SEVENTH FRAMEWORK PROGRAMME
THEME 3
Information and Communication Technologies

Grant agreement for:
Collaborative project, Small and medium-scale focused research project (STREP)

Deliverable D2.3:

A Set of Design Rules and Technical Recommendations for Adapting the Network Infrastructure

Project acronym: FRONTS
Project full title: Foundations of Adaptive Networked Societies of Tiny Artefacts
Grant agreement no.: 215270

Responsible Partner: UDRLS
Report Preparation Date: Monday, May 1, 2011

Contents

1 Introduction

The activities of WP2 are organized in three tasks: Task 2.1 - Adapting the communication infrastructure, Task 2.2 - Adapting the internal structure and roles of artefacts and Task 2.3 - Security for tiny artefacts, and have generated a large number of research papers and technical reports.

In D2.2 we presented five schemes for Designing the Adaptable Network Infrastructure. These schemes, that allow tiny devices to effectively react to the dynamic conditions in which they operate, have been selected among the large number of research targets identified in D2.1 as some of the most relevant features to be implemented on the limited resources of the devices.

In this deliverable we focus on two of the schemes presented in D2.2, namely the *fully decentralized recommendations in opportunistic networks* (SC1) and the *helper based security* (SC2) schemes and we discuss design rules and technical recommendations derived by the performed experiments.

The unifying experiments reported in deliverable D4.6 have shown that most of the techniques proposed by FRONTS can be actually exploited to build a society of tiny artefacts. Nevertheless, those experiments have also shown that the severe memory constraints imposed by the underlying hardware technology (current Isense nodes have 88KB of programming memory), mandate to accurately select the functions that can be implemented on the tiny artefact.

For this reason, among the five schemes presented in D2.2 we have selected SC1 and SC2 to be further investigated in this deliverable. The motivations of such choice can be summarized as follows:

- SC1 shows how a very limited amount of local information stored on nodes with limited computation and communication capabilities, can evolve over time (i.e. adapt) due to opportunistic interactions to support the provisioning of global accurate services (e.g. recommendations);
- SC2 shows how the issues of constrained resources can be circumvented with the support of *helpers*, namely special devices woven into the environment allowing tiny artifacts to perform computations otherwise impossible with their limited resources (e.g. security functions).

In the *helper based security* scheme, we envision a set of devices, generically called the *security helpers* that help small artefacts to perform cryptographic operations and that are part of cryptographic protocols involving tiny artefacts with the specific task of helping them. In this report, we show how helpers are useful for coping with the above discussed limitations of tiny artifacts in other contexts as well.

Helpers represent the way to convey the necessary support to allow tiny devices to quickly react to the environment in those cases where their limited memory, communications capabilities and computational power are not sufficient.

In this perspective, helpers are devices equipped with more resources with respect to the devices they are helping. Nevertheless, in SC1 we have shown how very limited devices such as passive RFID tags, can be used to help more powerful devices (smart readers) in satisfying their needs through the provisioning of accurate recommendations. However, the most interesting characteristic of such scheme, is the ability of providing a valuable global service (i.e. recommendation) exploiting very limited resources in a fully decentralized setting in which only opportunistic communication are available.

In section 2 we discuss the design rules and technical recommendations that emerge in this scheme, with reference to a new application scenario in which new social relationships are inferred using only the limited information carried in short messages (SMS) opportunistically exchanged among mobile users.

In section 3 we show how helpers can be used to significantly increase the security of a society of tint artefacts. It is well known that RFID tags can be read by anyone and the string stored on a tag, even though it is a ciphertext, can be used to trace the tag. In section 3.1, we discuss how helpers and in particular *randomizers* can guarantee that tiny artefacts, such as RFID tags, are difficult to be traced by unauthorized third parties.

Tiny artefacts are subject to attacks (like reset attacks) which are not possible if the adversary does not have physical access to the computing device (which is often the case for tiny artefacts). In section 3.2 we present a technique to guarantee immunity against node capture on a large scale. In this case, the helpers are the *broadcasting systems* used to refresh the set of temporal keys.

