Fully Distributed Secure Video Surveillance Via Portable Device with User Awareness

Arcangelo Castiglione, Ciriaco D'ambrosio, Alfredo De Santis, Francesco Palmieri

Fully Distributed Secure Video Surveillance Via Portable Device With User Awareness

Arcangelo Castiglione[1⋆], Ciriaco D'Ambrosio[1], Alfredo De Santis[1], and Francesco Palmieri[2]

[1] Dipartimento di Informatica, Università di Salerno
Via Ponte don Melillo, I-84084, Fisciano (SA), Italy
arccas@dia.unisa.it, cdambrosio@unisa.it, ads@dia.unisa.it

[2] Dipartimento di Ingegneria Industriale e dell'Informazione
Seconda Università di Napoli
Via Roma 29, Aversa (CE), I-81031, Italy
fpalmier@unina.it

Abstract. Internet-based video surveillance systems are now widespread in the modern e-Society, since they can be used to manage multiple physical security problems in a lot of contexts. Moreover, the growing diffusion of portable device, along with the necessity of keeping specific environments and motion events under control, brought out the need for more flexible and proactive systems, which allow the management of such scenarios. However, most of the state of the art video surveillance systems, are known to be unscalable, unreliable, insecure, and do not provide adequate guarantees for user awareness when a determined situation of interest occurs. Furthermore, almost all the currently defined systems, lack in operation flexibility: they are designed for a specific context and can not be easily adapted to other ones.

In this work, we propose a general purpose video surveillance system, which is fully distributed and accessible through ubiquitous portable devices. Such system, whose architecture is based on a self-organizing overlay network built on top of a mixture of already existing physical network connections, provides an high degree of reliability for the interactions among all its components, and ensures to its users, regardless of where they are located, the ability to receive notifications upon the occurrence of interesting events.

Keywords: P2P Surveillance, Ubiquitous User Awareness, Secure P2P Systems, Mission Critical Systems, Portable Surveillance Monitor, Kademlia

⋆ Corresponding author: Arcangelo Castiglione, Dipartimento di Informatica, Università di Salerno Via Ponte don Melillo, I-84084, Fisciano (SA), Italy, Italy, email: arccas@dia.unisa.it.

1 Introduction

The ever-growing need for security in the modern society, led to an increasing demand for surveillance activities in many areas, which may include transportation applications, monitoring of public places, remote surveillance of human activities, monitoring for quality control in industrial processes, remote surveillance in forensic applications and military sites [1]. The new generation video surveillance systems, are concerned with the monitoring of permanent and transients objects within a given area or environment, both indoor and outdoor [2], [3], [4], [5] and typically rely on *Computer Vision* techniques [6], [7], [8] [9]. By using such techniques, our system is able to automatically interpret the scene, as well as to understand and predict actions and interactions taking place among the observed objects, based on the information acquired by the involved observation camera(s).

Depending on scenarios in which they operate, that are often mission-critical and require real-time response, such systems must provide, fully or partially, the following basic features: availability, reliability, scalability and security [10]. At the state of the art, many network-based video surveillance solutions have been proposed, each one with its own specific characteristics, strengths and weaknesses. However, most of the widely known systems, are usually designed for a specific context, and can not be easily adapted to the others. Nowadays, those systems are typically structured according to the traditional client-server paradigm [11], [12], [13], [14], [15], [16] or are based on complex overlay communication [17], [18] and middleware [19], [10] architectures or, even when claim to be structured according to a resilient and robust *Peer-to-Peer* (*P2P*) [20] scheme, this is only partially true, because their operations still rely on the presence of a centralized directory service, that can be easily identified as the system's security and performance bottleneck.

Hence, it is easy to note that all the currently available systems, do not provide proper guarantees of availability, scalability and reliability that are, however, the fundamental requirements in a modern and really effective surveillance solution. Moreover, we point out that even if the surveillance systems are mainly used to monitor and improve the security in certain specific scenarios and environments, none of them addresses the problems of authentication and privacy among the involved parties, as well as the surveillance data integrity ones, except the one presented in [21], [22], which partially addresses the privacy issue among the interacting entities. Furthermore, no distributed video surveillance solution has yet been proposed, allowing a portable device to be efficiently and securely notified, in an ubiquitous manner, about the occurrence of certain interesting situations.

In order to cope with all the above issues, we propose a fully distributed flexible and adaptive video surveillance system, composed by a set of interacting peer nodes within a self-organizing overlay network, each connected with one or more camera(s). To ensure the needed robustness and scalability guarantees, such system is structured according to a completely decentralized and distributed model, based on the use of a P2P implicit communication architecture among

