A Private Intelligent Shopping Mall

Carlo Blundo*, Francesco Orciuoli† and Mimmo Parente*
* Dipartimento di Ingegneria dell'Informazione ed Elettrica e Matematica Applicata
† Dipartimento di Scienze Aziendali - Management & Innovation Systems
Università di Salerno
Italy

***Abstract*—In this paper we extend a previous recent work on Ambient Intelligence, deployed into a scenario of Intelligence Shopping Malls, with a privacy layer. In fact nowadays, in the Ambient Intelligence context, privacy issues are more and more considered an urgent and main issue to take care of. The success of this permeated ubiquitous intelligence seems to be strongly correlated to how much the scenario is able to protect the privacy and the rights of the users.**

The Intelligence Shopping Mall is a physical environment for commerce equipped with sensors and actuators for supporting shoppers. These latters have a wish list of the items to buy. Once in the mall, the wish list should be disclosed to steer the shopper towards the right shop selling the wished item. Anyway, from shops' point of view, shopping lists contain valuable information about shoppers. Indeed, from shopping lists one could easily infer users' personal preferences or tendency (e.g., users' lifestyle) that could be used for marketing purpose. Hence, shopping lists could reveal shoppers' sensitive information. In this paper, to preserve the shoppers' privacy without limiting the possibility to guide users towards shops selling the sought products, we propose an efficient and efficacious privacy preserving protocol. Using such a protocol, shops can steer shoppers towards the shops selling the desired items without knowing the items in their shopping lists (excluding the items bought in the shop itself).

I. Introduction

Ambient Intelligence (AmI) is a digital environment that supports people in their daily lives by assisting them in an intelligent way [1]. AmI systems have concrete environments and real occupants who interact with them. Therefore, AmI systems must be "intelligent", i.e., they have to intervene only when needed, and have to adapt their behaviour to current overall situations, users preferences and needs, and so on. AmI provides the chance to be used in numerous application scenarios, from smart homes to health monitoring and assistance, from transportation to emergency services, from education to workplaces. Moreover, the authors of [2] claim that AmI is the right opportunity to construct *Blended Shopping* ecosystems. Blended Shopping is defined by the authors of [3] as the execution of the transaction phases (information, mediation, negotiation, contracting, fulfillment and after-sales) involving both, real sales and presentation mechanisms as well as network based sales functionality.

In general terms, an AmI system is an intelligent system that surrounds the user and provides her with various and heterogeneous services. The authors of [4] provide a conceptual hierarchical model to describe a typical AmI application system. Such a model consists of five-layers encompassing the user, who is the center of the AmI system. The five-layers are the following ones:

- *Sensors and actuators*. This layer includes sensors that observe the environment and its inhabitants and actuators that act on the environment and communicate with its inhabitants. Three are the most common types of sensors (and actuators): embedded sensors (forming the equipment of the environment), wearable sensors and portable devices held by users (inhabitants of the environment).
- *AmI network and middleware*. This layer is the intelligent kernel of the AmI system. A variety of network architectures and middlewares can be deployed in this layer.
- *Devices*. This layer represents various service providing devices, such as TV, projectors, refrigerators, microwave ovens, etc.
- *Services*. This layer represents various services (location-based service, indoor navigation service, health service, etc.) provided by the indoor devices.
- *AmI applications*. This layer combines various services together according to the users requirements to provide inhabitants with potentially infinite applications.

Nowadays, in the AmI context, more and more importance is given to privacy aspects. The authors of [5] affirm that success of AmI will depend on how privacy and other rights of individuals can be protected and how individuals can come to trust the intelligent ambient that surrounds them and through which they move. In the same work, the AmI-based shopping application domain is presented as a context in which privacy issues are crucial.

The author of [6] argues that deploying and delivering personalized services needs for storing and sharing personal information. This aspect opens a challenging issue, the privacy one. Often, the privacy protection is considered more important than any potential benefits provided by technologies found in AmI applications [7]. In the last years, a growth of the number of research results that aim at mitigating the privacy and security risks of AmI has been recognized. Some of these researches focus on keeping sensed data such as location information private [8], while other activities are designing devices that can act as secure keys for providing and receiving personal information [9].

Therefore, the aim of this work is to extend the above introduced model [4] by adding a further level enabling privacy mechanisms. The layer, once theorized, will be detailed by

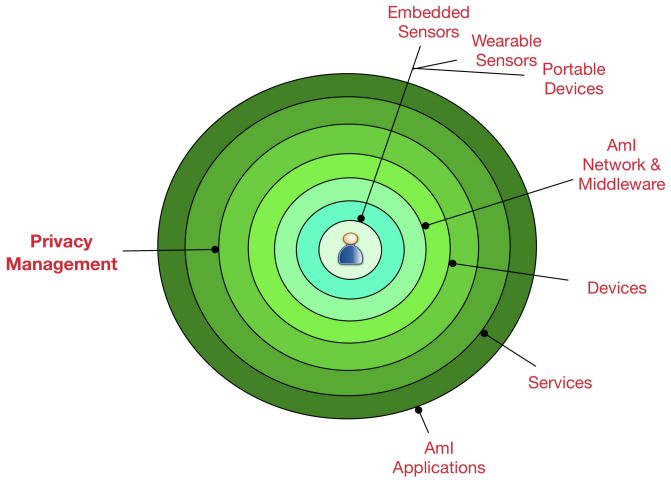

Fig. 1. Six-layers AmI Model.

considering a case study in the blended shopping domain. In particular, the case study will be focused on the definition of an Intelligent Shopping Mall, introduced in [10], in which the *AmI network and middleware* is based on the Cellular Automata (CA) model.

