CoDE - An Application-Layer Framework for Confidentiality in Distributed Environments

Jean Botev
University of Luxembourg
Email: jean.botev@uni.lu

Marco Milanesio
Institut Eurecom
Email: marco.milanesio@eurecom.fr

***Abstract*—A new generation of networked applications and social utilities, originating inter alia from the ongoing trend towards high-performance mobile devices, tightens exigencies relative to the processing of sensitive data. However, in order to ensure their correct operation, many of these applications require such information to be exchanged, and thus need to implement a differentiated confidentiality concept in this regard.**
This paper introduces *CoDE*, a unified framework defining a set of collaboratively enforced confidentiality levels through an application-layer protocol which covers a wide range of distributed scenarios with a particular focus on decentralized operation, efficiency and simplicity.
First evaluations show promising results with regard to a variety of network types involving different transmission ranges, packet loss rates or noise, along with a high churn degree.

***Index Terms*—MANET, confidentiality, application-layer protocol, collaboration, social utility**

I. Introduction

The ongoing trend towards high-performance mobile, interconnected devices like smartphones entails a surge of a new generation of networked applications and social utilities (e.g., Waze[1]). In order to operate properly and to ensure a certain quality of service, these applications generally rely on the sharing of some kind of data. It is therefore in the user's interest to partake in sharing that information and benefit from an enhanced quality of the provided functionality.
However, as for instance the problem with the collection and processing of iPhone location data by Apple Computer Inc.[2] shows, the gathering of potentially sensitive personal data has to be transparent to and not simply hidden from users. Knowledge on which data is shared or not and, most importantly, the availability of different options to control such a behavior in a clear and simple way are integral to the lasting acceptance and success of an application of the above mentioned kind.

One of the core issues therefore is to ensure *data confidentiality* in distributed scenarios whenever it is needed, and to which degree. Our main research question, thus, is *how to grant data confidentiality at an application level*. This research problem applies equally to mobile ad hoc networks and all other networked environments, from the Web to distributed virtual environments. As an example, let us look at the common case where an application comprises some collaborative filtering mechanism in order to compute a recommendation with respect to a particular domain. While certainly some users might be happy to share their data with practically everyone, particularly concerning more personal information others might want to do that with a specific class or even a handpicked set of communication partners only. It is thus straightforward that there should be different *levels* of confidentiality involved, depending on the individual requirements and under full user control. This aspect should be ideally embedded into the application layer and form an integral part of it, either for the explicit exchange of data or as a precondition for implicit data exchange as in the collaborative filtering example.

Data confidentiality is often treated as synonymous with data privacy and, as such, merely associated with the existence of an encryption scheme for securing message exchanges. Within this paper, however, we take another perspective on confidentiality, giving it a wider and more differentiated meaning by outlining a set of confidentiality levels considering also access authorization aspects. In doing so, we do not only focus on the encryption of messages, but involve a different approach in terms of an application's utilization with regard to the number and also the kind of users partaking in a distributed collaborative computation. Another aspect worth mentioning in this context is the conceptual difference regarding location privacy and anonymity, i.e., hiding *where* and *who* a user is respectively. We neither target full location privacy, nor complete anonymity as reaching these goals can be impeded within a small-enough geographic scale. Therefore, conceptually, rather the term *source* or *relation* obfuscation are the guidelines for the definition of our confidentiality model.

In this work, we introduce CoDE, a unified framework comprising simple yet effective means to enable such application-layer confidentiality in a multitude of distributed scenarios, particularly focusing on decentralized networks with inherently high dynamics.
The remainder of this paper is structured as follows: Section II defines the framework, detailing the confidentiality levels and the proposed protocols. An experimental evaluation is then presented in Section III. Before concluding, Section IV gives an overview of related work and developments. Finally, we summarize our contributions and results in Section V, also discussing shortcomings and potential future work.

[1] http://world.waze.com/ [accessed 2011-07-11]
[2] http://radar.oreilly.com/2011/04/apple-location-tracking.html [accessed 2011-07-11]

COLLABORATECOM 2011, October 15-18, Orlando, United States

DOI 10.4108/icst.collaboratecom.2011.247130

II. Framework

Our framework essentially comprises two aspects: a differentiated, user-centric notion of confidentiality, and a set of architecture-driven approaches to enforce the desired degree of confidentiality in distributed, collaborative environments.

In an information technology context, confidentiality is loosely defined according to ISO/IEC 27001 as the property that *information is not made available or disclosed to unauthorized individuals, entities, or processes*[3], and thus can be decreed in many different ways depending on specific usage scenarios.
In this section, before detailing the different architectural approaches, we first introduce and specify the intrinsic notion and different levels of confidentiality offered.

