier (see Figure 1(b)) using a picture of the user's face (see Figure 1(a)). Whenever a Glass user is in the vicinity of the user, FaceBlock forms an ad hoc connection with it and sends the face identifier along with the policy. In order to enforce the policy, the FaceBlock application running on Google Glass uses the face identifier to detect if the user who shared the policy is part of the pictures taken by the device (see Figure 1(c)). It then selectively obscures the face of all the people who have sent such a policy to the device (see Figure 1(d)). Using eigenfaces helps to preserve the privacy of the sender even if the transmission is intercepted and at the same time the enforcement of the policy rule ensures the privacy in any pictures taken.

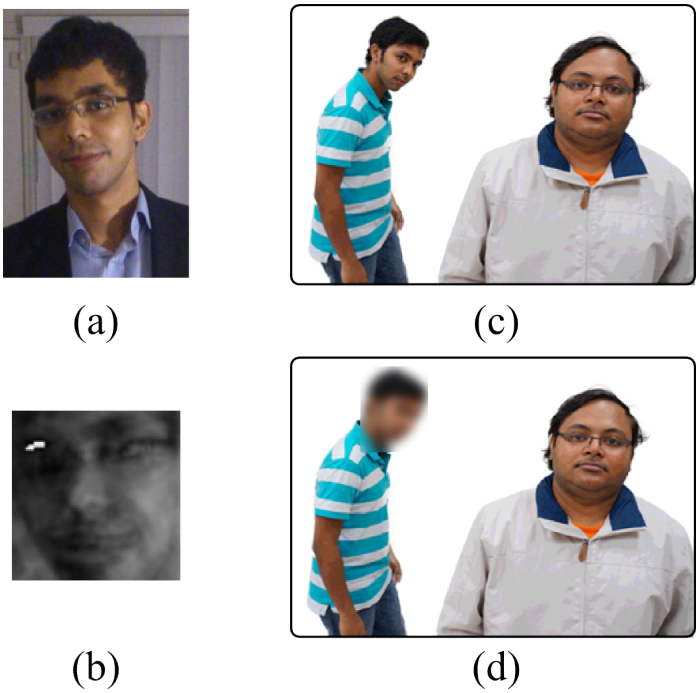

Fig. 1. Images involved in the process of obtaining a privacy-aware picture with FaceBlock: (a) picture of a FaceBlock user; (b) face identifier of the user generated by FaceBlock; (c) picture taken by a Google Glass user; and (d) privacy-aware picture generated by FaceBlock.

While being helpful in safeguarding privacy, such an all-or-nothing model does not help in many real-life situations. A person's preferences would depend on the context of the situation (e.g., time, place, activity and participants), who is taking the picture and with whom it may be shared. For example, policies like "I am okay being photographed by people I know at a private event" or "I do not like to be photographed when I am at public places". Therefore, to handle such

policies, the system should understand the semantics of concepts such as "public place", "people I know" or "private event" as well as other elements describing situation of the user.

FaceBlock's current privacy model allows a user's device to specify whether she wants her image obfuscated in pictures. Privacy-aware pictures are disabled by default privacy policy when there is no active policy sent by the other users. The current simple privacy model does not allow statements that photography is permitted, using the approach of "whatever is not explicitly prohibited, is permitted". Instead of asking everyone around if it is okay to take a picture, a Glass user might prefer or require having a positive statement from the subjects that the image can be displayed in public. Therefore, including a way for a person to grant permission is also an obvious way to improve FaceBlock.

2 Context-Aware Policies

Arguably one of the most accepted definition of the concept "context" was suggested by Dey and Abowd [1]: *"[...] any information that can be used to characterize the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and application, including the user and applications themselves."*. Dey and Abowd also decompose context into two categories: primary context pieces (i.e., identity, location, activity, and time) and secondary context pieces (context aspects that are attributes of the primary context, e.g., a user's phone number can be obtained by using the user's identity). This information about the context of a user can be modeled in an ontology (see Figure 2).