II. The Six-layers AmI Model

Starting from the model proposed in [4] and introduced in Section I, the idea underlying this work is to inject a further layer among *Devices* and *Services* layers. This new layer is called *Privacy Management* and has to be considered as a set of APIs available for the specialized services to configure the privacy polices and handle privacy issues. Fig. 1 shows the insertion of *Privacy Management* in the five-layers AmI model. In order to be more pragmatic and not too abstract in explaining our ideas, we decided to describe an specific instantiation of the *Privacy Management* layer in a specific scenario, the Intelligent Shopping Mall. To be more formal in describing the privacy management techniques, we adopted a concrete implementation at the *AmI network and middleware* based on Cellular Automata.

A. The Private Intelligent Shopping Mall: a concrete scenario

The scenario we choose to emphasize the needs for privacy is the domain of blended shopping and, in particular, the Intelligent Shopping Mall (ISM), a physical environment for commerce that is equipped by means sensors and actuators in order to support shoppers during their activities. In this context, the mall coordinator chooses to propose, in a fixed time slice, a set of product offerings. On the other hand, shoppers, who used a specific App (on their mobile or wearable devices) for managing shopping lists, are guided toward the shops providing the most suitable offerings for products that fit in their needs, i.e., match with specific items in their current shopping list. Thus, assume that the presence of Alice (a shopper) is recognized to be in the zone Y. The shopping list of Alice is transmitted by the processor deployed at zone Y by means of the associated sensor. At the same time, Alice receives, through her mobile or wearable device, recommendations indicating the direction she has to follow, one zone at a time, in order to move from the zone where she currently is to the shop providing the desired offering, which represents her target. These recommendations are received by Alice while she moves across different zones in the ISM as soon as a specific sensor recognizes her presence. When Alice achieves her destination (the shop in zone X providing the offering she is looking for) she can use her mobile or wearable device to gather the offering and purchase the needed product. This scenario provides advantages to both merchants and shoppers. Merchants can directly communicate their offerings to whom is interested in and agilely adapt their marketing strategies. On the other hand, shoppers can be advised (push logic) of the most suitable offerings fitting their real needs and, thus, pay less for the desidered items. In case of multiple offerings for the same product offered by more than one merchant, the ISM will suggest, one zone at a time, the path to reach the most convenient one (for instance, the one that produces the minor cost to buy the product). In this scenario, it is suitable that shoppers succeed in maintaining their privacy and do not share the information in their shopping lists with the sensing environment. Additionally, also the merchants' information could be protected when they are exchanged among the actors in the aforementioned environment.

B. Implementing the Middleware layer by using Cellular Automata

We chose to implement the AmI Middleware (see the model in Fig. 1) by using the Cellular Automata (CA) model in order to stress the aspects related to merchants' privacy in the ISM. In the best of our knowledge, this is one of the first tentatives to fully describe an AmI system by using only one formal computational model that has, among the other benefits, the following suitable characteristics: i) it can be easily used to represent a digital-physical intelligent environment; ii) it is natively flexible and scalable; iii) it can be implementable by using low-cost hardware. In fact, CA allows merchants' data to pass through processors deployed in other merchants' shops. In this situation, malicious management of processors could allow some merchants to take advantages knowing other merchants' offering details. In order to model a generic ISM with a CA, we consider a mesh (clearly it is possible also to choose alternative topologies) as depicted in Fig. 2, where each cell is connected to a number of neighbors, that is less or equal to eight, by considering the King's-move arcs (labeled by the compass directions: E , SE, S, SW, W, NW, N, NE). Each cell represents the *processor* of a specific zone within the ISM, receives input from the *sensors* deployed in such zone and provides output to the *actuators* and to the neighbors. The employment of CA as an algorithmic platform to design and solve problems for ISMs and, in particular, to define location-based services in the domain of blended commerce/shopping allows to effectively model a physical equipped environment allowing it to scale both horizontally

and vertically. In the commerce domain, the above mentioned characteristic natively allows the implementation of interesting commerce strategies like, for instance, geo-marketing. In fact, shoppers (the inhabitants) can be reached, in every zones they are, by means of context-aware information only if they need them. Fig. 2(a) indicates the existence of a *Management System for Products* that is needed to manage a product database useful to handle product offerings and to support the necessary centralized operations.