A. Confidentiality Levels

For covering different use cases and scenarios, we define three incremental levels of confidentiality, similar to the work of Clutterbuck et al. [1], where a commensurate set is proposed for determining the existence or the absence of an encryption scheme in a wireless LAN. However, as mentioned in the introduction to this paper, we give a wider and more differentiated meaning to the term confidentiality, centering on the user's perspective rather than on the network layer.
Assuming the existence of encryption facilities on both network and application layer, we define three different levels through integration of authorization class and cryptographic strength as summarized in Table I.

C-Level	Authorization	Cryptographic Strength
0	Public Access	None
1	Application-Based	Weak
2	Group-Based	Strong

TABLE I
Confidentiality Levels

The coarse granularity is intentional, as our initial and main design goal is simplicity, i.e., the provision of an easy-to-use tool set for application development.

1) Level 0 - No Confidentiality: Constituting the lowest level offered, no encryption whatsoever is used here and data are shared with any user within transmission range. Generally, this level is chosen in case of the information exchanged being perceived as uncritical and the priority lying in increasing the likeliness for a faster acquisition of aggregate information.

2) Level 1 - Weak Confidentiality: Weak encryption is applied to the data on this confidentiality level. Its focus is application-driven and constraints in terms of group formation regarding the shared data can be enforced. This level is picked if information is not particularly sensitive but relevant to - and only to - all the users running the specific top-level application.

[3]http://www.praxiom.com/iso-27001-definitions.htm [accessed 2011-07-11]

3) Level 2 - Strong Confidentiality: The highest proposed confidentiality level exploits strong encryption for securing the data. Information is classified as critical in particular from a user-specific point of view, rather than from the application side. Personal application-relevant data is preferably shared only with friends or another specific set of participating users, leading to a tighter indirect group formation and imposing stronger constraints.

In order to illustrate the utilization of the individual confidentiality levels, we briefly state a few exemplary use cases involving recommender systems:
A first scenario is a refectory to which employees of (maybe) different companies come for lunch. With a collaborative filtering application in mind, users share information on particularly recommendable dishes, and the different confidentiality levels would manifest themselves possibly as follows: (a) Level 0 - users want to exchange information with everybody in the refectory, no matter if they know them or if they are working in the same company; (b) Level 1 - information is shared only in between people of the same company; (c) Level 2 - information flows only among the members of a specific project team.
On a larger yet still restricted areal scale, some recommendation-based application might be run in a university setting. The different confidentiality levels entailing the respective encryption and constraints here might be distributed similarly among all users: (a) Level 0 - sharing of data with all people on the campus; (b) Level 1 - exchange data only with the students of the Computer Science department; (c) Level 2 - data exchange remains exclusive to the students of the Network Security class.
Naturally, depending on the definition of whom the application is allowed to share data with, these distinctions can easily be extrapolated to any scale, both in terms of user numbers and areal coverage.

B. Architectural Approaches

From an architectural point of view there are two basic ways for accessing a generic computation: centralized or decentralized. With regard to our work, a centralized approach is able to offer the highest confidentiality level available, as a server application carries all the load of providing strong encryption by exploiting, for example, a robust PKI. It will also achieve the best results from a performance point of view, as the central server takes care of both managing the application and the computation of the result for a given set of available nodes in a certain location. However, the drawbacks of centralized approaches are well known and not in the scope of this work. As a reminder, consider bottlenecks, single points of failure, pivotal and centralized management of sensitive data and so on. Albeit including it as an option, the proposed framework thus particularly aims at ensuring different levels of confidentiality without exploiting a centralized architecture. Beginning with a fully

decentralized network where each node provides the same functionalities as the others, we then refine the protocol with the variant of a hierarchical decentralized network, where a certain number of so-called *management nodes* are exploited for alleviating the overhead accompanying the provision of a preassigned confidentiality level. We compare these two approaches and evaluate them against the centralized approach, with a particular focus on intrinsic trade-offs. Among others, we look into the total overhead induced by the management messages for granting a predefined application-level confidentiality. We do not explicitly deal with the size of the respective payload as this is an application-specific feature.

For setting up our framework, we start from a mobile ad hoc scenario in which location obfuscation is not a prime concern. Following the examples, consider a restaurant, refectory, or similar area with clear spatial boundaries. In environments like these, we want to exploit the information from different people who actually *are* in the same place. For this reason, location privacy is not the goal, albeit the framework does not exclude also providing a mechanism to protect this kind of data. We aim, however, at protecting other sensitive application-relevant data, including the IP address (for Wi-Fi connections) or the Bluetooth MAC address, referring to this kind of data protection as source obfuscation. Besides Wi-Fi, we picked Bluetooth as exemplary communication technology due to its popularity and low power consumption. Alternate technologies as specified in IEEE 802.15.x with similar transmission ranges are available. Also combinations of the two technologies exist, compensating individual weaknesses [2].

