Path-Stamps: A Proposal for Enhancing Security of Location Tracking Applications

Ana I. González-Tablas and Arturo Ribagorda

Carlos III University, Leganés 28911, Madrid, SPAIN
aigonzal@inf.uc3m.es, arturo@inf.uc3m.es

Abstract. Location tracking technologies are penetrating increasingly in industrial environments. Several challenges arise when people or mobile assets are tracked. One of the main problems location tracking poses is security. In this position paper we want to address the long-term authentication and accountability of location tracking history information or path. In order to accomplish this aim, we generalize existent location-stamp definition, formulate the concept of path-stamp and present a path-stamping architecture and protocol. We define path-stamp as the evidence that, by itself or used with other information, allows a third party to prove that an entity has moved along a path enforcing a determined path-stamping policy. Our proposed solution is build on location-stamps, relative temporal authentication by using linking schemes, and path-stamp entanglement.

1 Introduction

In this last decade two specially important developments have significantly changed our world: the World Wide Web and the widespread adoption of digital mobile telephony. Several research issues and opportunities have emerged from the union of these two technologies in addition to other developments such as GPS, WLAN, and the evolution of electronic gadgets as laptops and handhelds. Many of these challenges are still not completely solved [4]. The addressing of these issues by academic and industrial communities, together with social and legal institutions, is leading step by step Weiser's vision to reality [14].
In this ubiquitous (or pervasive) computing world, location aware applications have been granted with a huge attention. Location and context awareness, along with its social and legal implications, are one of the ubiquitous computing challenges [12].
Several academic proposals have been developed in this area. See [3] for a good survey of context-aware computing research, and [6] for a more specific taxonomy of the properties of location systems and an evaluation of some of the most representative research and commercial location-sensing systems according to the proposed taxonomy. Several industrial markets for location based services have arisen and more are expected to arise. Three main sub-markets can be identified: tracking services, localized information services, and fun and entertainment. Our interest in this paper focuses in location tracking applications.

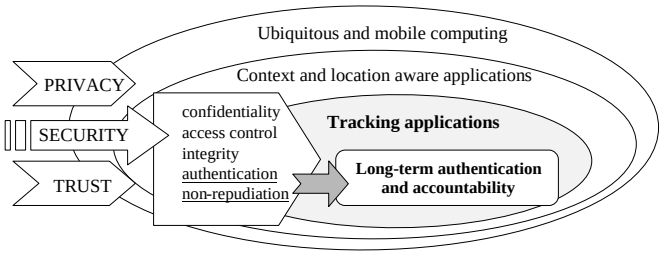

Fig. 1. Situation of the addressed problem in the context

This kind of services can be applied to such interesting areas like personal safety [1] and other applications such as fleet management, mobile office, field services and field sales. Nowadays, location tracking services are an standard technology to consider in industrial deployments.

A crucial aspect to consider in location aware applications, and specifically in location tracking ones, is security [12] [4]. Enormous efforts keep being carried out to meet privacy and trust challenges in location aware systems.

Let us present the following scenario. Alice is an efficient nurse at St. John's hospital. Suppose that St. John's hospital has installed an electronic monitoring system that tracks every movement its employees make at work. This system provides real-time and historical information on the location of Alice in addition to some other performance data about her work. This historical data allows her supervisor Larry to decide at the end of the year her wage increase and the continuance of her contract. Suppose now that Larry doesn't want Alice to continue working at St. John's anymore for some obscure personal reasons. So, he tampers Alice's records to provide a justification to the Human Resources Committee. Alice is defenseless regarding the digital tampered proofs and she gets fired. How could she prove that she had been at the right place at the right time within a certain period of her contract? How could she prove that she has been an efficient nurse at her employer's eyes and that she doesn't deserve to be fired? On the other hand, the case could be backwards, and Alice play the role of the "bad" nurse that tampers her records to conceal her irresponsible work behavior at St. John's. Or even she could have been a van driver that wants to get more free time for lunch instead of working properly.

These examples are not too far from real life, as it is presented in [2] and [11], and they point out the problem we want to address in this position paper. Our work addresses the problem of assuring location tracking information of an entity A along the time who has committed herself someway with an entity B for being tracked according to a certain policy. The main objective is that afterwards A's tracking history can be verified by authorized external entities. In other words, we want to propose a solution that provides long-term authentication and ac-

countability for location tracking applications (see Fig. 1).
This paper is structured as follows, section 2 presents related works to our problem and justifies the need for addressing long-term authentication and accountability for location tracking applications. Section 3 describes our solution by formulating the concepts of path, generalized location-stamp and path-stamp, and by presenting our proposed path-stamping architecture and protocol. Finally, in Section 4 some conclusions, remarks and future works are presented.