III. The Cellular Automata and the Algorithm

Cellular automaton (CA) consists of a regular *network* of *extremely simple computers*, (called *cells*) which are essentially Finite State Machines (FSMs). They have been studied since early $'60$s and are still investigated mainly because they combine a mathematical simplicity and elegance with an high level of computational efficiency and efficacy that makes them very suitable to implement real case scenarios [11], [12], [13], [14], [15].

The cells operate synchronously, at discrete time unit. At each time t, a *configuration* specifies the state of each cell. Time is discrete, and at each time step each cell is in one of a finite number of *states*. A *neighborhood* relation is defined, indicating the *neighbors* of each cell. All the cells have the same number N of neighbors, except a fixed number of *boundary cells* which have less neighbors (throughout our paper $N = 8$). A cell is intended to be linked to each of its neighbors through *communication channels* and can send and receive, at each time step, *messages*), which are binary sequences whose length is bounded by the constant capacity of the channels. At each time step, every cell updates its state in accordance to a *state-transition function* δ that takes as input the state of the cell itself along with the messages received from the cells in its neighborhood. A *computation step* modifies the configuration, in accordance to the transition function and depending on both the current configuration and the sequences sent by the cells. An *initial configuration* is a configuration at time 1. Observe that in the classical definition of CA, the transition function takes as input the state of the cell itself and those of its neighbors at the previous step. This classical definition is captured here when the capacity of the channels is $log|Q|$, where Q is the set of states of the CA, thus each cell can send its whole state in a single step.
In [16], [17] the author proposes the definition of a set of algorithms running over the Cellular ANTomata. and among them the Food Finding algorithm was described that, whose inputs and outputs naturally maps on our scenario and thus allows us to provide an efficient and effective solution for our Intelligent Shopping Mall scenario. The ANTomata are classical cellular automata with the feature that each cell is equipped with sensor for ants and goals. This way the messages flow through the network "below the surface that objects (ants and food) reside on". In our scenario the shopper plays the role of the ant whose goal is to reach the products listed in her wishlist.

A. Cellular ANTomata Algorithm

Recall that we have established a map among physical zones (in the shopping mall) and cells (in the CA). If a shopper is in a specific zone, the cell corresponding in the model is aware of this presence by means of the *sensors* occurring in the zone, the cell can elaborate this information and produces results by means of its *processor* and, lastly, it is able to interact with such shopper by using its *actuators*. Thus, a shopper is localized at a specific zone in the mall and communicates its wishlist to the corresponding cell. The cell has the task to recommend the next move to the shopper to let her reach the zones selling products that match one or more items in the shopper's wishlist. Once the shopper received the recommendations, she can follow one of them or she is free to move autonomously in the mall. Recommendations are represented by single steps toward the next zone to reach. These can be graphically presented to the shopper as proposed in Fig. 3 where the shopper receives the first recommendation that invites her to go ahead in the corridor and reaching zone c_x. When the shopper has moved to c_x, she receives a new recommendation inviting her to go toward the left corner to reach shop s_y. Labels c_x and s_y are zone identifiers that can be also replaced by intelligible names as shop names that can be simply recognized by the shoppers.

Thus, shoppers need the recommendation and some markers (e.g., zone names) in the environments in order to move according to the Intelligent Shopping Mall. A special kind of recommendation invites the shopper to buy a product (matching one in the wishlist) in the zone where she currently is. Of course, this zone is a shop.

Recommendations are generated by means of a distributed algorithm running over the CA. The algorithm, generally resembles the Food-Finding-Algorithm of [17]. The shoppers in the mall are the ants within the CA. Shoppers look for products (with suitable costs) in the mall as ants look for foods. In particular, in order to recommend a direction to a shopper for products in her wishlist, each cell (a zone in the mall) must match the items in the shopper's wishlist with the list of products which are in the shop or in case there are none (or the cell is associated to a zone that is not a shop), what is the next move to reach the products in the mall. The former point is satisfied by means of the sensors gathering information from the shoppers smartphones. The latter point is accomplished by exchanging a particular directive (sometimes we call it also *piece of information*), called I-HAVE-PRODUCT, over the CA. Due to lack of space the message exchange algorithm is not described here in detail. Informally the algorithm works as follows: every cell in the CA selling a product (managed by the algorithm) sends a message to all their neighbours at each time tick (step), repeatedly. The message provides information about both the product and the direction to follow in order to reach it, this information are available at the next time step. Fig. 4 shows two of the eight directives broadcasted to the neighbours to inform that node x has a product P and how to reach it. Each one of the eight messages is contextualized by