2 Fully Distributed Computation in Opportunistic Networks

2.1 Motivation and problem description

The limitation imposed by the hardware and the dynamic and challenging environments in which tiny artefacts might operate, in particular in mobile scenarios, make hard (or even impossible) in some contexts the construction of reliable network connections over which services can be delivered to the end-users. In these contexts, an opportunistic networking approach can be more effective. In opportunistic networks, communication opportunities are based on sporadic and intermittent contacts, disconnections and re-connections may frequently occur, and the assumption on the existence of an end-to-end path between the source and the destination is released.

In deliverable D2.2 we presented a scheme in which tiny artefacts cannot directly communicate and information exchange among them is mediated by users' collective and unpredictable interaction patterns, according to the principles of opportunistic networking. In such context, when tiny artefacts can eventually communicate, they need to effectively update and enrich their local view stored in the limited available memory, in order to support the provisioning of global and accurate services.

2.2 A Set of Design Rules and Technical Recommendations

In [BBC$^+$10] we propose and implement a proof of concept as a Java Card Applet of a fully decentralized approach for recommending new contacts in the social network of mobile phone users using SMS. With respect to existing solutions, our approach is characterized by some distinguishing features. In particular, the application we propose does not assume any centralized coordination: it transparently collects and processes user information, the so called profile, and exchanges it with other users. This information is used to recommend new friendships to other users and is exchanged transparently and opportunistically, by using the residual space in standard short messages occasionally exchanged between users. As a consequence, we do not ask users to change their habits in using SMS.

The key ingredient to implement such service, subject to the severe space constraints imposed by the residual space of SMS (the full available space in an SMS is about 140 bytes), are sketches. Sketches are compact and to - some extent - privacy preserving representations of the information of interest (i.e. the profile). The size of a sketch is small, 40 bytes in the present implementation. Since hash functions are used to compute sketches, they support some kind of privacy that is suitable for some of the envisioned applications.

Design rule: Sketches are compact and privacy preserving representation of data, that can be used to efficiently and effectively compute relevant statistics over data (e.g. estimate the similarity between sets) in a fully decentralized way.

According to [BBC$^+$10] recommendations are based on users' similarity. Roughly, two users are similar (i.e. they have a potential relationship) if their profiles are similar. A profile is a set of characteristics that describes a user and his behavior. The Jaccard coefficient, defined as $J(A, B) = \frac{|A \cap B|}{|A \cup B|}$, is a standard measure to estimate the similarity between two sets A and B and thus, it can be used to estimate the similarity between profiles. It turns out, that we do not need to have full knowledge of users' profiles to compute their similarity.

In section 3.1 of [BBC$^+$10], we described a technique to estimate the Jaccard coefficient of sets A and B using their sketches Sk(A) and Sk(B). This means that the similarity between two sets, irrespectively from their sizes, can be estimated by means of their sketches, which are of size much lower than that of the originating sets. Therefore, we claim the following technical recommendation.

Technical Recommendation: Similarity between two sets can be approximated using suitable defined sketches of the sets.

In [BBC$^+$10] we performed several experiments using the dataset provided by the Reality Mining Project. Precision and recall are two standard metrics to measure the quality of recommendation systems. As shown in figure 1, as the number of hash functions used to compute the sketches increases, as better we can estimate the Jaccard coefficient and provide good recommendations. However, already 10 hash functions are sufficient to provide good quality recommendations.

Technical Recommendation: There is a trade-off between the accuracy of the estimation and the size of the sketches (i.e. used memory).

2.3 Related FRONTS Technical Reports

In [BFV10] we used sketches to allow a central authority to select a subset of users whose past positions provide a good coverage of a given area of interest, without explicitly georeferencing users.