its components. In particular, it is based on Kademlia, a *Distributed Hash Table* (*DHT*) management facility for pure P2P organizations, used by many systems such as *Kad*, *Overnet*, *BitTorrent* and *Gnutella*. The system we proposed can be accessed in an ubiquitous manner through the use of any portable device (e.g., smartphones, tablets, laptops, etc.), even when it has limited hardware features. Moreover, it can be instructed to autonomously detect, recognize and classify certain situations of interest that may occur in monitored environments, by using sophisticated Computer Vision techniques. Given the scenarios where our system is able to operate effectively, which can be mission critical, highly risky, and also prone to attack given the potentially sensitive information that it has to manage, we paid particular attention to confidentiality and integrity of the involved surveillance data. In particular, authentication, confidentiality, integrity and non-repudiation are fully guaranteed for all the interactions and data exchange operations that take place among the system components. Ubiquitous surveillance capabilities are granted to any authorized mobile user, who, by using its own remote network access facilities, can arbitrarily choose to monitor his places of interest from everywhere and at any time. Furthermore, when there is a situation of potential interest and the mobile user is not connected to the system, he can be notified in real-time about such events through SMS and e-mail, in order to take the appropriate actions as quickly as possible. We engineered a simple proof of concept prototype of this system in order to evaluate its performance in terms of scalability, reliability, fault tolerance and security. In particular, we simulated the use of the proposed system in a WAN, and the results of the tests we performed, shown that it guarantees good performance with respect to objectives we set out above.

The remainder of the paper is organized as follows: Section 2 provides a description of the basic prerequisite concepts needed to better present the proposed solution. Section 3 gives a general system overview from the architectural point of view together with some implementation details. Section 4 describes the module responsible for the user situation awareness. Section 5 highlights all the security aspects of the system while Section 6 describes the proof-of-concept implementations and the functional tests performed on it. Finally, Section 7 shows some possible future extensions and draws the conclusions.

2 Background

2.1 Peer to Peer Overlay Organizations

A P2P overlay is a flexible virtual organization of logical associations between peer entities that is dynamically built and managed on top of existing network connections. The fundamental features of such organization is the ability of each participating entity of searching within the organization for some specific key or attribute and finding all the other networked entities within the overlay organization that are associated to that key/attribute in a very effective way, independently from its physical location and network dependent information.

Simply stated, to search a node(s), characterized by some specific attributes a querier does not need to know the IP address of the involved entities, but only attributes characterizing it. Moreover, such organizations are self-organizing, that is, participating peer nodes may dynamically join and leave the overlay in a seamless way without requiring complex reconfiguration operations or affecting the behavior and operations of other nodes in a significant way. From a performance perspective, modern structured P2P overlays support the localization of any resource/peer in a bounded time that scales with the total number of nodes n in the overlay as $O(log(n))$.

2.2 The Kademlia Overlay DHT System

Kademlia is a DHT management infrastructure for decentralized P2P networked systems [23] where each peer component is identified by a unique n-bits identifier (node ID), usually determined by using an hash function on its IP address. Basing its decisions on these identifiers the Kademlia P2P algorithm determines where to store information, and which peers are going to be responsible for it, according to a fully distributed hash table scheme. The distance between two peers is computed as the exclusive OR (XOR) of two node IDs and taking the result as an integer number. This ensures, due to the symmetric nature of the XOR operation, also the symmetry of the associated overlay structure. Each node stores contact information about the other ones in a properly crafted "*routing table*" needed to ensure the mutual reachability among nodes. Nodes are logically managed as leaves into a binary search tree where the position of each node is determined by the shortest unique prefix of its ID. In order to face the problem of stale contacts due to *churn* (departure of peers) [24], Kademlia uses redundancy, i.e., the routing table stores more than one contact (typically k) for a given distance. Every node keeps a list of: IP address, UDP port and node ID, for nodes of distance between 2^i and 2^{i+1} from itself, with $0 \leq i \leq n$, where n is the number of bits in the node ID. These lists, called k-buckets, have at most k elements. For example, in a network with $k = 20$, each node will have lists containing up to 20 nodes for a particular bit (a particular distance from itself). k-buckets are kept sorted by the time at which the associated contacts were last seen. The routing table is organized as a binary tree whose leaves are k-buckets. Thus, each lookup step has a choice of k different contacts for the next step. When a k-bucket is full and a new node is discovered for that k-bucket, the least recently seen node in the k-bucket is probed through a PING operation. If the node is found to be still alive, the new node is placed in a secondary list, called *replacement cache*, which is used only if a node in the k-bucket stops responding. In other words, new nodes are used only when older ones disappear. Kademlia has four messages, corresponding to remote procedure calls:

- *PING:* probes a peer to check if it is active.
- *STORE:* instructs a peer to store a {*key*, *value*} pair for later retrieval.
- *FIND_NODE:* takes an ID, and returns {*IP address*, *UDP port*, *NodeID*} triples for the k peers it knows that are closest to the target ID.

– *FIND_VALUE:* is similar to FIND_NODE, it returns {*IP address*, *UDP port*, *NodeID*} triples, except for the case when a peer received a STORE for the key, it just return the stored value.

A node which would like to join the network must first performs a bootstrap process with another node that is already participating in the Kademlia network. The joining node inserts the bootstrap node into one of its k-buckets and then does a FIND_NODE of its own ID against the bootstrap node. The "*self-lookup*" will populate other nodes' k-buckets with the new node ID, and the joining node's k-buckets with nodes in the path among it and the bootstra node. Afterwards, the joining node refreshes all the k-buckets further away than the k-bucket the bootstrap node falls in. Kademlia uses *iterative routing*, where the client is responsible for the entire lookup process. At each step, the client sends a lookup request to the next-hop peer and waits for a lookup reply. The reply lets the client know what the next hop is. In Kademlia, a peer must locate the k closest peers to some given node ID. This lookup initiator starts by picking α peers (*parallel routing*) from its closest non-empty k-bucket [25], and then sends parallel asynchronous FIND_NODE messages to the α peers it has chosen. If a FIND_NODE operation fails to return a peer that is closer than the closest peers already seen, the initiator resends the FIND_NODE to any of the k closest peers that has not been already queried.