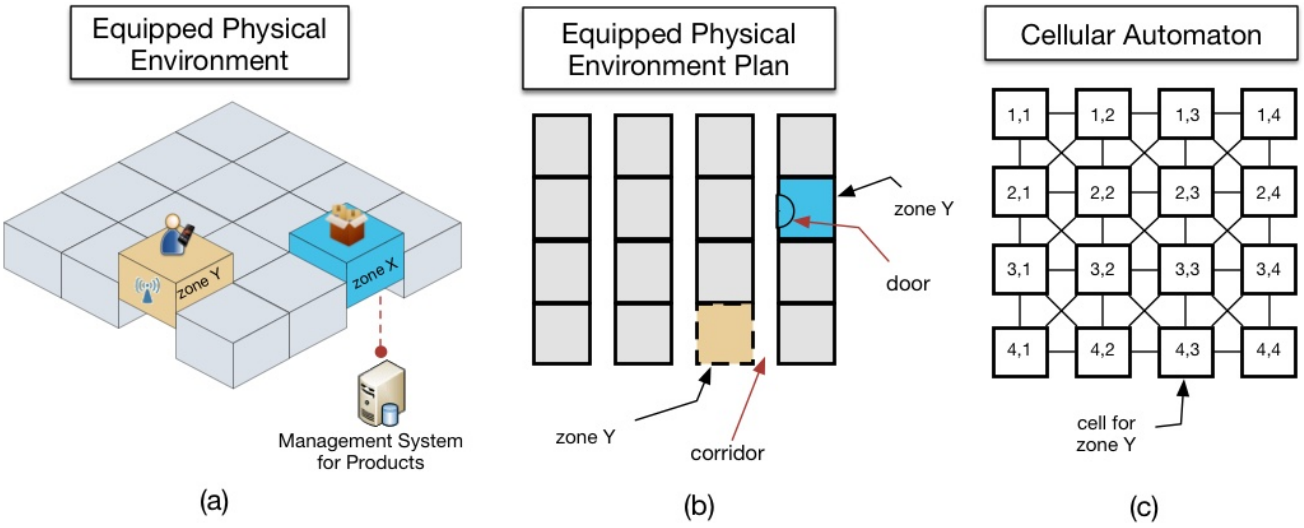

Fig. 2. Intelligent Shopping Malls with Cellular Automata

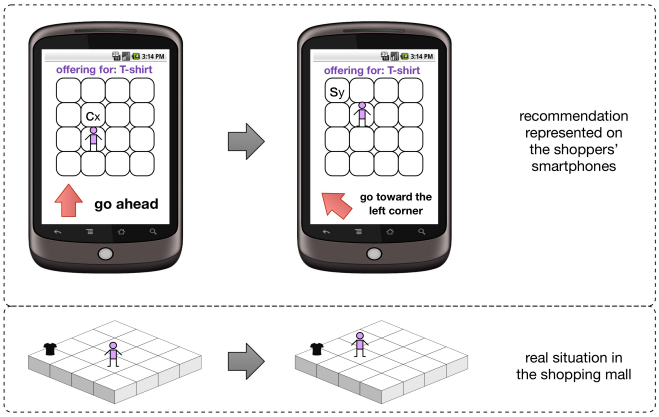

Fig. 3. Graphically representation of recommendations for shoppers.

considering its destination. In fact, the message for y invites to follow the direction SW (`south-west`).

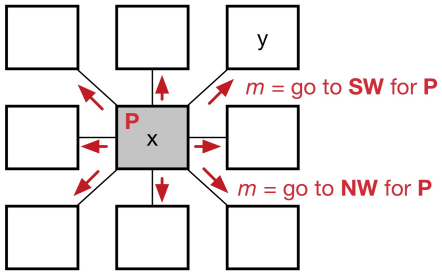

Fig. 4. Examples of I-HAVE-PRODUCT directive (broadcast).

Until the shop x sells its product P to some shopper, the I-HAVE-PRODUCT directive is broadcasted by the shop at each subsequent time step. In order to reach also cells/zone that are not neighbors of a broadcaster, a propagation mechanism is needed. The idea is that once a cell receives a message it must *relay* this directive to a subset of its neighbors.

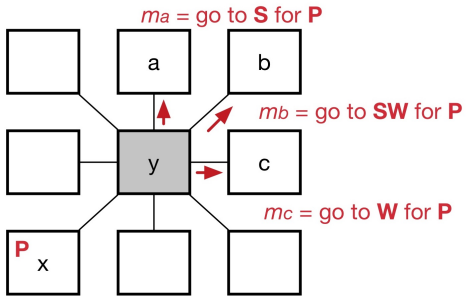

Fig. 5. Examples of RELAY directive.

Fig. 5 shows the RELAY operation. In particular, the original directive started from x at time $t+1$, will arrive at y at time $t+2$ and from here it is relayed. Thus at time $t+3$ the relayed directives arrive to a, b and c. Actually in the implementation, these two *directives* are merged and sent in a *unique message*.

By using this approach, after a number of iterations (steps), each cell is aware of the directions to recommend to possible shoppers for all the products managed by the intelligent system.