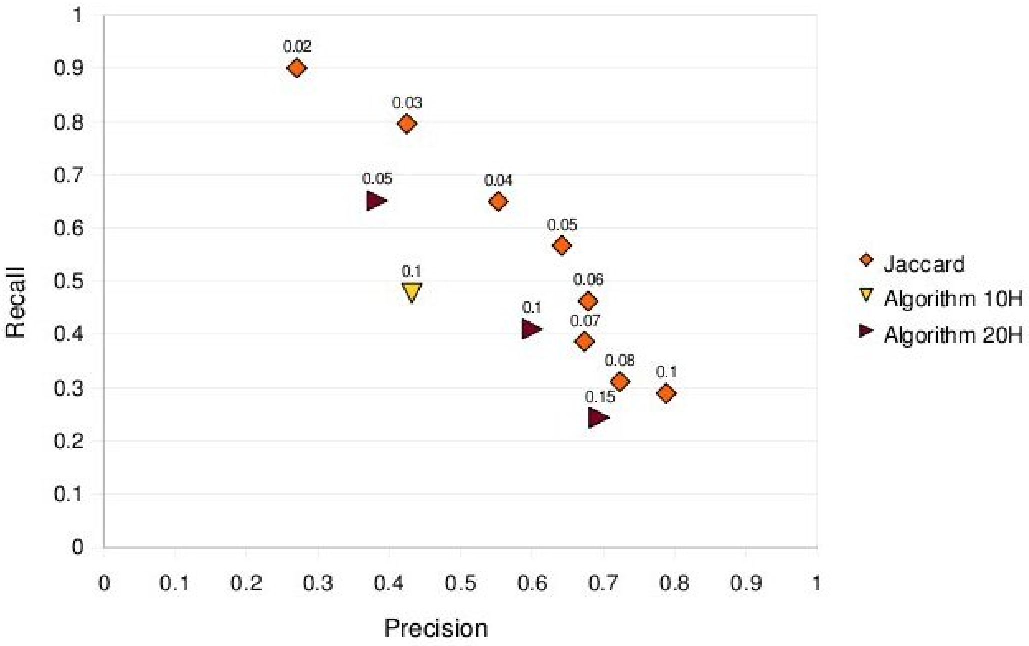

Figure 1: The figure shows the accuracy of precision and recall obtained using the Jaccard coefficient and its estimation with sketches based on 10 and 20 hash functions (see [BBC+10]).

Related first and second year FRONTS Technical Reports In [BCMSV10] we propose and investigate the effectiveness of stigmergy-based, collaborative filtering techniques. These are particularly interesting for use in pervasive systems of small devices with limited communication and computational capabilities, such as RFID tags. In particular, we do not assume any transmission capabilities between the tags: information exchange among them is mediated by users' collective and unpredictable navigation patterns.

In [CCK+10] we presented a FRONTS unified scenario in which MURPESSs - MUlti Radio Pedestrian Energy Scavenging Sensors - obtain recommendations interacting with other MURPESSs or with passive RFID tags.

3 Helper Based Security

In this section we show how the issues of constrained resources, can be circumvented with the support of *helpers*, namely special devices woven into the environment allowing tiny artifacts to perform computations otherwise impossible with their limited resources.

3.1 Private Tracking

In deliverable D4.6, the experimental results of two modules related to tracking are discussed. The purpose of the tracking module (TR) is to provide tracking information for mobile targets, to mobile trackers inside a wireless sensor network topology. We observe that in this module, *Passive* nodes can be considered as helpers. They allow the *Trackers* to track the *Targets* efficiently propagating the traces through the topology in a binary tree-gradient fashion. The Private Tracking module (PTR) foresee the use of helpers (the randomizers) to avoid the traceability of tiny artefacts by malicious trackers. In the following we discuss the design rules and technical recommendations that are derived from the experimental activity on Private Tracking.

3.1.1 Motivation and problem description

Security is a major requirement for any system and the study of security for tiny artefacts poses new conceptual problems other than the ones studied in Internet security. This new challenge calls for new paradigms and schemes on which security needs to be based. The limited computational power of the devices that must perform the computation required by the cryptographic protocol is the main novelty.

In D2.2 we envisioned a set of devices, generically called the *security helpers*, that help small artefacts to perform cryptographic operations and that are part of cryptographic protocols involving tiny artefacts with the specific task of helping the tiny artefacts. We envision the security helpers as woven into the environment and providing their service to all the devices that request it. One crucial question is "*how much can we trust a security helper*?" Ideally, one would like to be able to make minimal trust assumptions on the helpers but typically we believe it is reasonable to assume that the helper perform as prescribed. However, the following design rule must be consider in order guarantee misbehavior and manipulation of helpers.

Design rule: In no case, the helper must receive secret information from the artefacts (like for example secret keys).