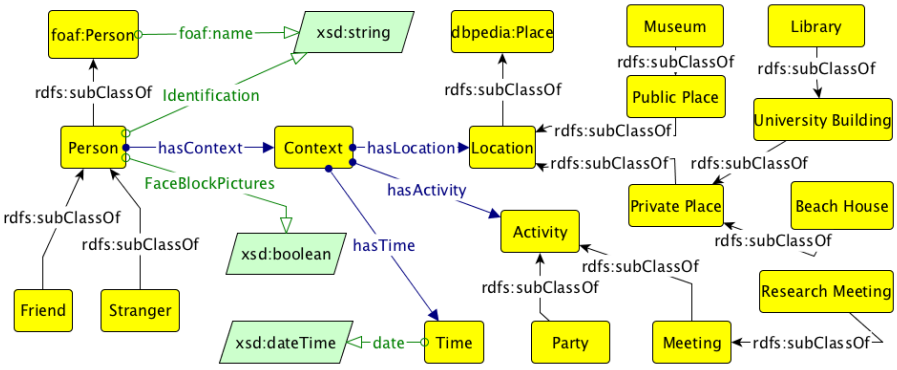

Fig. 2. Part of an initial context ontology developed for FaceBlock.

The payload of FaceBlock's current policy includes the face identifier and privacy policy (allow vs. disallow pictures). However, more fine-grained privacy

policies can be added to the FaceBlock system using Semantic Web technologies. The use of ontologies to represent privacy policies has been studied before in [7]. Incorporating context-based rules will allow a higher degree of granularity and control in the application of the policy rules for generating privacy-aware pictures. This would allow privacy policy to be applied at the specific context piece mentioned or a subclass of that concept from the ontology. In the following we present the different types of context-aware semantic policies.

Location-aware policies. The basic assumption for the FaceBlock application is that both concerned parties (the Google Glass user and a mobile user) are at the same location. This is justified by use of peer-to-peer (P2P) networks for sharing the policy between two users. In our scenario we treat location context semantically as in "at the bar", "at the University campus" or "inside my home". We define location hierarchy by referencing existing entities from linked data ontologies such as DBpedia [2] or GeoNames whenever possible. Usage of ontologies enables FaceBlock to apply privacy policy for specialization of general concepts. For example, if a student has specified that she does not want her pictures to be taken at the University buildings, it is assumed that she does not want any pictures to be taken at the University library unless specified by policy that she does not mind pictures being taken at the library.

Activity-aware policies. Activity adds another dimension to the meaning of location. For example, a classroom used for private meetings versus public lectures. Activity recognition is included in current smart devices at some extent. For example, Google recently introduced through Android API version 8 and above the ability to recognize simple activities like "in a vehicle", "on a bicycle", "running", "still, "tilting", and "on foot". For other complex activities, there are systems [6, 5, 11] that can infer the activity using information provided by the user, sensors on the device, or even information from third party sources such as calendar, event announcement, and email messages. Example policies which would be activity dependent are: "don't allow my picture when I'm dancing" (shared by a user), "don't allow pictures during meetings" (shared by a meeting room). The later policy will be applied to different types of meetings defined in the ontology (e.g., business meeting and research meeting).

Identity-aware policies. While sharing personal information such as pictures, the identity of entities (such as organizations or people) with whom it is being shared is an important aspect. The identity of user can be an unique user ID based on the device's MAC ID or a DigitalID verified by a trusted third party or other sources such as ontologies of social networks (e.g., FOAF and Facebook) or simply activity or location based (users who are at the same location and performing same activity, e.g., participants in a confidential meeting). The identification property in context ontology is used to identify the user when first encountered. Example of an identity-driven privacy policy could be: "don't allow my picture by people who are not in my close-friend circle on Google Plus

```
Person(?p) ∧ Collegue(?p) ∧ Context(?c) ∧
hasTime(?c,?t) ∧ hasDay(?t,?day) ∧ WeekendDay(?day) ∧
hasLocation(?c,?loc) ∧ BeachHouse(?loc) ∧
hasActivity(?c,?act) ∧ Party(?act)
   → FaceBlockPictures(?p,False)
```

Fig. 3. Example of a context-aware privacy policy encoded as a SWRL rule.

social network", "don't allow my picture by people who are not in my colleagues circle if I am at the office". Depending upon the ontological definition, the last policy would apply for the team members in the same office as well as remote collaborators who might occasionally visit the office.