IV. The Privacy Layer

In this section we add a privacy layer to the algorithm described in Section III. We consider the wishlists (shopping list) a valuable information that should be kept private. The privacy layer we add will hide to shops the products in the wishlists, while allowing users to follow the directions (recommendations) where the sought products are sold. These seem two contrasting requirements, but we can satisfy both

by resorting to suitable cryptographic primitives. To hide the product ids one could simply encode (i.e., encrypt) them. Notice that, during the shopping experience, at some point, encoding should be compared to suggesting recommendations. We do not use a randomized encoding (i.e., randomized encryption) as private preserving comparison of randomized encoding are more time expensive than their deterministic counterparts. Moreover, in our scenario, we do not need to recover the encoded message (i.e., product's id) but just to compare it to other encoded messages. Therefore, resorting to public (private) key encryption scheme supporting equality test whould be overkill. We could simply encode products ids by using a hash function H (i.e., a function that maps arbitrary size data to fixed size data) as it cannot be inverted, but anybody can check whether its value, say y, maps a message m by computing $z = H(m)$ and testing whether z equals y. This solution provides no privacy at all as we assume that products' ids are publicly known. Therefore, our solution should be based on some secret information, should be deterministic, and should be secure. We could use either symmetric encryption or a keyed-hash message authentication code (HMAC) defined below. Since, in our setting, we do not need to decrypt the encoded procuct ids, we can resort to HMAC. Once encoded the products ids, the *intelligence* in the mall should compare user's shopping list with the shops' recommendations. We could simply solve this problem by letting the user sending his/her encoded shopping list to the nearest shop. The shop, comparing his/her recommendations with the user's shopping list (i.e., computing a set intersection), could suggest to the user where to move next. This solution leaks some information on the user's list (i.e., its length) and on the direction taken by the user. We can avoid such a privacy leaking by resorting to a Private Set Intersection protocol defined below. Thus, to sum up, in our privacy preserving algorithm we will use two cryptographic primitives, namely the HMAC and the Private Set Intersection.

Below, we briefly recall their definition.

Informally, an HMAC of an arbitrary message m for a given key k (i.e., $\mathsf{HMAC}(k, m)$) is a digest of m computed by cleverly applying a cryptographic hash function H to m using the key k. A cryptographic hash function is a mechanism used to guarantee integrity of information (i.e., it assures that data has not been tampered with). It compresses large messages down to smaller (fixed-length) ones. According to [18] $\mathsf{HMAC}(k, m)$ can be defined as follows:

$$\mathsf{HMAC}(k, m) = \mathsf{H}(k \,||\, \mathsf{opad} \,||\, \mathsf{H}(k \,||\, \mathsf{ipad} \,||\, m)).$$

Assume that the cryptographic hash function H, on input m, outputs a digest of d bytes by iterating a basic compression function on m's blocks of b bytes. Then, the key k is a random string of any length less than b and

$$\begin{aligned} \mathsf{ipad} &= \text{the byte 0x36 repeated b times} \\ \mathsf{opad} &= \text{the byte 0x5C repeated b times} \end{aligned}$$

If there are space constraints, as it could be our setting, we can resort to a *truncated* HMAC that is, instead of considering the whole digest, we can use only t bits (say, the first ones). In [18] it is recommended that t be at least half the length of the hash output and not less than 80 (i.e., according the above description $t \geq \max\{80, 4\mathsf{d}\}$).

Private Set Intersection (PSI) is a cryptographic primitive involving two interacting parties: Client with input $C = \{c_1, \ldots, c_w\}$ and Server with input $S = \{s_1, \ldots, s_v\}$. At the end of the interaction, Client learns $C \cap S$, and Server – nothing. Using traditional secure two-party computation definitions (see [19]), and assuming wlog that $|A| = v$ and $|B| = w$, the PSI functionality can be described as the secure implementation of: $\mathcal{F}_{\mathsf{PSI}} : ((C, w), (S, v)) \mapsto (C \cap S, \perp)$. Previous notation means that Client, with input C and w (i.e., the size of the set held by Server) interacting with Server, with input S and v, computes $C \cap S$ while Server learns nothing (denoted by $\perp$). We refer to [20], [21], [22] for the description of some PSI protocols.

Before describing how to modify the algorithm described in Section III to add a privacy layer, in the next subsection we need to set up our notation.

A. Setting

We denote by $\mathcal{S}$ the set of all shops in the mall; while, we will use the small letter s to refer to a generic shop in $\mathcal{S}$. We assume that products in the shopping mall, even though they are sold by different shops, can by identified by a unique alphanumeric reference (e.g., ID). In other words, we assume that products' ID are independent of the shops they are sold. Finally, we denote by $\mathcal{U}$ the set of all users (i.e., shoppers) in the mall and by $\mathsf{u} \in \mathcal{U}$ a generic shopper. We use $\mathsf{ShoppingList}_{\mathsf{u}}$ to represent the list of products' IDs user u wants to buy.