Name	Description	Size
K_p^+, K_p^-	Public and private keys	128
BT_p	Bluetooth MAC address	6
IP_p	IP address	4
ID_p	Overlay identifier	16
$JOIN$	Join flag	32
GET	Request flag	32
C	Confidentiality level	1
$K_p^-(\dots)$ $K_p^+(\dots)$	Signature operations	

TABLE II
MESSAGE BUILDING BLOCKS AND NOTATION FOR NODE p (SIZE IN BYTES)

Table II provides the notation utilized in the following protocol specification. The blocks and identifiers refer to the perspective of a generic user running the application on a node p in the network (e.g., a smartphone). The actual payload sent and received by each node depends on the high-level messages which are transmitted. As these are application-dependent, we do not impose any constraints on their format or size. With RSA as PKI, key sizes are set to 128 bytes. The two message flags $JOIN$ and GET contain the actual position of a node, for instance specified by two double precision coordinates (16 bytes) as well as an application identifier (16 bytes).
For encoding the three confidentiality levels, 2 bits are sufficient. As recorded in Table II however, the size of the confidentiality level C is set to 1 byte, allowing for potential sub-classification or readjustment of the granularity.

At a very high level, and independent of architectural choice, the collaborative and cooperating distributed application protocol includes basically two steps, before exchanging the payload of the application: find *available* users and find *suitable* users. Available users are users within the deployment range, actually running the current application. Suitable users on the other hand are users also granting the requested confidentiality level. For determining this second set of users, we here informally define the rule that a user requesting a certain confidentiality level is able to grant (at least) the same confidentiality level for a distributed computation. The idea behind this assumption is straightforward: if a user participates in the protocol with a device (for example) unable to sign messages (e.g., no encryption available) it is useless for a strong confidentiality request, as the user himself will not be able to read encrypted messages. Depending on the different approaches depicted next, both finding available and suitable users have to be executed enforcing what we called source obfuscation.

1) Centralized Approach: The first architectural variant we consider is the centralized one. Rather used as reference, we here already introduce some of the basic features of the proposed overall approach. Every device has to communicate with a central service in order to achieve a desired goal. Given our current collaborative filtering example, a device has to communicate its availability in a certain place in order to enable the service to compute the result for the neighbor set at that particular point in time.
Protection of source device data can be achieved by forcing the application not to send device-sensitive data (e.g., the IP address) to the service in a plain fashion. Please note again that the obfuscation of location data is not the goal here, since location data is also inherent to the recommendation and part of the payload users are sending to the service, if encrypted or not.

Step	Function	Format
1	$p.connect(cs)$	$K_p^-(IP_p), C, JOIN$
2	$p.sendData(d)$	$K_p^-(d)$
3	$p.listen()$	$K_p^-(GET)$
4	$p.receiveResult()$	$K_p^+(result)$

TABLE III
CENTRALIZED APPROACH

A node newly arrived in a certain location (e.g., a user entering a restaurant) has to first contact the service (e.g., by starting the application) and send a $JOIN$ message with the current location (e.g., GPS coordinates) along with the requested confidentiality level (C), as shown in Table III.
Step 1 constitutes the connection to a central server cs, sending a request to join the service with the $JOIN$ flag, that contains the position. The central service will then wait for the sensitive information to be sent. As nodes carry on arriving, the central service keeps track of all available nodes

in a certain location and exploits the cumulative data to perform the computation. With regard to the refectory use case mentioned in Section II-A, steps 2 to 4 comprise the actual recommender application in action, in which each node sends the data on which to perform the collaborative filtering algorithm and receives relevant data in return.

2) Decentralized Approach: In this architecture each node participating has to cope with both the network management and the execution of the application-level algorithms. All nodes are equals and thus behave in an identical fashion. For retrieving the desired information, each device first needs to scan its local range for other available devices. With that knowledge on the surrounding area, then a distributed computation based on respective local data is started in order to compute and retrieve the results.

Security within this approach has to be enforced at every step. At the initial bootstrap phase, users should not send plain addresses and other device information in a scan (Bluetooth) or a broadcast message (Wi-Fi). Anonymous requests are not feasible, as users need to be identified as actors within the protocol, so there is a need for granting source obfuscation in a way that contacted nodes are able to trust the user. As an assumption, and depending on the distributed computation taking place, a small amount of information on each device has to be granted. Therefore, there must be some results synchronization methods avoiding well known problems like Byzantine failures [3]. Furthermore, depending on the requested and the available confidentiality level, encryption has to be granted [4], and potentially also nonces and key escrow have to be secured in case of an identity-based security approach like the one in [5]. In the previously discussed centralized approach, the design and the granting of all these assumptions are up to the central entity maintaining the service, whereas in this case, this has to be enforced in a collaborative fashion between participating nodes.