Time-aware policies. While time is ingrained into other aspects of context, it allows for an all-embracing notion of privacy in pictures without worrying about the location and activity of users. For example, users could mention in their privacy policy that no pictures of them should be taken after 5:00 pm on Fridays irrespective of the activity or location. Time driven policy examples could be "don't allow my picture on Friday after 5:00 pm till Monday until 8:00 am", "don't take my picture during holidays". Continuing from the last example, as Spring Break would be classified as holidays, FaceBlock will be active during the time.

Composite policies. Most user situations demand a combination of various types of context-aware policies described above. Complex policies could possibly include rules that take into consideration more than one or all aspects of context as defined above. A possible example of such a scenario would be "do not allow my social network colleagues group (identity context) to take pictures of me (identity context) at parties (activity context) held on weekends (time context) at the beach house (location context)". We propose to predefine these rules using the Semantic Web Rule Language (SWRL) based on FaceBlock's context ontology (see Figure 2). For example, Figure 3 shows the SWRL rule that models the previous policy. Notice that FaceBlock creates an maintains updated an instance of the *Context* class with the current context of the user. Also, to model whether a user is allowed to take pictures or not of another person we use the data property *FaceBlockPictures(Person,xsd:boolean)*.

3 Creating Privacy-Aware Pictures

We describe the cross device process for generating privacy-aware pictures as shown in Figure 4. The user's context is represented using an OWL ontology and privacy policies are described using SWRL rules. Finally, we use a Description Logics reasoner to infer if the current context of the user matches with any of the privacy policies defined. Using wireless communications, such as Bluetooth

and WiFi, FaceBlock creates peer-to-peer networks to share the privacy policies among devices. The user device holds the ontology and reasoner and is in charge of checking the policies that should be applied (in [9] we have shown that current smart devices can handle Semantic Web technologies). Then, the Google Glass device receives a simple policy consisting of a directive to *allow* or *disallow* unobscured pictures for each user.

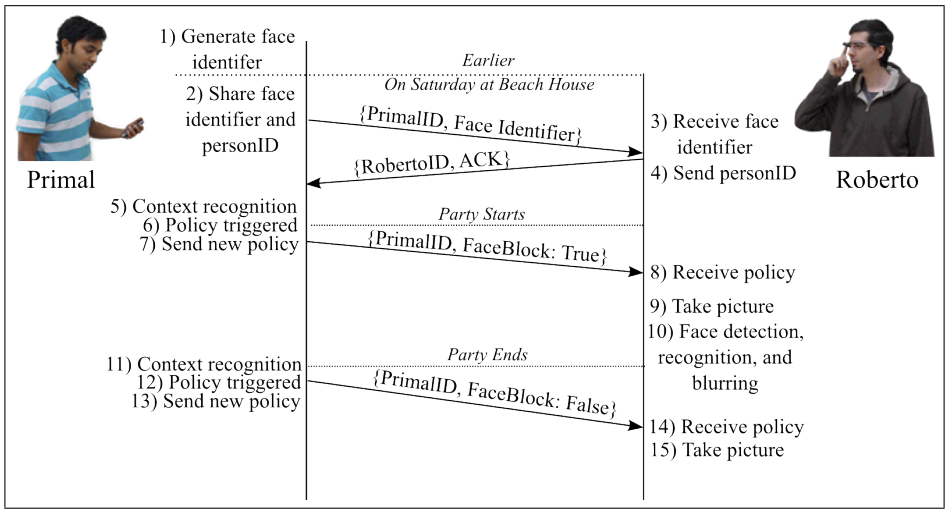

Fig. 4. Handshake diagram for the creation of privacy-aware pictures.