In supporting key-based searches on its own DHT infrastructure, Kademlia does not introduce periodic overhead, but exploits the previous search transactions to stabilize the overlay network connections.

3 The System Overview

The proposed surveillance system is composed by three logical entities: the surveillance node, the portable device needing to access the surveillance data and the P2P overlay communication and search infrastructure, used to implements a fully distributed index for the rapid localization of surveillance nodes associated to specific monitoring environments. The overall system architecture is sketched in Figure 1.

3.1 Basic architectural choices

The system is able to support a large number of video monitoring stations, located in different physical places, each one with its own access privileges and private views. These stations are controlled by surveillance nodes, reachable though the Internet or a mix or public and private networks and typically run on generic *Commercial Off-The-Shelf* (*COTS*) workstation hardware. Each surveillance node, associated to one or more camera(s), is autonomously responsible for all the interactions among controlled cameras and portable devices, involved in the monitoring of certain areas or environments, in order to detect situations of interest.

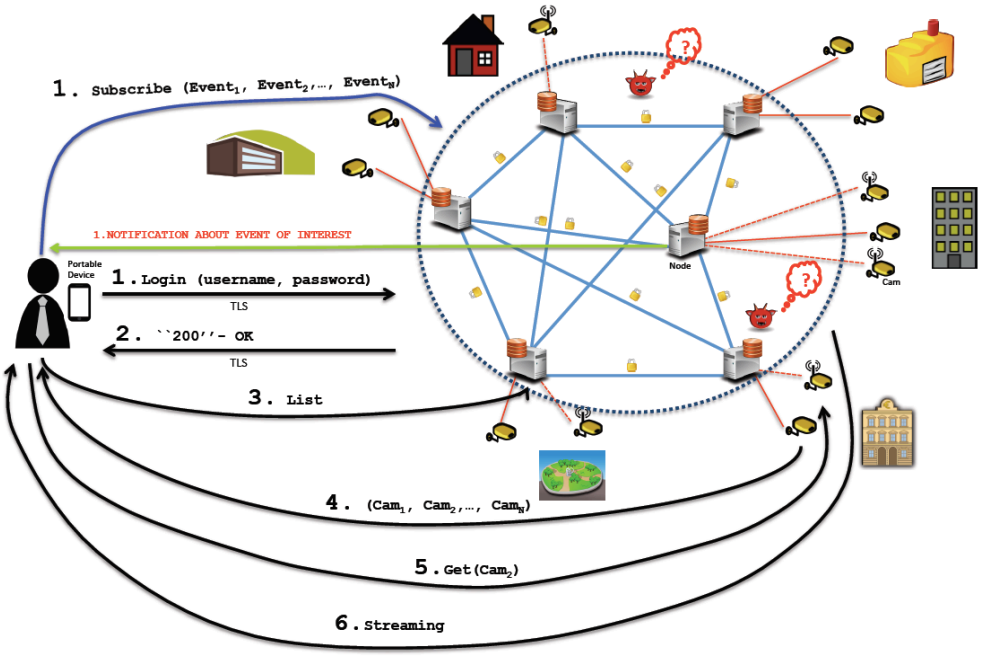

Fig. 1. The overall system architecture and the components interaction

The surveillance node, also deals with the acquisition of relevant information from camera(s), along with the management of the whole Computer Vision process. These activities are managed by using the user situation awareness module, which issues real-time notifications when a situation of interest subscribed by a portable device occurs.

The portable device, instead represents the system interface for the mobile user, which is interested in monitoring a specific area or environment and wants to have an awareness about the occurrence of specific situations or events.

The P2P overlay communication and search infrastructure is based on Kademlia [23], which, as we have previously seen, is a DHT for decentralized P2P overlay organizations. We made that choice since Kademlia, among the existing similar systems, is the one which minimizes the number of messages sent by each node in order to acquire information about the registered surveillance services and their associated nodes.

We model our system in a fully distributed way, where the whole architecture is organized as a P2P network of nodes, which may be geographically located everywhere on the Internet or connected to any combination of local and wide area IP-based communication infrastructures, and may use completely different kinds of networking and communication technologies, as shown in Figure 1.

We have chosen this architectural scheme since among the similar available, it is the one which best supports our needs of reliability, scalability and availability,

by exploiting the self-organization capabilities of the DHT overlay communication infrastructure. In fact, in traditional systems, the surveillance stations are accessible by their users only through a centralized brokering services, imposing severe limits on the scalability and reliability of the whole solution. That is, the system performance, in presence of n client and server nodes, has to scale with $O(n^2)$ due to the $n \cdot (n-1)$ potential relationships among the n involved entities, and this is clearly not acceptable for very large n, for example in systems with many monitoring sites and an huge number of users, such as the publicly accessible ones, usually deployed in the tourism and travel sectors. In other words, the number of hosts that can be monitored at a given time, is limited by the bandwidth and the processing power of the central brokering system, and hence the solution does not scale. Furthermore, once the single central brokering unit fails, the whole system will lose its functionality and the entire set of controlled places will be without any kind of surveillance. Simply stated, any centralized brokering service becomes a single point of failure.