In a dynamic mobile scenario, a full scanning of the actual devices or a broadcast message to be sent to whoever is on-line and within range has to be performed. Moreover, a request for a particular confidentiality level has a set of further requirements: (a) each user has to claim his ability to provide a certain confidentiality level (e.g., securing communications) and (b) each user has to filter the available devices against its desired level. It is worth noting that also this aspect results in a certain additional overhead. After the bootstrap phase, in which an incoming node is actually inserted into the network, the distributed computation can start, following some application-dependent rules (e.g., gossiping techniques as in [6]). The result itself is computed directly within the actual node, but some overhead is carried within the several encrypted messages for the distributed algorithm.
The messages used are summarized in Table IV. The running application is in charge of dealing with incoming Bluetooth connections or broadcast messages and to manage them in case a *JOIN* flag is contained within the message (step 1). All nodes that answer match with the requested confidentiality

Step	Function	Format
1	$p.scanDevices(C)$	$K_p^-(BT_p), C, JOIN$ $K_p^-(IP_p), C, JOIN$
2	$p.retrieveList()$	$K_p^+(BT_1, \ldots, BT_n)$ $K_p^+(IP_1, \ldots, IP_n)$
3	$p.sendData(d)$	$K_p^-(d)$
4	$p.listen()$	$K_p^-(GET)$
5	$p.computeResult()$	$K_p^+(result)$

TABLE IV
DECENTRALIZED APPROACH

level C. By answering, they either send their Bluetooth MAC or their IP address back to the node, and claim to be able to grant the requested C. With that, for instance, they are signaling to be capable of sending encrypted messages. The returned list (step 2) then is used to bootstrap in the network. Steps 3 to 5 in Table IV constitute the steps of the distributed computation.

3) Hierarchical Approach: Much of the overhead induced by broadcast messages or by scanning activity in the decentralized approach can be mitigated with the introduction of a trusted party (tp) for the management of the network. Instead of a flat network structure, this class of nodes can form an additional overlay within the network, taking over these duties. Top-level nodes, so-called management nodes, are beaconing in periodic intervals within transmission range. Those beacons will be intercepted by incoming nodes, which then learn which nearby management node to contact for the participation in the protocol. At the price of the overhead for this beaconing, said bootstrap phase can be alleviated, and the management nodes proceed acting as normal nodes afterwards.

Step	Function	Format
1	$p.connect(tp, C)$	$K_p^-(BT_p), C, JOIN$ $K_p^-(IP_p), C, JOIN$
2	$p.retrieveList()$	$K_p^+(ID_p, l)$
3	$p.sendData(d)$	$K_p^-(d)$
4	$p.listen()$	$K_p^-(GET)$
5	$p.computeResult()$	$K_p^+(result)$

TABLE V
HIERARCHICAL APPROACH

As indicated in Table V, the computation of the result is then carried out as in the decentralized case, for instance exploiting a gossip-based aggregation scheme like in [7]. The basic assumption behind the hierarchical approach is thus the existence of a trusted party within range of the device. A distributed algorithm for the creation of such a party, its maintenance and the distributed election of management nodes is out of the scope of this work and currently under investigation. At this stage, we refer the reader to the current state of the art for resembling supernode approaches in a Peer-to-Peer context. Here, the SG-2 protocol described in [8] for instance provides an efficient, gossip-based peer sampling service minimizing the message size and communication cost. It is important to mention, however, that we pursue a transparent, self-organized random sampling mechanism in order to avoid misuses. All communication in the following is enabled through this trusted

party. Among others, the management nodes index and on request return the list of available nodes within scanning distance matching a confidentiality level (C).

The difference to the decentralized approach thus lies essentially in the bootstrap phase. Instead of involving costly Bluetooth scans or broadcast messages, directed messages to available management nodes are sent for retrieving the list of the participating nodes within a certain distance.

An application-specific overlay is built and maintained via the indexes on the management nodes. For each node, an identifier (ID_p) is computed, for instance through consistent hashing of the device address IP_p (or BT_p) and the requested confidentiality, and sent back to the user along with a list l of available identifiers within a specific range.

III. Experiments

For an initial assessment of our framework, we have conducted experiments and measurements utilizing the network topology generation and simulation tool TopGen [9]. Allowing for a variety of - both modeled and traced - mobilities, dynamic transmission ranges as well as node failure and churn schemes, it provides everything necessary for the analysis of our approach on a simulation level.

A. Setup

We simulated 100 users on a surface of 50×30 meters over a period of 2 hours, modeling a common refectory scenario as outlined briefly already in Section II-A. Of the 100 users, 70 are running the same application for which half of the users (35) require weak confidentiality, whereas 20% (14) chose the strong confidentiality level. The remaining 21 users classify the data exchanged as non-critical and thus run the application with no confidentiality enforced at all.

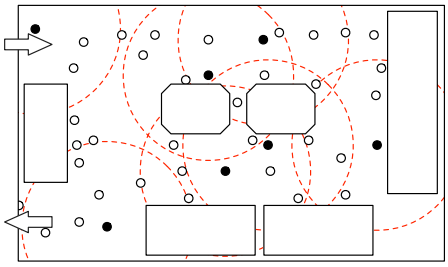

Fig. 1. Simulated food counter area with example set of nodes

Figure 1 schematically depicts the food counter area, where various counters serving different dishes along with salad and dessert bars are located. Following arrival (top left), the customers collect trays and cutlery, then queue and move in between the counters to make their choice and put together their lunch. Finally, they leave the area in order to consume their meal (bottom left). In this particular view, a set of nodes operating on the highest confidentiality level is visualized as solid with the respective transmission ranges dashed (in this case Bluetooth-based).