3.1.2 A Set of Design Rules and Technical Recommendations

In D2.2 we discussed an efficient construction of *untraceable* tags in the model with randomizers, to be used for passive RFID tags carrying private encrypted information. The interaction with the anonymizers is intended to guarantee that tags are not traced, and this is achieved by re-randomizing the ciphertext (that is, by producing a new ciphertext that carries the same message). The re-randomization procedure is carried on without having access to the private key used to generate the ciphertext. This extra important property guarantees privacy of the information even with respect to the randomizers. Our construction uses pairing-based cryptography that allows high level of security with very limited storage, making our construction applicable to RFIDs.

Design rule: Pairing-based cryptography guarantees a high level of security while requiring limited storage, and it is applicable to RFIDs.

The security guaranteed by the proofs in [BDCP09] does not cover the case in which the same RFID is read by two (possibly distinct) adversaries (called the tracers) between two consecutive re-randomizations. Indeed in this case the two tracers can certainly trace the RFID tag.

During the second year of the project we have conducted experiments and shown the effectiveness of our approach against static tracers.

Design rule: Randomizers can help in preventing RFID tracing in a static scenario.

Clearly if tracers are able to physically follow tags then they are trivially able to trace the tags. We observe though that in this case the tracing happens independently from the existence of the tag as the adversary actually traces the individual carrying the tag. We are interested instead in limiting the tracing ability of mobile adversaries that move independently from the tags. We considered established and well-studied mobility models for tags and tracers and we study the probability of success of a set of tracers present in the network to trace one tag as a function of their number and of other parameters. We found that the reading range of a tracer is crucial for its success probability and since devices with longer reading range are more expensive we studied the tradeoff between number of tracers and their reading range. We conducted experiments to evaluate this tradeoff with different number of tags and randomizers and size of the world. The results among the considered models were qualitatively the same (i.e. similar patterns were obtained). However the so called *Boundless mobility model*, which we believe to better reflect actual behavior of people, gave better results.

Our experiments show that to achieve the assigned probability of success (i.e, 0.1 probability in tracing at least a tag, see Figure2), the adversary must deploy either a high number of tracers or tracers with a wide reading range. Further experiments to evaluate the efficiency of the randomizing algorithms have been conducted with the the context of our unifying experiment and are reported in Deliverable D4.6. We observe that, if the number of randomizers increases, then tracers need to employ more resources, either higher reading range or number of deployed tracers. The results of the third year can be summarized in the following technical recommendations and design rules.

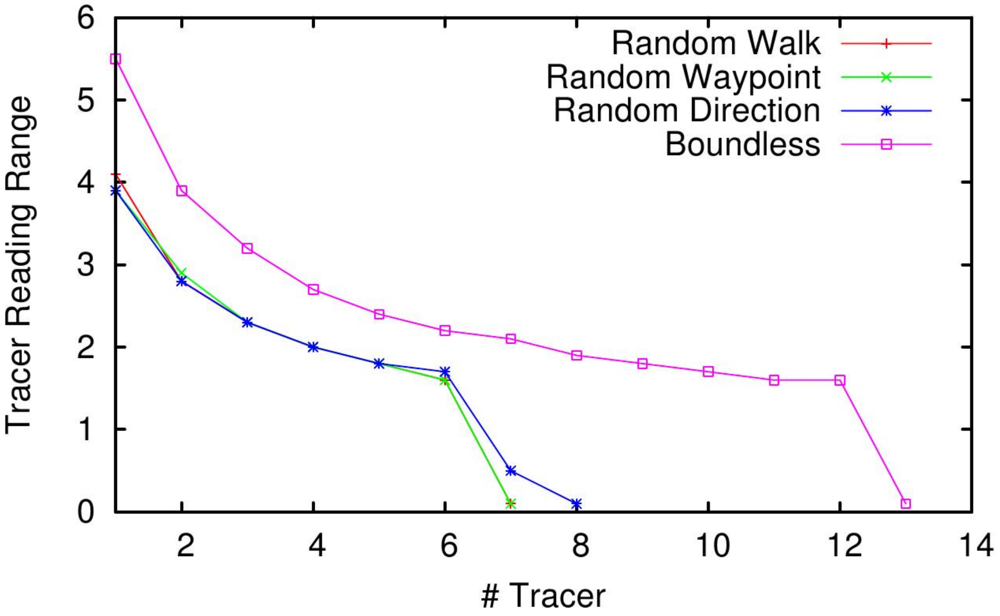

Figure 2: Breach of Security. Probability 0.1 of success for a tracer, namely the probability in tracing at least a tag, as a function of the number of deployed tracers and tracers reading range according to different mobility model. The figure refers to a scenario in which 40 tags and 16 randomizers are placed in a unit square of size 16.