On the contrary, the proposed solution avoids the necessity of any centralized directory or brokering services providing access to surveillance nodes and their managed resources. In particular, by eliminating single points of failure and performance bottlenecks, such solution provides our system with the ability of allowing quick topology changes and easily grow/shrink by adding or removing new nodes or portable devices according to a plug-and-play paradigm, without complex configuration and management tasks. In this way, it is easier to ensure an high degree of scalability and flexibility, and the overall architecture is able to survive to failures or disruptions of any of its components, without stopping its global operations, so that any kind of damage only affects the locally involved nodes.

Moreover, in presence of a very large number of nodes scattered throughout the Internet, the Kademlia-based solution allows the delivery of multiple parallel queries for the same key to different peers. In this way, any delay or timeout on a specific route to destination do not necessarily affect the search process, thus ensuring faster and more reliable searches, also in presence of a large number of nodes continuously joining and leaving the overlay network.

3.2 Implementation Details

The system, is accessible to both the surveillance nodes and mobile monitoring devices through a publicly available hostname, registered on the *Internet Domain Name System* (*DNS*). Such an hostname, is dynamically mapped to one of the nodes in the P2P network, through *Dynamic DNS* [26] techniques. A new node or portable device who want to join our system, must only know such hostname, along with the proper access credentials (username and password, or digital X.509 certificate). The above hostname, must be also used as the Kademlia bootstrap node for all the entities (surveillance nodes and portable devices devices) that need to access the overlay network infrastructure. Surveillance nodes, when joining such network, register their monitoring capabilities, in terms of associated camera(s)/monitored environment(s), by storing on the Kademlia

DHT each environment identifier (*key*), together with the serving node IP address (*value*). Portable devices, which want to join the surveillance system, can search the Kademlia overlay for keys corresponding to the environments of interest under monitoring, by obtaining the IP addresses of associated surveillance nodes, in order to connect to them for visualizing the cameras' video materials.

Moreover, a portable device that is not currently joined, can be asynchronously notified about the occurrence of events of interest (motion detection occurrences etc.). It is important to point out that there is no explicit communication among surveillance nodes, which only interact among themselves indirectly, by exposing and sharing their service information (monitored environments) through the overlay DHT facilities. Instead, the communication among portable devices and surveillance nodes takes place according to the traditional client-server paradigm, where each surveillance node assumes the role of server for all the mobile nodes' queries. In particular, the interaction between such two parties, is carried out via (secure) *TCP* socket, by using an ad-hoc *FTP-like* protocol, for the delivery of control messages (queries and results) and video surveillance data. Like in *passive-mode FTP*, such ad-hoc protocol (as defined in [21]) in order to overcome limitations introduced by firewalls or NATs, as well as restrictions and policies imposed by cellular operators, forces the portable device to open two communication channels, one for the sending of control messages and the other for data transfer, by thus avoiding the opening of any connection backwards, from the surveillance node to the mobile one. The messages used for the interaction among a node and a portable device are four: *Login*, *List*, *MGet* and *Subscribe*. The *Login* message is used by the portable device for communicating to the node its login credentials, i.e., username, password and optionally its own certificate, in the case in which strong mutual authentication among endpoints is needed. The authentication phase, accomplished through the use of such message, can be successful or not. If it has been successfully completed, the node sends to portable device a message which contains the "200" return code, as in the standard *FTP*, and the interaction between these two parts continues normally. In the case of failure, the portable device is notified about that by the node, and it is shown an alert message on its display. Upon a successful authentication, a portable device can use the *List* message for requesting a preview (typically in *JPEG*) of each of the environments monitored by the node to which it is connected. As soon as a node receives such command, it takes a snapshot from each of its camera(s), and sends such snapshots to the portable device. When a portable device, based on the snapshots it has received, chooses the particular environment which intends to monitor (identified by an univocal code), at this point it can use the *MGet* command. Once a node receives this command, it creates a data channel with the portable device, and through this mechanism it sends a video streaming to the latter. It is important to point out that a portable device, after its successful connection to a surveillance node, is able to communicate the possible events in which it is interested and for which it wants to receive notifications, by using the *Subscribe* message.

4 The User Awareness Module

If an event of interest for a particular place occurs when the user is not connected to the system, such user has no awareness about what has happened, and therefore has no information about that. Hence, we decided to provide our system with an asynchronous notification facility, that enables the user to have the full control on what happens, even when it is not connected. For this reason, our system allows the user to specify places and scenarios which he/she intends to monitor, as well as the events of interest.

When an event in a place of interest occurs, the user is notified as quickly and reliably as possible about that, in order to appropriately manage such a situation. The notification, must reach the user in an ubiquitous way. For this reason, we think that *GSM* network coverage is a fairly realistic assumption, so our system sends to the user a notification message through the SMS system. However, in some particular circumstances, the user may not be connected to the cellular network and may only use a local data network (such as W-Fi). Therefore, our system uses as a notification method, at the same time, the one based on SMS and the other based on e-mail. The notification message, includes all the information defining the event, along with any other useful thing to remedy it in the most appropriate and quick way. In order to guarantee the user awareness about the monitored scenarios, we enable our system to semantically interpret detected objects behavior.