The refectory setting naturally lends itself to initial simulations as it has clearly defined spatial bounds and a manageable amount of nodes, while at the same time showcasing high dynamics (churn) necessary for testing our framework under stress. The churn rate was set to ca. 60% for a five-minute interval, i.e., the average customer remains no longer than this time span within the area.

The motion follows a probability map mobility pattern congruent with the actual distribution of the counters as displayed in Figure 1, as well as the paths characterized by real queueing and fluctuation behavior of the customers.

The users in this scenario for instance could exchange recommendation data according to an epidemic collaborative filtering scheme as presented in [6] or [7]. However, we do not focus on the actual payload induced by the collaborative filtering but on the analysis of the management messages exchanged within the CoDE protocol that enable the compliance with confidentiality exigencies.

The centralized approach requires an Internet connection, which we presume is held throughout. Technology-dependent transmission ranges here are thus not a factor. For the distributed approaches however, i.e., the decentralized and hierarchical protocol, the transmission ranges vary according to the underlying technologies from maximum 10 meters for Bluetooth as standardized under IEEE 802.15.1 to 100 meters for Wi-Fi according to the current IEEE 802.11n standard [2]. As a reference, we also picked an intermediary 25m standard range with no noise and less packet loss representing the area of interest in a virtual environment in order to assess the protocol for distributed virtual environments based on geometric Peer-to-Peer overlays (P2P DVE).

B. Results

We have obtained a series of results showcasing the individual characteristics of the different approaches and their interrelations with regard to the set of confidentiality levels. The averaged data from five runs per parameter combination is condensed numerically into Tables VI, VII and VIII, as well as visualized in Figures 2 to 4 and the plot series in Figures 5 and 6. The Bluetooth scenario is particularly interesting due to its fragmentary coverage of the simulated space, while the others reflect the case of partial (P2P DVE) and full coverage (Wi-Fi) respectively.

C-Level	Centralized	Hierarchical	Decentralized
0	562.41	1440.44	6087.29
1	578.20	974.44	4984.88
2	512.17	904.77	6106.48

TABLE VI
NUMERICAL COMPARISON OF AVERAGE LOAD IN BYTES (BLUETOOTH)

The performance of the centralized approach is both independent from technology-induced transmission ranges and the confidentiality level requested. It depends on an initial message to a central server with global knowledge which then returns a fully computed result, constituting thus the lower bound for

the distributed approaches. In the Bluetooth runs, on average a little less than 550 bytes are exchanged by protocol-related messages.

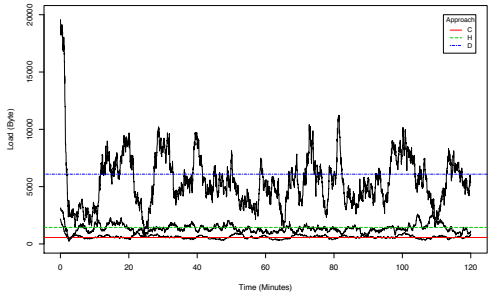

Fig. 2. Confidentiality Level 0 (Bluetooth)

As one can see from the development through from Figure 2 to 4 and the corresponding numbers in Table VI, the hierarchical approach performs substantially better than the entirely decentralized one with rising confidentiality level.

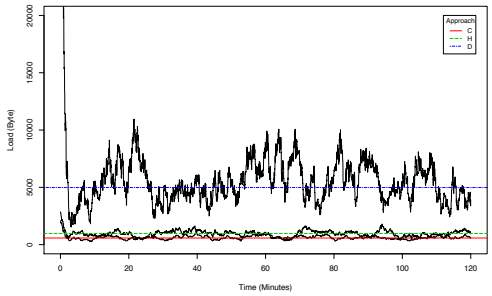

Fig. 3. Confidentiality Level 1 (Bluetooth)

While the latter remains stable in terms of load throughout, the hierarchical approach - a priori on a substantially lower level due to its design - shows an actual load reduction tendency.

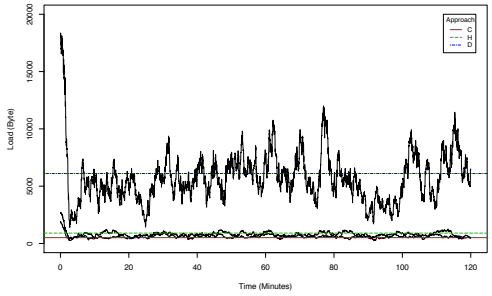

Fig. 4. Confidentiality Level 2 (Bluetooth)

The experiments with the other two settings P2P DVE (cmp. Table VII and Figure 5) and Wi-Fi (cmp. Table VIII and Figure 6) with their varying transmission ranges show, that the hierarchical approach also here outperforms the entirely decentralized approach, converging well to the complete-knowledge centralized one.