Design rule: Randomizers can help in preventing RFID tracing in a dynamic scenario.

Technical recommendation: It is simpler to read the content of an RFID tag than to write on it. Therefore the success probability of a tracer is directly influenced by its reading range.

Technical recommendation: The boundless mobility model appears to be the mobility model that is more resilient to tracing.

Technical recommendation: Analyse the tradeoffs between the number of tracers (adversaries) the number of tags and the number of helpers (randomizers).

3.1.3 Related FRONTS Technical Reports

In [DSV09a, DSV09b] we have extended the standard model for security and privacy introduced by Vaudenay to take into account Denial of Service attacks. Specifically, we extended the model to better reflect the real-world scenario, where these attacks are easy to mount (e.g., by physically destroying/making inactive tags). Besides refining Vaudenay's privacy model to deal with DoS and DoS-like attacks, we introduce an additional privacy notion, referred to as *semi-destructive* privacy, which takes into account hardware features of some real-world tags. Then, we show an efficient RFID protocol that, by only using symmetric-key cryptography (which is much more efficient than public-key cryptography), satisfies the notion of semi-destructive privacy, under standard complexity-theoretic assumptions.

3.2 Establishment of Keys

The Group Key Establishment (GKE) module discussed in deliverable D4.6, aims at distributing one, potentially long (and hence strong) secret key to all nodes within a given group. In the following we discuss the design rules and technical recommendations that are derived from the experimental activity on Group Key Establishment.

3.2.1 Motivation and problem description

Implementation of basic security mechanisms (such as link confidentiality and authentication) requires key material for executing cryptographic protocols. Key management turns out to be a nontrivial task - building an X.509 infrastructure fails even in a much easier scenario of personal identity management. For tiny devices there are additional problems: limitations on memory size, lack of tamper-resistance, computational limitations (lack of crypto co-processors for standard ASP in tiny devices), no physical control over devices (among others they can be captured by an adversary) and heterogeneity of the system. Using symmetric schemes with key predistribution, despite its weaknesses, may solve many of these problems. In [CGK09] we have designed a set of schemes that helps to keep key material relatively immune against key capture. The scheme is based on refreshing the set of keys (with backward compatibility) when visiting special sites, where additional checks can be performed based on physical presence. However, this way of upgrading random key predistribution does not help to solve the problem of device authentication: it is impossible just as in case of the regular random key predistribution. Additionally, updating the keys might be tedious in many application scenarios.

3.2.2 A Set of Design Rules and Technical Recommendations

In [CGK10] we design a new paradigm of managing keys from random key predistribution. The key pool is no longer a single one: there is pool of permanent keys (distributed according to random key predistribution) and short-time temporal pools. For regular tasks (like establishing secure links) only temporal keys are used. The scheme requires periodic broadcasting (by an helper) of a new set of temporal keys. These keys are encrypted by permanent keys. The main (and counter intuitive) trick is that the pool of temporal keys should be smaller than the pool of permanent keys by a constant factor, say k times. Consequently, each temporal key is encrypted k times with k different permanent keys. The assignment of repetitions is a (pseudo)random one and changes completely from stage to stage. Consequently, two devices may share a temporal key due to possession of a pair of different permanent keys. The number of these *accidental* keys is quite high (see figure 3) and enable to detect forging a device by an adversary holding some number of the permanent keys. Unexpectedly, even if the adversary captures a quite large fraction of all permanent keys, detection probability remains fairly high, namely the adversary cannot impersonate somebody else.