The system we propose is autonomously able to identify and learn from events and occurring interactions, that take place in a given monitored environment [1]. In particular, we provide our system with a component dealing with the so-called *Computer Vision*, which allows video processing, real time scene recognition with related data analysis and decision making with respect to them [6], [7], [8], [9], [11]. We implemented the Computer Vision module through the use of the *Open Source Computer Vision* (*OpenCV*) library [27], [28]. OpenCV is a library of programming functions, mainly aimed at real-time Computer Vision and is released under the BSD license. The Computer Vision, can be considered as a process constituted by a number of phases that may vary, depending on the operating scenario and the specific system application domain. Such phases, can be typically grouped into four main blocks, which are *image preprocessing*, *object recognition* with *motion detection*, *object monitoring* and *reasoning* with *activity recognition* [29], [5], [4], [2].

In the preprocessing phase, the image sequence produced by one or more camera(s), is processed by our Computer Vision module in order to ensure resampling, noise reduction, contrast enhancement and scale space representation. The recognition phase, instead, finds an object within an image or video sequence, also when such object is partially obstructed from view. In particular, in such phase, image features as lines, edges and ridges, along with any other localized points at various levels of complexity, are extracted in order to obtain the segmentation of one or multiple image regions which contain a specific object of interest. The motion detection phase, detects a change in position of an object relative to its surroundings or the change in the surroundings relative to

an object. Object monitoring (or tracking) instead locates a moving object (or multiple objects) over time by using one or more camera(s). The main aim of such phase is to associate target objects in consecutive video frames. It is important to point out that the association can be especially difficult when the objects are moving fast relative to the frame rate, or when the tracked object changes orientation over time. In order to manage such situations, our system employs a motion model which describes how the image of the target might change for different possible motions of the object. The last Computer Vision phase implemented by our system is the activity recognition, that is, the process of recognize actions and goals of one or more actors from a series of observations on their actions and environmental conditions. After such phase, if a potential situation of interest for the mobile user is detected, then the system triggers the notification module which takes care of notifying users about the events of interest. Such a component, dealing with the user notification, is composed by two modules, the former takes secure and efficient delivery of SMS messages, while the latter takes care of sending e-mail messages. It is easy to note that the portable device, when it is joining the system, it must provide one or more e-mail addresses, along with one or more telephone numbers, by which it intends to be notified. In Figure 2, we show the various components which constitute the user situation awareness module, along with its flow of events. Video streaming, acquired from different camera(s) connected to each node of our system, is continuously analyzed by the corresponding module responsible for the Computer Vision process. As soon as such module detects, based on how it has been trained by, the occurrence of a certain situation of interest, it provides the user with an overview of the situation that has arisen, through an appropriate alert message, which is sent either by SMS and through e-mail.

5 The Security Services

It can be easily observed that due to its mission critical nature, any surveillance system is particularly exposed to security problems [30], especially because of the scenarios where it has to operate. In detail, such type of system, may be subjected to several types of threats, such as eavesdropping, data modification, IP address spoofing, *Denial-of-Service* (*DoS*) [31] and *man-in-the-middle* attacks [32], [33], [34]. However, the absence of single point of failure, and hence of elements which may become an easy target for DoS attack, makes the proposed solution sufficiently robust, because of Kademlia, due to its decentralized architecture that is the only critical component for the overall system operations, is resistant against most of the known DoS attacks.

Moreover, in relation to the environments in which it operates, our system may be particularly vulnerable to *compromised-key attacks*, carried out by using social engineering techniques. In order to avoid, or at least to limit such kind of attacks, we must take into account the interaction among all the system parties, in particular we consider the one among the various nodes and the other between a node and a portable device.

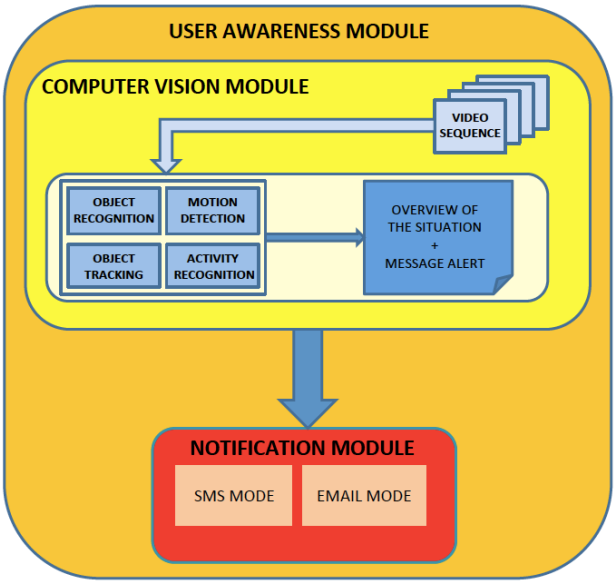

Fig. 2. The user awareness module

In order to ensure security during the access to the Kademlia overlay information exchange infrastructure, we have chosen to use cryptographic functions provided by *maidsafe-dht* library [35]. Such library, introduces a strong encryption layer to ensure secure operations within the DHT overlay. It also ensures to our system *NAT* traversal capabilities, TCP emulation for fault tolerance, routing of queries through low-latency paths as well as use of asynchronous and parallel queries to avoid timeout delays from failed nodes. Furthermore, maidsafe-dht also includes some significant enhancements to the traditional Kademlia implementation, by providing a downlist modification with notification of dead nodes in searches as well as forcing partner bucket to contain the most recent closest nodes, in order to further increase the reliability of the whole DHT system.