In this context, it is important to mention again the underlying assumption that a trusted party already exists. This deliberate exclusion of the associated selection mechanism leads to a slight deviation of the data in favor of the hierarchical approach.

C-Level	Centralized	Hierarchical	Decentralized
0	540.08	4157.62	30661.44
1	654.50	2710.69	32007.89
2	626.56	1141.26	31291.90

TABLE VII
NUMERICAL COMPARISON OF AVERAGE LOAD IN BYTES (P2P DVE)

As we have pointed out already in Section II-B3 however, very efficient distributed mechanisms for that kind of election problem exist, and also the sporadic fan-out beaconing messages of the trusted parties only lead to a marginal increase in terms of overall load. Specialized algorithms are currently under investigation and a combined evaluation is planned in the near future.

C-Level	Centralized	Hierarchical	Decentralized
0	608.06	7285.80	50220.70
1	586.56	3452.72	49808.95
2	665.60	1378.04	49555.28

TABLE VIII
NUMERICAL COMPARISON OF AVERAGE LOAD IN BYTES (WI-FI)

Before moving on to an overview of related research in Section IV, we briefly deal with some other, noteworthy aspects of the results presented here.

First of all, the burst showing at the beginning of each two-hour period in Figures 2 to 6 originates from the initial pseudo-random distribution of the users in the simulated space. It induces this peak at bootstrap when all nodes send their first requests at the same time. This has been introduced deliberately in order to illustrate the hypothetical case of a synchronized join of many users, constituting a possible attack scenario. Also here, the hierarchical approach performs substantially better than the entirely decentralized one.

Finally, another aspect to mention in relation to the data presented here is that the 5-fold averaged evaluation results in the absence of zero values, i.e., periods when for instance no request message is sent, yet they do exist in the individual simulation runs.

IV. Related Work

With the surge of mobile devices and ad hoc networking in the previous decade, quickly focus shifted also to the associated security issues. The research, however, initially focused more on the technical network layer as in [4] or [10], where a survey of threats and solutions as well as an overview of secure routing protocols are given.

Such a routing protocol tailored to mobile ad hoc networks is presented for instance in [11]. It prevents attacks against the route discovery process by exploiting both an asymmetric key infrastructure for securing the routing paths and a symmetric key scheme for providing the confidentiality of the data exchanged. The SPREAD framework [12] takes up another stance and offers a path set optimization algorithm enabling devices to find multiple paths with a desired feature

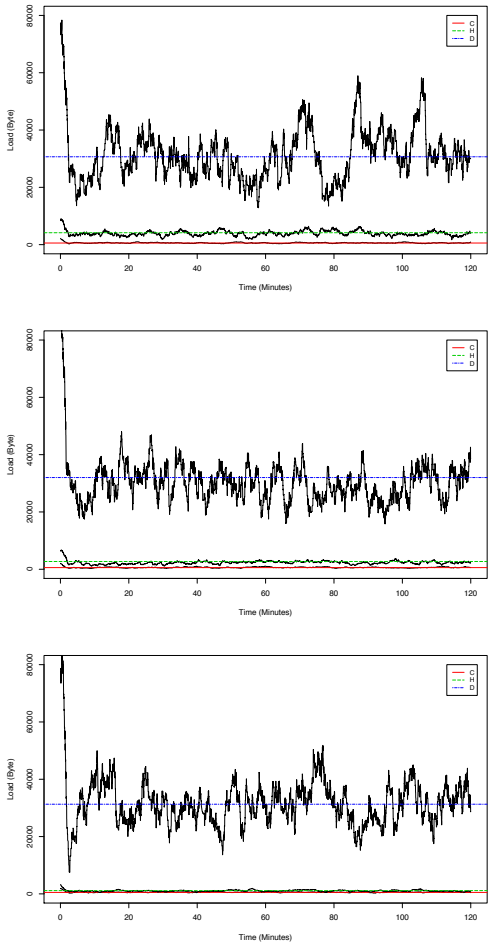

Fig. 5. Confidentiality Level 0-2 (P2P DVE, from top to bottom)