In this *new* scenario, we assume that the set of products recommended (available) to the shoppers are the ones sold at minimum price within the mall. We will refer to such a set as MinimumPrice. The central authority, hereinafter referred to as MallManager, collecting the products' prices from all shops, determines where any given product is sold at minimum price. Then, the central authority defines $\mathsf{MinimumPrice}_{\mathsf{s}} \subseteq \mathsf{MinimumPrice}$ as the set of minimum-price products available at shop $\mathsf{s} \in \mathcal{S}$. We will show later how the MallManager computes the set $\mathsf{MinimumPrice}_{\mathsf{s}}$ for any $\mathsf{s} \in \mathcal{S}$

B. The Private Protocol

We can add a privacy layer to the protocol described in Section III by simply modifying how the shops' *recommendations* are computed and by representing the sets of products available in shops in a different way. Assuming that the number of distinct products available in the whole mall is n (i.e., $n = |\mathsf{MinimumPrice}|$), the algorithm in Section III encodes the I-HAVE-PRODUCT and RELAY directives as n-bits messages. The I-HAVE-PRODUCT directive is encoded by a characteristic vector of n bits representing the occurrence of the products in the shop. Similarly, the RELAY directive is encoded by a characteristic vector of n bits representing that the products are reachable through the shop (i.e., by following

the RELAY directives). The algorithm in Section III combines the I-HAVE-PRODUCT and RELAY directives by computing the bit-wise OR of some characteristic vectors. For instance, the binary message p_{M^E} sent by a generic shop s towards East is

$$p_{M^E} = q_{\mathsf{s}} \vee \mathbf{in}_{NW} \vee \mathbf{in}_{W} \vee \mathbf{in}_{SW},$$

where q_{s} is the n-bits characteristic vector representing the occurrence of the products in the shop s, while, for $Y \in \{N, NE, E, SE, S, SW, W, NW\}$, $\mathbf{in}_Y$ is the message received by shop s from direction Y (i.e., message representing the combination of I-HAVE-PRODUCT and RELAY directives) at the previous time step.

In this section, instead of representing a set through its characteristic vector, we will represent it as a collection of elements. The boolean operations executed by the algorithm to combine I-HAVE-PRODUCT and RELAY directives into one single message are substituted by the corresponding set operations. Overall, the algorithm remains unchanged.

The MallManager, at the beginning of each day, randomly generates a daily key k to be used for computing the $HMAC$ of products' ID. At the beginning of each day the MallManager

- Collects the prices of all products sold by the shops in the mall, i.e., the MallManager receives from each shop $\mathsf{s} \in \mathcal{S}$ the set $\mathsf{PriceList_s} = \{\langle ID, price_{ID}\rangle\}$, where product ID is sold by shop s at price $price_{ID}$.
- For each product ID, determines in which shop, say shop s, it is sold at the minimum price.
- Computes the value $\mathsf{HMAC}(k, ID)$ and adds it to the set $\mathsf{MinimumPrice_s}$.
- Sends the set $\mathsf{MinimumPrice_s}$ to shop s.

Notice that, due to the $HMAC$ security, any shop s, not knowing key k, cannot determine which products *belongs* to the sets $\mathsf{MinimumPrice_s}$ (i.e, he does not know which products he sells at the minimum price). Shop s can get some information only if either $|\mathsf{PriceList_s}| = |\mathsf{MinimumPrice_s}|$ (i.e., s will learn that he is the cheapest shop in the mall) or $|\mathsf{MinimumPrice_s}| = 0$ (i.e., the shop will learn that other shops in the mall sell his products at a lower price).

Any shopper $\mathsf{u} \in \mathcal{U}$ entering the mall receives from the MallManager the daily $HMAC$ key k. Then, she computes the *encoded* version of her shopping list $\mathsf{ShoppingList_u}$ by computing the $HMAC$ under key k of the products' ID she is interested to. User u stores, into two different sets, the computed $HMAC$s and the the tuples (ID, $HMAC$). More formally, user $\mathsf{u} \in \mathcal{U}$ computes the following sets:

$$\mathsf{ESL}_\mathsf{u}^{(1)} = \{\langle ID, \mathsf{HMAC}(k, ID)\rangle \,|\, ID \in \mathsf{ShoppingList_u}\},$$

$$\text{and } \mathsf{ESL}_\mathsf{u}^{(2)} = \{\mathsf{HMAC}(k, ID) \,|\, ID \in \mathsf{ShoppingList_u})\}.$$

At this point, we just need to show how the basic operation of the algorithm should be modified in order to add a privacy layer. For any $\mathsf{s} \in \mathcal{S}$ and $\mathsf{dir} \in \{N, NE, E, SE, S, SW, W, NW\}$, we denote by $\mathsf{MP}_\mathsf{s}^\mathsf{dir}$ the set of encoded minimum prices received from shop s by the neighbor located towards dir. Message $\mathsf{MP}_\mathsf{s}^\mathsf{dir}$ corresponds to message $\mathbf{in}_\mathsf{dir}$ used by algorithm in Section III. Once, shop s receives, for all $\mathsf{dir} \in \{N, NE, E, SE, S, SW, W, NW\}$, $\mathsf{MinimumPrice}_\mathsf{s}^\mathsf{dir}$ he can compute the messages to be sent to his neighbors. For instance, the message that will be sent to neighbor located towards E is computed as

$$\mathsf{MinimumPrice_s} \cup \mathsf{MP}_\mathsf{s}^{NW} \cup \mathsf{MP}_\mathsf{s}^{W} \cup \mathsf{MP}_\mathsf{s}^{SW}.$$

This is similar to the execution of the algorithm in Section III where boolean operations to merge I-HAVE-PRODUCT and RELAY directives into one single message are substituted by the corresponding set operations. Indeed, the previous message corresponds to the message p_{M^E} computed by the algorithm in Section III.