We use an example with two users, *Primal* and *Roberto*, to explain how the FaceBlock system works using the privacy policies. Primal is the user who wishes to protect his privacy and Roberto is the user with the device for taking pictures, that is Google Glass. Initially, Primal takes a picture of himself using FaceBlock and it generates a face identifier (*step 1* in Figure 4). He also specifies the context constraints for his pictures using FaceBlock. At the Beach House, Primal's smartphone detects and shares the face identifier with Roberto's Google Glass (*step 2*) along with a unique identification. Later, both devices periodically check whether the other device has left the surroundings by greeting/acknowledgment messages. Roberto's device receives this information, stores it and sends back his UID and an acknowledgment of the previous message (*steps 3 and 4*).

Afterwards, FaceBlock on Primal's device continuously collects information about his context and checks if any rule should be triggered by using the reasoner (*step 5*). In this case, the context has changed (the party started) and the rule presented in Figure 3 gets triggered requesting Roberto to FaceBlock pictures of Primal (*step 6*). The corresponding privacy policy for Primal is shared with Roberto's device (*step 7*). Each privacy policy has a Time To Live (TTL) associated with it during which the policy should be applied to the pictures of the user. Currently, we are using a uniform TTL for every policy. Roberto's device

accepts the privacy policy from Primal's device (*step 8*) and whenever he takes a photo FaceBlock converts it into a privacy-aware picture (*step 9*). For that, if faces are detected in the recently taken picture, FaceBlock checks if Primal is present in the picture by comparing the detected face with Primal's face identifier and obscures it (*step 10*). Thus, FaceBlock creates a privacy-aware picture for Roberto and protects the privacy of Primal.

4 Discussion and Future Work

In this paper, we have described a new policy management module for FaceBlock, an approach to preserve user privacy when taking pictures with smart devices. We have an implementation of the basic FaceBlock concept for trivial allow/disallow policies that works on Android-based devices, including Google Glass. We have also done work in our research group on context modeling, learning, acquisition, use and sharing for mobile devices [6, 11]. We are currently working on developing and implementing a new FaceBlock version that integrates these two streams. Once that is done we can evaluate its effectiveness and performance.

From a security and privacy perspective, Faceblock's framework has both advantages and disadvantages. On the plus side, a device does not need to identify its user to participate nor must it reveal any information about its context model. The use of peer-to-peer networking reduces exposure to identification by network location, especially if a device spoofs its MAC address. A device wishing to protect its user needs to provide only an eigenface and periodic *allow picture* and *disallow picture* messages. Eigenfaces enable a privacy preserving method to share face identifiers which retains enough information for FaceBlock to perform face recognition while making it difficult for humans. We acknowledge face recognition techniques can be inaccurate which might mistakenly apply the policy to someone who looks similar. We are further investigating different face recognition techniques to reduce the number of false positives and false negatives. Our architecture also assumes that each user device computes and maintains its own context, so the cost of this task is distributed.

On the other hand, like any privacy policy based solutions FaceBlock might suffer from malicious policies. Since Google Glass acts as the server for privacy-aware picture requests, a device could also launch a kind of denial of service attack on a Glass device by sending it many requests to block different eigenface images that appear to be from different devices via MAC address spoofing. Even though face identifiers and unique identification lessen the privacy loss involved in confirming the identity of users during the generation of privacy-aware pictures, we are exploring the possibility of a zero-knowledge protocol. Lastly, the picture taking device follows policies voluntarily; there is no mechanism to guarantee, or even incentivize, enforcement.

We are exploring mechanisms to assist users in defining their policies, which involve using Graphical User Interfaces (GUIs) and forms to generate SWRL rules. We are additionally exploring the possibility of supporting location-based

beacons that can broadcast organization's policies. These are the same as context-aware policies but do not require the exchange of a face identifier between participants. For example, many museums have a policy that no pictures are ever allowed (location-aware). A church might have a policy that pictures are allowed when a service is not taking place and disallowed when one is (activity-aware). This will enable FaceBlock to generate privacy-pictures for inanimate objects.

FaceBlock has to make sure that the privacy policy under consideration reflects the current context of the user. An interrupt driven approach, in which user actively enters context change into the application, would be more accurate and less power consuming. But this requires user to actively participate in the generation of privacy-aware pictures and might fatigue them. A sensor based polling approach can automatically detect context change without interference from the user at a higher power cost. We are currently exploring both these approaches to evaluate the tradeoffs of accuracy versus efficiency [4]. On the Google Glass, maintenance is performed after receiving a privacy policy by checking for other policies received from the same user and deleting them as necessary. Additionally, we would also take into consideration the cost of broadcasting change (in terms of messages, power, time) in context and optimizing it for various parameters.