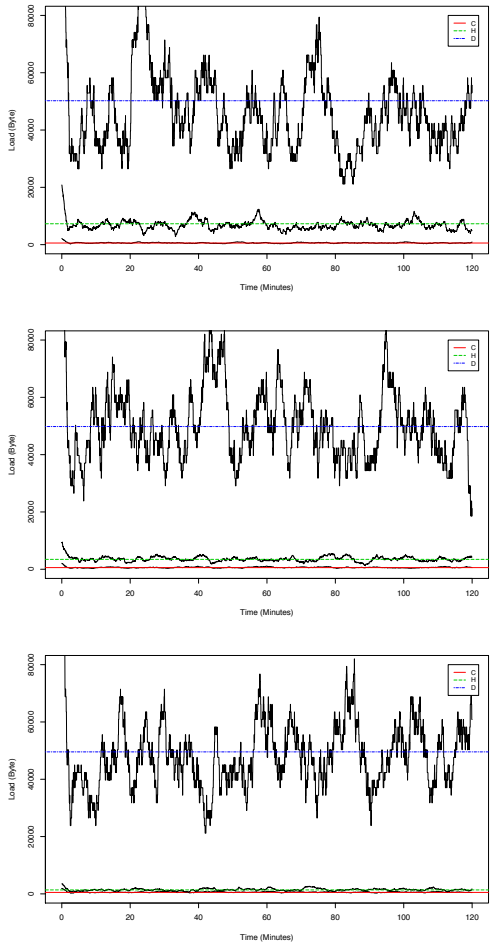

Fig. 6. Confidentiality Level 0-2 (Wi-Fi, from top to bottom)

(e.g., matching best in terms of security). This is achieved through a subdivision and transformation of a secret message into multiple shares following secret sharing schemes, which are then delivered via independent paths to the destination. The exploitation of different levels of security on the same MANET is achieved in [5]. Authors present a security suite covering both security and authentication schemes. This suite can be used with any routing algorithm, at the price of a small delay in delivering messages with strong encryption and authentication.

A differentiated perspective on confidentiality is taken by Clutterbuck et al. [1], where a software audit tool for analyzing the level of confidentiality in WLAN systems is presented. For this, the authors define also three different levels of confidentiality, yet only over a single parameter: the existence or absence of an encryption scheme in the tested network.
Another take on source privacy is presented in [13]. Here, the authors separate sender anonymity, recipient anonymity and relationship anonymity, with the latter constituting the weakest of the three. Based on this distinction, they propose a secure source-anonymous message authentication scheme and a privacy-aware communication protocol. Their analyses show its resilience against common attacks with a high message delivery ratio granting full anonymity to the sender.

Obfuscation techniques for the protection of location privacy form a different, strong line of research [14]–[16]. Albeit not the main focus of our work, on another scale it could also be utilized to achieve location privacy. We are therefore stating a few examples of work dedicated to that issue, again pointing out substantial differences due to their operation mostly on the network layer.
The authors in [17] propose a two-level decentralized proxy-based architecture for intercepting and managing location data requests, capable of reducing network traffic. Location privacy of individual data fragments by means of artificially generated perturbations of associated location information is suggested in [18]. That information is mapped with a relevance value which is utilized to quantitatively measure the degree of privacy artificially introduced. This approach defines a trade-off between accuracy requirements on the part of service providers and the user-side need for protecting personal location data.
Finally, in [19] a formal framework for protecting location

privacy through obfuscation is described. Privacy is granted by exploiting so-called *fake* measurements with the same probability as the real position, through an algorithm for efficiently computing an obfuscated location-based service.

V. Conclusion

In this paper we introduce CoDE, a unified framework enabling application-layer data confidentiality in distributed environments for collaborative social utilities. Based on a differentiated and wider conceptual interpretation of the confidentiality term, it is designed to cover practical needs with regard to the exchange of sensitive data in a fine-grained and user-centric way. To attain this goal, the framework offers a set of different, collaboratively enforced confidentiality levels and architecture-related approaches.

First analyses of the behavior and the load induced by the protocol on different network technologies reveal promising results of the distributed approaches in comparison to the lower-bound centralized method. In particular, the hierarchical approach opens up as a potential trade-off measure enhancing the bootstrap process and overall protocol performance by harnessing asymmetries.

As this work constitutes the initial step towards an integrated approach, there are a few prospective research lines we are interested in following. We are currently carrying out a security analysis of possible attacks against the framework, and the specialized election of the trusted party along with the handling of key revocation is under investigation. Motion in the experiments has been generated by an artificial mobility model, while further analysis will be based on live deployment and recorded traces. Another aspect is that, at this stage, each of the three sets contains nodes with one specific confidentiality level and is distinct to the others. In order to exploit additional resources, however, moving to subsets or introducing further classes might be useful.

Acknowledgment

The work presented in this paper was supported in part by the Fonds National de la Recherche (Luxembourg).

References

[1] P. Clutterbuck, T. Rowlands, and O. Seamons, "Auditing the Data Confidentiality of Wireless Local Area Networks," *Electronic Journal of Information Systems Evaluation (EJISE)*, vol. 10, no. 1, pp. 45–56, 2007.

[2] G. Ananthanarayanan and I. Stoica, "Blue-Fi: Enhancing Wi-Fi Performance using Bluetooth Signals," in *Proceedings of the 7th International Conference on Mobile Systems, Applications, and Services (MobiSys'09)*, Krakow, Poland, 2009, pp. 249–262.