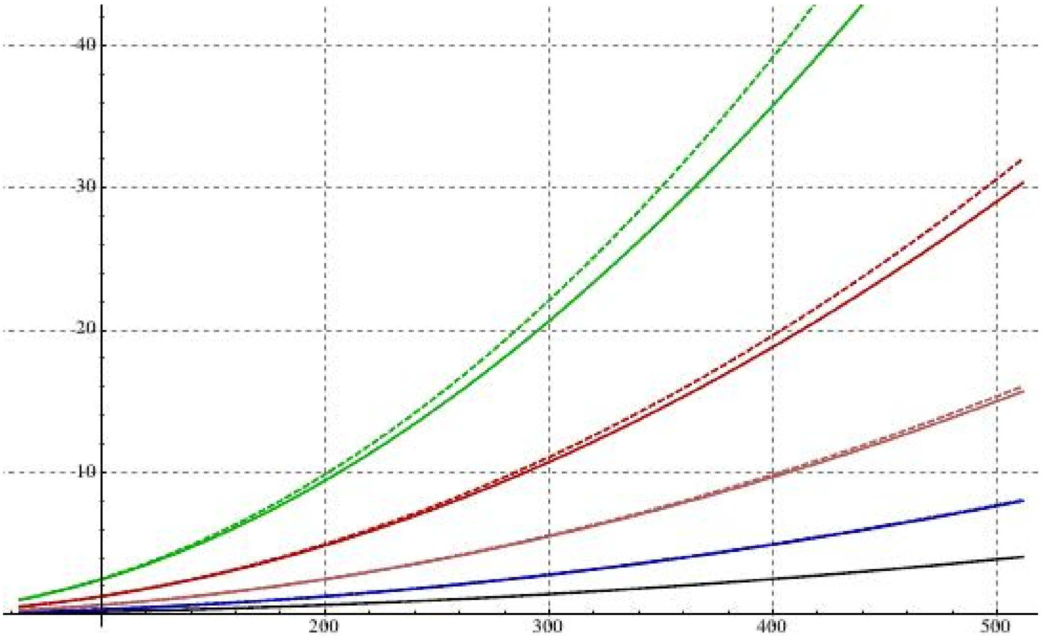

Figure 3: The expected number of temporary keys shared between two nodes (y-axis) as a function of the size of the pool of permanent keys (x-axis). The different curves represent different size of the pools of temporal keys; pools of smaller size (green) share more keys (see [CGK10]).

Implementation of a key establishment protocol accomplished within WP4 is based on [CGK09]. It also exploits some follow up results concerning connectivity of the network discovered during implementation work.

Whenever broadcasting by an helper is technically feasible, [CGK10] provides a powerful extension for long time security.

Design rule: For key predistribution, whenever broadcasting is technically feasible, temporal key pools should be used.

Technical Recommendation: The major limitation for symmetric key management on sensor nodes is memory size for the keys and neither computational power nor the code size for auxiliary procedures. Additional memory need not to be protected in a strong way by hardware means or by the operating system, if temporal keys are to be stored there.

3.2.3 Related FRONTS Technical Reports

[CGK10] provides a key re-distribution scheme and the architecture of permanent key pool and temporal key pools. Behavior of the schemes is investigated using tools of analytic combinatorics yielding concrete and provable guarantees of system security and safety.

Related first and second year FRONTS Technical Reports In [CGK09] we describe a scheme upgrading random key predistribution with the concept of levels (which can be applied to [CGK10] as well) and continuous updating the keys to new levels.

4 Further contributions

The above presented schemes do not represent all the work done in this work package. In the following we briefly report on the results concerning other schemes presented in D2.2, namely the Dynamic file allocation strategy and Geographic routing with energy efficiency schemes.

In the *Dynamic file allocation strategy* scheme, we investigated power-aware strategies for sharing data in mobile ad hoc networks. In [JLNR10] we study the problem of data gathering in wireless networks and more particularly the cooperative routing mechanisms that balance the energy consumption among the nodes participating to the process. We discuss the relationships between different optimality criteria (single and multi objectives) and show the equivalence. We also determine the energy optimal network structures that ensure the optimality of energy balance routing mechanisms. Moreover, we characterize maximal network structures that are energy optimal. These structures have the properties that adding edges does not help in gathering more data. In the *data gathering problem* that we consider, the nodes make a multi-hops networks, generate data and, cooperate to route the data towards a unique sink. We assume that the nodes can choose between different levels of energy transmission. In summary, our results provide an algebraic characterization of the optimal communication graphs over a given network. For instance, we show that two levels of transmission are sufficient in some situations.