The current FaceBlock protocol relies on P2P networks for information exchange between two users. But in real-life scenarios there would be more than one user in the vicinity of a Google Glass user who wishes to protect his privacy. With higher number of communicating devices, relying on a synchronous connection-oriented link over P2P networks (e.g., Bluetooth) might result in degradation of quality of service. Therefore, other wireless mechanisms, such as WiFi, or even a centralized cloud-based approach could be considered. Also in cases where multiple devices share conflicting policies, FaceBlock would require a conflict resolution mechanism.

Mechanisms for protecting user privacy in social circles have to balance between privacy requirements and the easiness of utilizing them. Privacy preference models such as P3P [3], which received considerable attention, was ignored by organizations and users due to the difficulty and lack of value. While on the other hand, licenses such as creative commons is well known and commonly used. FaceBlock not only protects users from pictures taken by others but also helps photographers to respect the privacy of others. We are taking into consideration various methods for further incentivizing the usage of FaceBlock and the generation of privacy-aware pictures so that the usage of this service becomes ubiquitous. With millions of cameras in the world, due to the explosion of mobile devices such as smartphones and tablets, mechanisms to preserve the privacy of users are needed. We believe that FaceBlock is a right step towards handling privacy needs in private and public spaces from photography devices.

Acknowledgments. This research work has been supported by the NSF grants 0910838 and 1228673, CICYT project TIN2010-21387-C02-02, and DGA FSE.

References

1. Abowd, G.D., Dey, A.K., Brown, P.J., Davies, N., Smith, M., Steggles, P.: Towards a better understanding of context and context-awareness. In: First International Symposium on Handheld and Ubiquitous Computing (HUC 1999). pp. 304–307 (1999)
2. Bizer, C., Lehmann, J., Kobilarov, G., Auer, S., Becker, C., Cyganiak, R., Hellmann, S.: DBpedia - a crystallization point for the web of data. Web Semantics 7(3), 154–165 (2009)
3. Cranor, L.F.: Web privacy with P3P - the platform for privacy preferences. O'Reilly (2002)
4. Das, P.K., Joshi, A., Finin, T.: Energy efficient sensing for managing context and privacy on smartphones. In: Int. Workshop on Society, Privacy and the Semantic Web – Policy and Technology (PrivOn 2013) (2013)
5. Ghosh, D., Joshi, A., Finin, T., Jagtap, P.: Privacy control in smart phones using semantically rich reasoning and context modeling. In: IEEE Workshop on Semantic Computing and Security (SPW 2012). pp. 82–85 (2012)
6. Jagtap, P., Joshi, A., Finin, T., Zavala, L.: Preserving Privacy in Context-Aware Systems. In: 5th IEEE International Conference on Semantic Computing (ICSC 2011). pp. 149–153 (2011)
7. Kagal, L., Finin, T., Joshi, A.: A policy based approach to security for the Semantic Web. In: International Semantic Web Conference (ISWC 2003), pp. 402–418 (2003)
8. Sirovich, L., Kirby, M.: Low-dimensional procedure for the characterization of human faces. Journal of the Optical Society of America A 4(3), 519–524 (1987)
9. Yus, R., Bobed, C., Esteban, G., Bobillo, F., Mena, E.: Android goes semantic: DL reasoners on smartphones. In: Second International Workshop on OWL Reasoner Evaluation (ORE 2013). pp. 46–52 (2013)
10. Yus, R., Pappachan, P., Das, P.K., Mena, E., Joshi, A., Finin, T.: FaceBlock: Privacy-Aware Pictures for Google Glass. In: 12th International Conference on Mobile Systems, Applications, and Services (MobiSys 2014). pp. 366–366 (2014)
11. Zavala, L., Dharurkar, R., Jagtap, P., Finin, T., Joshi, A.: Mobile, Collaborative, Context-Aware Systems. In: AAAI Workshop on Activity Context Representation: Techniques and Languages (2011)