[3] C. Crepeau, C. R. Davis, and M. Maheswaran, "A Secure MANET Routing Protocol with Resilience against Byzantine Behaviours of Malicious or Selfish Nodes," in *Proceedings of the 21st International Conference on Advanced Information Networking and Applications Workshops (AINAW07)*, Ontario, Canada, 2007, pp. 19–26.

[4] J.-P. Hubaux, L. Buttyán, and S. Capkun, "The Quest for Security in Mobile Ad Hoc Networks," in *Proceedings of the 2nd ACM International Symposium on Mobile Ad Hoc Networking & Computing (MobiHoc 2001)*, Long Beach, CA, USA, 2001, pp. 146–155.

[5] J. T. Chang, S. Gundala, T.-S. Moh, and M. Moh, "VESS: a Versatile Extensible Security Suite for MANET Routing," in *Proceedings of the 2009 IEEE Pacific Rim Conference on Communicatios, Computers and Signal Processing (PacRim09)*, Victoria, B.C., Canada, 2009, pp. 944–950.

[6] R. Schifanella, A. Panisson, C. Gena, and G. Ruffo, "MobHinter: Epidemic Collaborative Filtering and Self-Organization in Mobile Ad-Hoc Networks," in *Proceedings of the 2008 ACM Conference on Recommender Systems (RecSys'08)*, Lausanne, Switzerland, 2008, pp. 27–34.

[7] P. Gratz and J. Botev, "Collaborative Filtering via Epidemic Aggregation in Distributed Virtual Environments," in *Proceedings of the 5th International Conference on Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom 2009)*, Washington, D.C., USA, 2009, pp. 1–9.

[8] G. P. Jesi, A. Montresor, and O. Babaoglu, "Proximity-Aware Superpeer Overlay Topologies," *IEEE Transactions on Network Science and Service Management*, vol. 4, no. 2, pp. 74–83, 2007.

[9] I. Scholtes, J. Botev, M. Esch, A. Höhfeld, H. Schloss, and B. Zech, "TopGen - Internet Router-Level Topology Generation Based on Technology Constraints," in *Proceedings of the 1st International Conference on Simulation Tools and Techniques for Communications, Networks and Systems (SIMUTools 2008)*, Marseille, France, 2008, pp. 1–10.

[10] D. Wang, M. Hu, and H. Zhi, "A Survey of Secure Routing in Ad Hoc Networks," in *Proceedings of the 9th International Conference on Web-Age Information Management (WAIM'08)*, Zhangjiajie Hunan, China, 2008, pp. 482–486.

[11] K. S. Ng and W. K. G. Seah, "Routing Security and Data Confidentiality for Mobile Ad Hoc Networks," in *Proceedings of the 57th IEEE Semiannual Vehicular Technology Conference (VTC 2003 - Spring)*, Jeju, Korea, 2003, pp. 1821–1825.

[12] W. Lou, W. Liu, and Y. Fang, "SPREAD: Enhancing Data Confidentiality in Mobile Ad Hoc Networks," in *Proceedings of the 23rd IEEE Conference on Computer Communications (IEEE Infocom 2004)*, Hong Kong, 2004, pp. 2404–2413.

[13] J. Ren, Y. Li, and T. Li, "SPM: Source Privacy for Mobile Ad Hoc Networks," *EURASIP Journal on Wireless Communications and Networking*, vol. 2010, pp. 5.1–5.10, 2010.

[14] A. B. Brush, J. Krumm, and J. Scott, "Exploring end user preferences for location obfuscation, location-based services, and the value of location," in *Proceedings of the 12th ACM International Conference on Ubiquitous computing (Ubicomp'10)*, Copenhagen, Denmark, 2010, pp. 95–104.

[15] M. L. Damiani, E. Bertino, and C. Silvestri, "The probe framework for the personalized cloaking of private locations," *Transactions on Data Privacy*, vol. 3, no. 2, pp. 123–148, 2010.

[16] K. P. N. Puttaswamy and B. Y. Zhao, "Preserving privacy in location-based mobile social applications," in *Proceedings of the Eleventh Workshop on Mobile Computing Systems & Applications (HotMobile'10)*, Annapolis, MD, USA, 2010, pp. 1–6.

[17] P. Bellavista, A. Corradi, and C. Giannelli, "Efficiently Managing Location Information with Privacy Requirements in Wi-Fi Networks: a Middleware Approach," in *Proceedings of the 2nd International Symposium of Wireless Communication Systems (ISWCS 2005)*, Siena, Italy, 2005, pp. 91–95.

[18] C. A. Ardagna, M. Cremonini, S. De Capitani Di Vimercati, and P. Samarati, "An Obfuscation-Based Approach for Protecting Location Privacy," *IEEE Transactions on Dependable and Secure Computing*, vol. 8, no. 1, pp. 13–27, 2011.

[19] M. Duckham and L. Kulik, "A Formal Model of Obfuscation and Negotiation for Location Privacy," in *Proceedings of the 3rd International Conference on Pervasive Computing (PERVASIVE 2005)*, Munich, Germany, 2005, pp. 152–170.