2 Related Work

Several industrial applications use location technologies to track mobile assets, persons or vehicles driven by these along a path. On the other hand, location history is also used in several academic context-aware applications (see Hightower and Borriello context-aware applications survey [6]). As location technologies penetrate deeply into our society, more relevance will be granted to location tracking data. This information will be more considered in contracts and will increasingly affect the relation between the located entity and the verifier of her path. In some close future, legal validity will be probably granted to entities' location tracking information. Usually this data is kept in clear or with some access control enforcement but this is not enough because it can be modified to benefit or harm any of the concerned entities. Few of the existent industrial applications and academic proposals address location tracking security. Specifically, how to provide long-term authentication and accountability for location tracking information has not been yet considered, to our knowledge.
The proposal of Kabatnik and Zugenmaier [8] is closely related to this problem. They point out the necessity that arises because location aware services use uncertified location information, and propose certifying this location information for GSM mobile terminals. This certified or long-term authenticated location information is called location stamp by them. The main objective of their work is to provide location liability for the signing of contracts. Certainly, their work has a lot in common with the problem addressed in this paper, although the main difference stays in that they do not consider the certification of the location information along time, that is, the certification of the whole location tracking history or path.
Zugenmaier, Kreutzer and Kabatnik enhance their previous work in a proposal of location stamps that could have legal impact for locating GSM subscribers at a specific moment [15]. Again, the main lack of this work, considering the problem addressed in this paper, is that no history information is certified.
Location-stamps are inspired in well known time-stamps [15]. A time-stamp certifies that some document has been created or signed before or at a certain time. Time-stamping schemes can be classified into three types: simple, linking and distributed. Simple schemes are so that time-stamps do not include data from other time-stamps, whereas linking ones do include it. Distributed schemes are such that the time-stamp is computed by several issuers which collaborate. Une [13] realizes a deep analysis on the security of time-stamping schemes and pro-

poses a security evaluation method and classification. Simple schemes provide absolute temporal authentication, while linking schemes provide relative temporal authentication [7].
On the other hand, Maniatis and Baker have recently addressed secure history preservation of the states of a system which provides a service within a domain [9]. The problem they consider is similar to the one addressed in this paper, because their principal aim is to obtain tamper-evident historic record of the system states, with provable relative temporal authentication. However, they do not consider at all location tracking applications. They call their solution secure timelines, and is based in time-stamping schemes and authenticated dictionaries. They also propose a technique, which they call timeline entanglement, that aims to create a common, tamper-evident history of the collective timelines of a set of mutually distrustful domains. The main difference between their work and the work presented in this paper is the object of certification: in its case it is the history of the states of a service within a domain (or a set); in our case, the certification considers the location tracking history of an entity. The problem behind is similar, but the context of their proposal and the one addressed in this paper are radically different.
The works cited above have as main goal aim to provide long-authentication

	What	**Temporal authentication**	**Location authentication**	**History**
Simple time-stamps [13]	Existence or signing of a document	Absolute	-	-
Linked time-stamps [13]		Relative	-	-
Location stamps [8][15]	Existence of an entity or signing of a document	Absolute	Absolute	-
Generalized location-stamps	Event or action	Absolute	Absolute	-
Path-stamps	Path (location history) of an entity	Relative	Absolute	Location
Secure timelines [9]	History of the states of a system	Relative	-	System states

Fig. 2. Comparison of related works in front of what path-stamps aim to certificate

and accountability. The main difference between them is the particular fact or object which they want to certify (see Fig. 2). The distinctive characteristics of what we attempt to certify in this paper are an entity's location and its history or evolution along time. Our proposed solution is inspired in linking time-stamp schemes, in location-stamps, and in the entanglement technique.

3 Proposal on Path-Stamps for Location Tracking Applications

In this section, our proposal is described. First we formulate the concepts of path and location stamp. Then, we generalize the location-stamp definition, and

propose the concepts of path-stamp and path-stamping policy. Finally, a path-stamping architecture and protocol are presented.

Path Definition. We define *path* of an entity A the ordered sequence of locations li A moves on along time: $p(A) := (l_i)_{i=1,n}$

Location Stamps. Zugenmaier, Kreutzer and Kabatnik [15] [8] define location-stamp as the certificate used to prove that a mobile under the control of some certain subscriber was seen at certain time or that he signs some specific document at some certain location at a certain time.