Instead, regarding the interaction among a node and a portable device, our system uses the *Transport Layer Security* (*TLS*) protocol, which ensures security and privacy for stream-oriented communications. We also paid particular attention to security concerning user notifications about situations of interest.

In particular, security properties of component which deals with SMS based notification, rely on the *SEESMS* architecture (*Secure Extensible and Efficient SMS*) [36], initially presented in [37].

SEESMS is a framework for the exchange of secure SMS, which aims to be efficient through the support of several cryptosystems by using a modular architecture, as it is shown in Figure 3. In particular, it is important to point out that the SMS notification module of our system is implemented by using the Secure SMS Management Center of SEESMS, while the SEESMS client is included

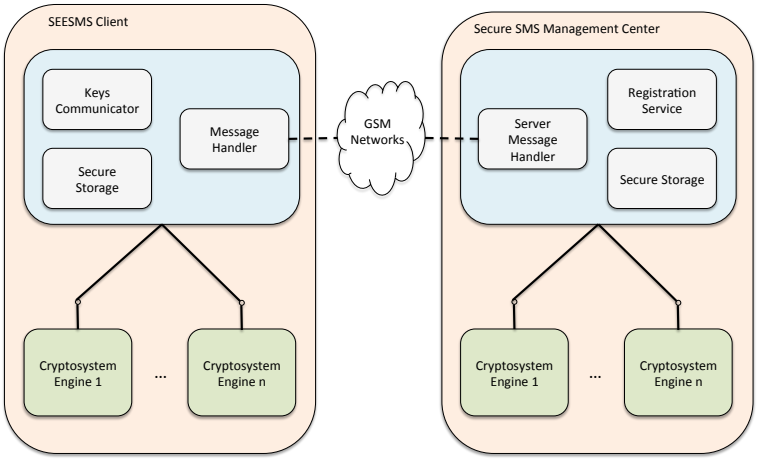

Fig. 3. The SEESMS architecture

within the portable device. Such framework, represents a tool which uses an SMS based communication mechanism to exchange encrypted, non-repudiable and tamper-proof messages. One of the main advantage of SEESMS over similar systems, is the possibility to choose which combination of cryptosystem/security parameters to use during message exchange. Moreover, one of the two parties (node or portable device), could set a minimum security level to be fulfilled during the communication, giving the other peer the possibility to increase (but not decrease) it. From an architectural point of view, as can be seen from Figure 3, the flexibility of SEESMS has been made possible by the adoption of a modular architecture, where the cryptographic functions of the framework are not built into SEESMS, but are delegated to some external pluggable modules.

On the other hand, with respect to the security of e-mail based notification, we chose to rely on the *Secure/Multipurpose Internet Mail Extensions* (*S/MIME*) standard [38], in order to use X.509 certificate for signing and encrypting each e-mail notification message sent. By doing this, we intend to guarantee identification (authentication), confidentiality, integrity and non-repudiation of all the notification messages sent, in order to avoid fake alerts to be maliciously sent to the monitoring users.

6 Proof of concept and functional evaluation

We engineered a very simple proof-of-concept prototype of our system in order to validate its functional behavior and test the effectiveness of the aforemen-

tioned surveillance architecture, with an emphasis on the use of currently available COTS devices and open-source components. The testing was carried out on three surveillance nodes connected to the network in a stable way, along with three other ones which dynamically connect and disconnect from it. Each node used for the functional testing operations, consists of conventional PCs with different hardware characteristics, each one controlling a single camera, interconnected through a Local Area Network (LAN). We also used several common portable devices (smartphones and tablets), connected to the network in different ways, ranging from *Wi-Fi* LAN connections to *3G/UMTS* ones, provided by traditional cellular Internet service providers.

In general, our preliminary functional tests, shown that, by dynamically varying the number of its parties as well as the amount of data exchanged, our system continues to behave correctly and is essentially not affected from the above events, both in terms of performance and efficiency. We point out that, due to the self-organizing nature of its basic association mechanism, the system tends to be highly reliable. The resources exposed from surveillance nodes dynamically joining the P2P network, become correctly available to portable devices almost immediately after the successful connection and registration of the corresponding information. Search operations on the overlay DHT by the mobile nodes, are carried out instantaneously and are resilient to multiple node failures until other nodes are able to respond about a specific key. However, this feature can be useful only in presence of different surveillance nodes controlling the same camera(s), according to an architectural scheme which introduces redundancy at the surveillance node level, since in presence of a single node controlling a set of camera(s), its failure implicitly isolates all the associated devices.