Now, we can describe how the user u interacts with shops during her shopping. A simple solution would be for the user to send to shop s her list $\mathsf{ESL}_\mathsf{u}^{(2)}$. Shop s computes the following set intersections

$$\mathsf{Items} = \mathsf{MinimumPrice_s} \cap \mathsf{ESL}_\mathsf{u}^{(2)}$$

and

$$\mathsf{Items}_\mathsf{dir} = \mathsf{MP}_\mathsf{s}^\mathsf{dir} \cap (\mathsf{ESL}_\mathsf{u}^{(2)} \backslash \mathsf{Items}),$$

for $\mathsf{dir} \in \{N, NE, E, SE, S, SW, W, NW\}$. Then, shop s sends back the results to user u that can make her choice (i.e., where to buy the products she is looking for). It is clear that such simple protocol leaks some information to shop s as it leaks the directions where there can be found the products the user is interested and towards the user will probably move next.

We can avoid such a privacy leaking by resorting to some runs of a Private Set Intersection protocol. Indeed, user $\mathsf{u} \in \mathcal{U}$, approaching a shop $\mathsf{s} \in \mathcal{S}$ engages with s nine runs of a Private Set Intersection protocol where she plays the role of the Client (i.e., she will learn the intersection), while the shop engages the protocol as the Server (learning nothing at the end of the protocol). In particular, in the first run u's input is $\mathsf{ESL}_\mathsf{u}^{(2)}$, while S's input is $\mathsf{MinimumPrice_s}$. At the end of the protocol, u will privately compute $\mathsf{Items} = \mathsf{MinimumPrice_s} \cap \mathsf{ESL}_\mathsf{u}^{(2)}$. Notice that, in this case, the user u will compute the intersection while the shop s does not gain any information on the user's shopping list. If $\mathsf{Items} \neq \emptyset$, then shopper u will know what products are sold by shop s. Indeed, for any $val \in \mathsf{Items}$, she will lookup val in $\mathsf{ESL}_\mathsf{u}^{(1)}$. In the next eight runs, u's input[1] is $\mathsf{ESL}_\mathsf{u}^{(1)} \backslash \mathsf{Items}$, while s's input is $\mathsf{MP}_\mathsf{s}^\mathsf{dir}$ for $\mathsf{dir} \in \{N, NE, E, SE, S, SW, W, NW\}$. At the end of the eight runs, u will privately compute

$$\mathsf{Items}_\mathsf{dir} = \mathsf{MP}_\mathsf{s}^\mathsf{dir} \cap (\mathsf{ESL}(U) \backslash \mathsf{Items}),$$

for $\mathsf{dir} \in \{N, NE, E, SE, S, SW, W, NW\}$. Then, user u, analyzing $\mathsf{Items}_\mathsf{dir}$ and following her own policy, will head towards a new direction.

[1]We are assuming that user's policy is to buy, as soon as possible, all products in her shopping list. This means that if $\mathsf{Items} \neq \emptyset$, then all products identified by Items will be bought by the shopper at shop s and removed from her shopping list.

The above sketched protocol guarantees user's privacy. Anyway, a major concern with the previous protocol is that any user in the mall knows the daily key k. Therefore, any malicious user could leak such a key to a shop that can try to gain some information from the messages exchanged during the algorithm run. For instance, the shop can check whether a product identified by ID is sold in some shop towards dir simply checking whether $\mathsf{HMAC}(k, ID)$ belongs to $\mathsf{MP}^{\mathsf{dir}}\mathsf{s}$. We plan to address to such an issue in the final version of this paper. Resorting to a *Deterministic Commutative Encryption Scheme* could solve this problem. Anyway, any devised solution cannot avoid that a malicious shopper leaks his/her private key to a shop allowing it to discover offers and products from a competitor shop.

V. Conclusions and Future Works

This paper focuses on the enhancement of the existing Five-layers AmI Model by adding a *Privacy Management* layer that is critical in numerous application scenarios. In order to demonstrate the applicability of such enhanced model we proposed the Private Intelligent Shopping Mall, an application scenario in which it is needed to face privacy issues, in particular, with respect to shoppers' informations. Furthermore, the adoption of Cellular Automata, as a formal computational model implementing the *AmI Network and Middleware* layer, stresses privacy issues also for merchants who need to share their offerings across all the environment. Of course, other alternative approaches are plausible for defining the intelligent shop but cellular automata allows to provide a solution that is natively flexible, scalable and robust and that is easily implementable by using low-cost components. Furthermore, cellular automata represents a framework that can be easily configured to face different blended commerce scenarios (not only the one provided by the paper). Lastly it is possible to provide simple simulations of cellular automata to early test the effectiveness and the realizability of scenario-specific AmI system. In the future, we will focus on the contextualization of the proposed model in new blended commerce scenarios and in further different domains.