In the *Geographic routing with energy efficiency* scheme we propose geographic routing as a tool to support energy efficient communications among the tiny artefacts. In geo-routing, nodes have first to become aware of their position via some localization method. However, traditional localization techniques are dependent on hardware that is sometimes unavailable (e.g. GPS), or on sophisticated virtual localization calculus which have a costly overhead. In [HJLR10] instead of actually localizing nodes in the physical two-dimensional Euclidean space, we use directly the raw distance to a set of anchors to produce multi-dimensional coordinates. We prove that the image of the physical two-dimensional Euclidean space is a two-dimensional surface, and we show that it is possible to adapt geographic routing strategies on this surface, simply, efficiently and successfully.

References

∗ [BBC^{+}10] Luca Becchetti, Lorenzo Bergamini, Ugo Colesanti, Luca Filipponi, Andrea Vitaletti, Giuseppe Persiano, and Elisa Baglioni. A lightweight privacy preserving sms-based recommendation system for mobile users In *Proceedings of the fourth ACM conference on Recommender systems*, RecSys '10, pages 191–198, 2010. FRONTS-TR-2010-43.

∗ [BCMSV10] Luca Becchetti, Ugo Colesanti, Alberto Marchetti Spaccamela, and Andrea Vitaletti. Recommending items in pervasive scenarios: models and experimental analysis *Knowledge and Information Systems*, pages 1–24, 2010. FRONTS-TR-2010-3.

∗ [BDCP09] Carlo Blundo, Angelo De Caro, and Giuseppe Persiano. Untraceable tags based on mild assumptions In *2nd SETOP International Workshop on Autonomous and Spontaneous Security*, 2009. FRONTS-TR-2009-36.

∗ [BFV10] Luca Becchetti, Luca Filipponi, and Andrea Vitaletti. Opportunistic privacy preserving monitoring In *In proceedings of PhoneSense 2010*, 2010. FRONTS-TR-2010-44.

∗ [CCK^{+}10] Ioannis Chatzigiannakis, Ugo Colesanti, Spyros Kontogiannis, Guy Leshem, Alberto Marchetti Spaccamela, Jan Mehler, Giuseppe Persiano, Paul Spirakis, and Andrea Vitaletti. Murpess - multi radio pedestrian energy scavenging sensor network In *eChallenges e2010*. IEEE, IIMC International Information Management Corporation, October 2010. FRONTS-TR-2010-25.

∗ [CGK09] Jacek Cichon, Jaroslaw Grzaslewicz, and Miroslaw Kutylowski. Key levels and securing key predistribution against node captures In *Algorithmic Aspects of Wireless Sensor Networks*, volume 5304 of *LNCS*, pages 64–75. Springer-Verlag, July 2009. FRONTS-TR-2009-66.

∗ [CGK10] Jacek Cichon, Zbigniew Golebiewski, and Miroslaw Kutylowski. From key predistribution to key redistribution In *ALGOSENSORS*, pages 92–104, 2010. FRONTS-TR-2010-36.

∗ [DSV09a] Paolo D'Arco, Alessandra Scafuro, and Ivan Visconti. Revisiting dos attacks and privacy in rfid-enabled networks In *ALGOSENSOR*, LNCS, 2009. FRONTS-TR-2009-27.

∗ [DSV09b] Paolo D'Arco, Alessandra Scafuro, and Ivan Visconti. Semi-destructive privacy in rfid systems In *The 5th Workshop on RFID Security – Revised Selected Papers*, 2009. FRONTS-TR-2009-63.

* [HJLR10] Florian Huc, A Jarry, Pierre Leone, and Jose Rolim. Virtual raw anchor coordinates: a new localization paradigm In *Algosensors*, 2010. FRONTS-TR-2010-41.

* [JLNR10] Aubin Jarry, Pierre Leone, Sotiris E. Nikoletseas, and José D. P. Rolim. Optimal data gathering paths and energy balance mechanisms in wireless networks In *DCOSS*, pages 288–305, 2010. FRONTS-TR-2009-61.