The portable devices we used, including those with limited hardware features, showed for access, monitoring and control of recorded data, a response time in the order of a few milliseconds. Also the notification task, appears to be very lightweight and well tolerated by portable devices. For measuring the response time which concerns detection and notification of interesting events, we configured the system in order to make it able to recognize and notify the user about the occurrence of common situations, such as the entrance of people into a given environment or the move of an object. We also empirically evaluated the time elapsed since the detection of an event by a node, until the receipt of a notification by the portable device. Such time, may vary from one to ten minutes, depending on the complexity associated to the detection and understanding of the occurred event. However, the asynchronous notification task, affects that time only by a negligible factor. The security of our system, was assessed by subjecting it to several attacks, carried out through exploits and tools for sniffing (e.g., Wireshark) and man-in-the-middle (e.g., Ethercap) attacks. In addition, we also used a vulnerability scanner (Nessus) to asses the whole testing network where the surveillance nodes have been located. The system has been found to be sufficiently secure, with respect to the analysis carried out by using such tools.

7 Conclusions and Future Work

The system we propose, ensures to the mobile user a complete awareness about the scenarios under consideration, over the whole P2P-based surveillance organization. The user awareness module, combines together video analysis, intelligence and ability to cope with real-time events of interest.

Our system, guarantees to mobile users, real-time monitoring of scenarios and notification about relevant events as soon as they occur, always by paying particular attention to all the security issues that may arise. In the future, we intend to provide our system with a Web-based interface, which enables the uniform access to its services, thus avoiding the use of a specific client for that purpose. Furthermore, in order to improve the efficiency of data exchange, we plan to use data compression techniques over the communication channel among the portable device and the surveillance node.

We also plan to take advantage of new features provided by the "*Smart Cameras*", and in general by the "*Embedded Smart Devices*", especially to alleviate the computational load that each node must handle, considering the number of operations it performs. Our further aim, is also to improve the system security, by involving biometric techniques, smart cards and trusted hardware modules, in order to prevent compromised-key attack and threats from insider, due to social engineering techniques. Moreover, we intend to store some interesting data acquired, and to protect them by using an *Attribute-Based Encryption Scheme*, which permits the fine-grained access control over them [39]. We think that, it would be particularly interesting to provide each node also with different types of sensor(s), such as detectors for smells (to prevent gas leaks), vibrations (for earthquakes) and noises (to help the system in low light conditions).

Finally, it may be useful to use optimization algorithms, for the exact positioning of camera(s) and sensor(s), according to the specific monitored place. By adding such new features, we intend to create an even more intelligent system, which enables the user to have an additional support, concerning not only the monitoring, but also the deployment of camera(s) and sensor(s).

References

1. Valera, M., Velastin, S.: Intelligent distributed surveillance systems: a review. In: Vision, Image and Signal Processing, IEE Proceedings-. Volume 152/2., IET (2005) 192–204
2. Hu, W., Tan, T., Wang, L., Maybank, S.: A survey on visual surveillance of object motion and behaviors. Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on **34**(3) (2004) 334–352
3. Hampapur, A., Brown, L., Connell, J., Ekin, A., Haas, N., Lu, M., Merkl, H., Pankanti, S.: Smart video surveillance: exploring the concept of multiscale spatiotemporal tracking. Signal Processing Magazine, IEEE **22**(2) (2005) 38–51
4. Buxton, H., Gong, S.: Visual surveillance in a dynamic and uncertain world. Artificial Intelligence **78**(1) (1995) 431–459

5. Cucchiara, R., Grana, C., Piccardi, M., Prati, A.: Detecting moving objects, ghosts, and shadows in video streams. Pattern Analysis and Machine Intelligence, IEEE Transactions on **25**(10) (2003) 1337–1342
6. Hartley, R., Zisserman, A.: Multiple view geometry in computer vision. Volume 2. Cambridge University Press (2000)
7. Forsyth, D.A., Ponce, J.: Computer vision: a modern approach. Prentice Hall Professional Technical Reference (2002)
8. Parker, J.R.: Algorithms for image processing and computer vision. Wiley Publishing (2010)
9. Moeslund, T.B., Granum, E.: A survey of computer vision-based human motion capture. Computer Vision and Image Understanding **81**(3) (2001) 231–268
10. Kornecki, A.: Middleware for distributed video surveillance. Distributed Systems Online, IEEE **9**(2) (2008) 1–1
11. Cucchiara, R., Grana, C., Prati, A., Vezzani, R.: Computer vision techniques for PDA accessibility of in-house video surveillance. In: First ACM SIGMM international workshop on Video surveillance. IWVS '03, New York, NY, USA, ACM (2003) 87–97
12. Javed, O., Rasheed, Z., Alatas, O., Shah, M.: KNIGHT trade;: a real time surveillance system for multiple and non-overlapping cameras. In: Multimedia and Expo, 2003. ICME '03. Proceedings. 2003 International Conference on. Volume 1. (july 2003) I – 649–52 vol.1
13. Comaniciu, D., Berton, F., Ramesh, V.: Adaptive Resolution System for Distributed Surveillance. Real-Time Imaging **8**(5) (2002) 427 – 437
14. Cucchiara, R., Prati, A., Vezzani, R.: A multi-camera vision system for fall detection and alarm generation. Expert Systems **24**(5) (2007) 334–345
15. Ostheimer, D., Lemay, S., Ghazal, M., Mayisela, D., Amer, A., Dagba, P.F.: A modular distributed video surveillance system over IP. In: Electrical and Computer Engineering, 2006. CCECE'06. Canadian Conference on, IEEE (2006) 518–521
16. Castiglione, A., Cepparulo, M., De Santis, A., Palmieri, F.: Towards a Lawfully Secure and Privacy Preserving Video Surveillance System. In Buccafurri, F., Semeraro, G., eds.: E-Commerce and Web Technologies. Volume 61 of Lecture Notes in Business Information Processing. Springer Berlin Heidelberg (2010) 73–84
17. Yuan, X., Sun, Z., Varol, Y., Bebis, G.: A distributed visual surveillance system. In: Proceedings. IEEE Conference on Advanced Video and Signal Based Surveillance, 2003., IEEE (2003) 199–204
18. Dias, H., Rocha, J., Silva, P., Leao, C., Reis, L.P.: Distributed surveillance system. In: Artificial intelligence, 2005. epia 2005. portuguese conference on, IEEE (2005) 257–261
19. Desurmont, X., Bastide, A., Czyz, J., Parisot, C., Delaigle, J.F., Macq, B.: A general purpose system for distributed surveillance and communication. Intelligent Distributed Video Surveillance Systems (2006) 121–156
20. Wu, Y.S., Chang, Y.S., Juang, T.Y., Yen, J.S.: An Architecture for Video Surveillance Service Based on P2P and Cloud Computing. In: Ubiquitous Intelligence & Computing and 9th International Conference on Autonomic & Trusted Computing (UIC/ATC), 2012 9th International Conference on, IEEE (2012) 661–666
21. Albano, P., Bruno, A., Carpentieri, B., Castiglione, A., Castiglione, A., Palmieri, F., Pizzolante, R., You, I.: A Secure Distributed Video Surveillance System Based on Portable Devices. In: CD-ARES. (2012) 403–415
22. Albano, P., Bruno, A., Carpentieri, B., Castiglione, A., Castiglione, A., Palmieri, F., Pizzolante, R., Yim, K., You, I.: Secure and distributed video surveillance