Generalized Location-Stamps. We propose to generalize this concept of location stamp by defining *generalized location-stamp* as the evidence or information that either by itself or when used in conjunction with other information is used to establish proof about an event or action that happens or has happened at a certain location. Therefore, their location stamp [15] [8] can be interpreted as a particularization of the generalized location-stamp. From now on, we will use indistinctibly both "generalized location-stamp" and "location-stamp" to refer to "generalized location-stamp", otherwise it is clearly indicated.

Time in Generalized Location-Stamps. Although the generalized location-stamp definition does not include explicitly time, it is considered by the use of "happens or has happened" because it is implicit in the meaning of the verb. Therefore, a generalized location-stamp can be used to ascertain that something happens (that is, now, at a certain time) at a certain location if the fact is proved to happen in "real-time" or within a small time frame (as in [8] and [15]). Otherwise, or additionally, it can be used to ascertain that something happened at a certain location prior to the issuing of the location-stamp.

Path-Stamp and Path-Stamping Policy. Our solution for providing long-term authentication and accountability to A's location tracking history is based in both concepts defined above: path and location-stamps. A's path, as we see it, is an ordered set of locations. So, a first proposal could consider a set of ordered location-stamps issued by a path-stamp issuer and computed for each location contained in the path, altogether becoming what we may call a path-stamp. However, it is important to notice that the meaning or interpretation of the path-stamp obtained depends in great manner on the selection of the specific locations which compose the path. So, it is strongly determinant which *path policy* is enforced to select the set of locations.

Consequently, we define *path-stamp* as the evidence that, by itself or used with other information, allows a third party to prove that the located entity A has moved along a path enforcing a determined path-stamping policy. We define *path-stamping policy* as the set of conditions that determine the computation of a location-stamp for an entity A at some certain location l_i in order to compute a path-stamp.

The conditions of the path-stamping policy must include IDA, the identification under which A is located, and IDB, the identification/s of the authorized receiver/s of the path-stamp. A *path-stamp authorization policy*, defining A's path-stamps access control rules, must also be specified.

Some Path-Stamping Policy Examples. An example of a condition spec-

ification in the path-stamping policy could be "compute a location-stamp for entity identified as IDA whenever the relative distance from A's current location l_i to last IDA's location l_{i-1} stamped is greater than a parameter Mld, or Maximum location distance, that states the maximum distance allowed between two consecutive location-stamp, or whenever the relative temporal distance between current time t_i and time t_{i-1} when last location-stamp was computed is greater than 1/mf, being mf a parameter that states the minimum allowed frequency between consecutive location-stamps".
Another example could also be "compute a location-stamp whenever an entity identified as IDA moves on some of the following determined and unordered set of locations: $L_0, L_1, L_2, ..., L_{n-2}, L_{n-1}, L_n$".
Relative Temporal Authentication by Linking Location-Stamps. Both cited examples are valid according to our definition of path-stamping policy, but first one includes time measure explicitly in their definition and second one does not. As the set of locations in the path are ordered, an external verifier should be able of verifying this order with the information or evidence provided by the path-stamp. So, although in some path-stamp policies time measure might not be considered, the location-stamps must prove the order of its computation independently. This requirement is just addressed by the concept of relative temporal authentication [7].
Relative temporal authentication is based in one-way hash functions [10] (assuming its existence) and it has been extensively used in time-stamping linking schemes [13]. Applying a linking schema to build the path-stamp, each location-stamp includes data from previous location-stamps. This way we preserve temporal order of location-stamps within the path-stamp.
Security Considerations, Publishing and Path-Stamp Entanglement. For the scope of this paper, we consider that the path-stamp issuer, or path-stamp authority, is a trusted third party, so she is not supposed to collude with another entity to fake the path-stamp by taking out one of the location-stamps or changing any of them in someway. But, if the location-stamps are cryptographically linked, it is more difficult for her, as she must change consequently the whole rest of location-stamps which comprise the path-stamp. Furthermore, considering that the path-stamp issuer is reliable does not prevent that if the whole set of location-stamps are not cryptographically bounded, a malicious claimant of the path-stamp takes out one certain and not desired location-stamp from the path-stamp. If the verifier is not careful in his verification procedure, he might be deceived.
As suggested by Just in [7] to prevent *fake attacks*, the chain or some part of it must be published from time to time in some widely witnessed medium such a newspaper. We publish the linking information of some location-stamps on the on-line site (public database) associated to the path-stamp issuer, and use this data to initialize next path-stamps. The published location-stamps are the last ones closing the path-stamps, and others are randomly selected from just issued location-stamps. This way we obtain an entanglement of linking data from different path-stamps, complicating a possible forgery of the path-stamp authority.

We must remark that the security of the proposed path entanglement requires further verification, although other works use similar techniques [9].