References

[1] A. Vasilakos and W. Pedrycz, *Ambient Intelligence, Wireless Networking, And Ubiquitous Computing*. Norwood, MA, USA: Artech House, Inc., 2006.

[2] M. Gaeta, V. Loia, F. Orciuoli, and M. Parmentola, "A genetic approach to plan shopping in the ami-based blended commerce," in *2013 IEEE International Symposium on Industrial Electronics (ISIE)*, May 2013, pp. 1–6.

[3] B. Fuchs, T. Ritz, B. Halbach, and F. Hartl, "Blended shopping: Interactivity and individualization," in *e-Business (ICE-B), 2011 Proceedings of the International Conference on*, July 2011, pp. 1–6.

[4] C. Rui, H. Yi-Bin, H. Zhang-Qin, and H. Jian, "Modeling the ambient intelligence application system: Concept, software, data, and network," *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, vol. 39, no. 3, pp. 299–314, 2009.

[5] M. Friedewald, E. Vildjiounaite, Y. Punie, and D. Wright, "Privacy, identity and security in ambient intelligence: A scenario analysis," *Telematics and Informatics*, vol. 24, no. 1, pp. 15 – 29, 2007. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0736585305000778

[6] D. Wright, "The dark side of ambient intelligence," *info*, vol. 7, no. 6, pp. 33–51, 2005.

[7] D. J. Cook, J. C. Augusto, and V. R. Jakkula, "Ambient intelligence: Technologies, applications, and opportunities," *Pervasive and Mobile Computing*, vol. 5, no. 4, pp. 277–298, 2009.

[8] Y. Cho, S. Cho, D. Choi, S. Jin, K. Chung, and C. Park, "A location privacy protection mechanism for smart space," in *International Workshop on Information Security Applications*. Springer, 2003, pp. 162–173.

[9] F. W. Zhu, M. W. Mutka, and L. M. Ni, "The master key: A private authentication approach for pervasive computing environments." in *PerCom*, 2006, pp. 212–221.

[10] F. Orciuoli and M. Parente, "An agent-based framework for indoor navigation in blended shopping," in *IEEE Symposium Series on Computational Intelligence, SSCI 2015, Cape Town, South Africa, December 7-10, 2015*, 2015, pp. 640–646.

[11] J. Gruska, S. La Torre, and M. Parente, "Optimal time and communication solutions of firing squad synchronization problems on square arrays, toruses and rings," in *Developments in Language Theory, (DLT)*, ser. Lecture Notes in Computer Science, vol. 3340. Springer, 2004, pp. 200–211.

[12] ——, "The firing squad synchronization problem on squares, toruses and rings," *Int. J. Found. Comput. Sci.*, vol. 18, no. 3, pp. 637–654, 2007.

[13] S. La Torre, M. Napoli, and D. Parente, "Synchronization of a line of identical processors at a given time," *Fundam. Inform.*, vol. 34, no. 1-2, pp. 103–128, 1998. [Online]. Available: http://dx.doi.org/10.3233/FI-1998-341204

[14] E. F. Moore, "The firing squad synchronization problem," *Sequential Machines, Selected Papers*, pp. 213–214, 1962.

[15] H. Umeo and K. Kubo, "A seven-state time-optimum square synchronizer," in *Cellular Automata*, ser. Lecture Notes in Computer Science, S. Bandini, S. Manzoni, H. Umeo, and G. Vizzari, Eds. Springer, 2010, vol. 6350, pp. 219–230.

[16] A. L. Rosenberg, "Cellular antomata: food-finding and maze-threading," in *Parallel Processing, 2008. ICPP'08. 37th International Conference on*. IEEE, 2008, pp. 528–535.

[17] ——, "Cellular antomata," *Advances in Complex Systems*, vol. 15, no. 06, p. 28, 2012.

[18] H. Krawczyk, M. Bellare, and R. Canetti, "Hmac: Keyed-hashing for message authentication," in *RFC 2104*, 1997.

[19] O. Goldreich, *Foundations of Cryptography*. Cambridge University Press, 1994.

[20] E. D. Cristofaro and G. Tsudik, "Experimenting with fast private set intersection," in *Trust and Trustworthy Computing - 5th International Conference, TRUST 2012, Vienna, Austria, June 13-15, 2012. Proceedings*, 2012, pp. 55–73.

[21] G. Ateniese, E. D. Cristofaro, and G. Tsudik, "(if) size matters: Size-hiding private set intersection," in *Public Key Cryptography - PKC 2011 - 14th International Conference on Practice and Theory in Public Key Cryptography, Taormina, Italy, March 6-9, 2011. Proceedings*, 2011, pp. 156–173.

[22] C. Blundo, E. D. Cristofaro, and P. Gasti, "Espresso: Efficient privacy-preserving evaluation of sample set similarity," *Journal of Computer Security*, vol. 22, no. 3, pp. 355–381, 2014.