via portable devices. Journal of Ambient Intelligence and Humanized Computing (2013) 1–9
23. Maymounkov, P., Mazieres, D.: Kademlia: A peer-to-peer information system based on the XOR metric. Peer-to-Peer Systems (2002) 53–65
24. Rhea, S.C., Geels, D., Roscoe, T., Kubiatowicz, J.: Handling churn in a DHT. Computer Science Division, University of California (2003)
25. Stutzbach, D., Rejaie, R.: Improving lookup performance over a widely-deployed DHT. In: Proc. Infocom. Volume 6. (2006)
26. Vixie, P., Thomson, S., Rekhter, Y., Bound, J.: Dynamic Updates in the Domain Name System (DNS UPDATE). RFC 2136 (Proposed Standard) (April 1997) Updated by RFCs 3007, 4035, 4033, 4034.
27. Bradski, G.: The OpenCV library. Doctor Dobbs Journal **25**(11) (2000) 120–126
28. Bradski, G., Kaehler, A.: Learning OpenCV: Computer vision with the OpenCV library. O'Reilly Media, Incorporated (2008)
29. Wijnhoven, R., Jaspers, E., et al.: Flexible surveillance system architecture for prototyping video content analysis algorithms. In: Electronic Imaging 2006, International Society for Optics and Photonics (2006) 60730R–60730R
30. Castiglione, A., De Prisco, R., De Santis, A.: Do You Trust Your Phone? In Noia, T., Buccafurri, F., eds.: E-Commerce and Web Technologies. Volume 5692 of Lecture Notes in Computer Science. Springer Berlin Heidelberg (2009) 50–61
31. Naoumov, N., Ross, K.: Exploiting P2P systems for DDoS attacks. In: Proceedings of the 1st international conference on Scalable information systems, ACM (2006) 47
32. Wallach, D.S.: A survey of peer-to-peer security issues. In: Software Security - Theories and Systems. Springer (2003) 42–57
33. Urdaneta, G., Pierre, G., Steen, M.V.: A survey of DHT security techniques. ACM Computing Surveys (CSUR) **43**(2) (2011) 8
34. Wang, P., Tyra, J., Chan-Tin, E., Malchow, T., Kune, D.F., Hopper, N., Kim, Y.: Attacking the KAD network. In: Proceedings of the 4th international conference on Security and privacy in communication netowrks, ACM (2008) 23
35. Irvine, D.: Kademlia DHT with NAT traversal. `http://code.google.com/p/maidsafe-dht/` (2013) [Online; accessed 16-June-2013].
36. Castiglione, A., Cattaneo, G., Cembalo, M., De Santis, A., Faruolo, P., Petagna, F., Ferraro Petrillo, U.: Engineering a secure mobile messaging framework. Computers & Security **31**(6) (2012) 771–781
37. De Santis, A., Castiglione, A., Cattaneo, G., Cembalo, M., Petagna, F., Ferraro Petrillo, U.: An Extensible Framework for Efficient Secure SMS. 2010 International Conference on Complex, Intelligent and Software Intensive Systems **0** (2010) 843–850
38. Ramsdell, B.: S/MIME version 3 message specification (1999)
39. Goyal, V., Pandey, O., Sahai, A., Waters, B.: Attribute-based encryption for fine-grained access control of encrypted data. In: Proceedings of the 13th ACM conference on Computer and communications security, ACM (2006) 89–98