Requirements and Architecture. Two main actors are identified in this scenario: the *located entity* A; and the entity that will prove A's path, *the claimant*, who could probably be B, although A could also play the role of proving herself path. The path-stamping requirements of A and B comprise: (1) long-term authentication and accountability of A's path under a certain path-stamping policy; (2) authentication of the located entity A; (3) confidentiality of path information, including time or other conditions if they are present, associated to A's identity; (4) access to path information must only be granted to authorized entities enforcing a certain path-stamp authorization policy; (5) privacy of located entity must be respected. In this work we address only the first one of the above requirements.

We assume that entities A and B establish a commitment (or contract) that states that A must be located during her work frame-time under some Path-Stamping Policy (PSP), which reflects the conditions of A's tracking, and the Path-Stamp Authorization Policy (PSAP). A may chose to be located or tracked by a pseudonym which only B, or the authorized entities, can correlate with real A's identity. The commitment document must reflect B's counterpart, if it exists. Afterwards, A, B, or an external entity V, or *verifier*, in case of dispute, must be able of proving or verifying that A had accomplish some route according to some certain conditions.

Neither A, nor B trust each other for keeping a nave log of A's tracking information, as both are implicated entities, so they require the services of a trusted *Path-Stamping Authority (PSA)*. B contracts A's location tracking to her. The PSA is the entity who issues for several entities their path-stamps. Also the PSA is who creates the A's *Path-Stamping Policy Enforcement Agent (PSPEA)* entity every time A initiates the path-stamping service, which is in charge of enforcing the Path-Stamping Policy and, consequently, she is who requests the issuing of each location-stamp for A at certain locations l_i according to the Path-Stamping Policy, and who verifies the correctness of the Path-Stamping Authority's procedures related to her requests. The Path-Stamping Authority has two databases. The first one, DB_v, is private and there is where location-stamps and path-stamps are stored, being accessible for clients after the enforcement of the Path-Stamp Authorization Policy. The public one, DB_u, is where she publishes the linking information of some selected location-stamps.

We consider the existence of a *Location Service*, which can both locate an entity and track her movements, and which we assume that provides trusted and reliable location information. We assume also that the Location Service authenticates A previously to her localization, and, if time values are used, they are provided by a trusted time source. Hereafter we describe the path-stamping issuing protocol (see Fig. 3). In it, we have also included absolute time measure for illustrating reasons.

Path-Stamping Protocol Initialization. First, B (or A) contracts the path-stamping service to a Path-Stamping Authority PSA of an entity identified as

IDA according to a path-stamping policy PSP, which includes a path-stamp authorization policy PSAP. This PSP is signed by the implicated entities (A, B and the PSA). A is given a special device that allows her location and tracking as IDA. (1) IDA requests the initialization of a path-stamp sending to PSA a $REQPSINIT(IDA, P)$ being P the signature by A, B and PSA over the hash value of the PSP. PSA verifies her self signature over P and, if it succeeds, (2) initiates a path-stamping policy enforcement agent PSPEA bounded securely to the PSP and requests the tracking of IDA to the Location Service. (3) The path-stamp PS_m requested by IDA is initialized with the creation of a new record in the private path-stamp database DB_v. PS_m is initialized with a path-stamp serial number m, IDA, the PSA identification IDPSA, and P.
The first location-stamp LSm,1 of PSm is computed as follows. (4) PSA authenticates IDA, with uniqueness and timeliness guarantees [10]. (5) PSA requests to LS to locate IDA sending her $REQLOC(IDA, IDPSA)$, and (6) LS sends her back $LOC(IDA, l_1, IDLS)$ being l_1 IDA's location. (7) PSA selects (p, R_p, L_p), which is the last published linking information in DB_u and corresponds to some certain location-stamp LS_p. She computes $L_1 = (R_p, H(L_p))$. Then, she computes $R_1 = (m, n(1), IDA, IDPSA, P, l_1, t_1, L_1)$ being n(1) the $LS_{m,1}$'s serial number ($LS_{m,1} = LS_{n(1)}$) and t_1 the time when it is computed. Afterwards, she computes a signature over $R_1, S_1 = sig_{PSA}(m, n(1), IDA, IDPSA, P, l_1, t_1, L_1)$. Finally, $LS_{m,1} = (R_1, S_1)$. (8) PSA adds to PS_m record in DB_v $LS_{m,1} = (R_1, S_1)$. (9) PSA sends to PSPEA the first computed location-stamp with $PSPEAINIT(LS_{m,1}, IDPSA)$.
PSPEA verifies the correctness of $LS_{m,1}$ and PSA procedures to compute it. (10)

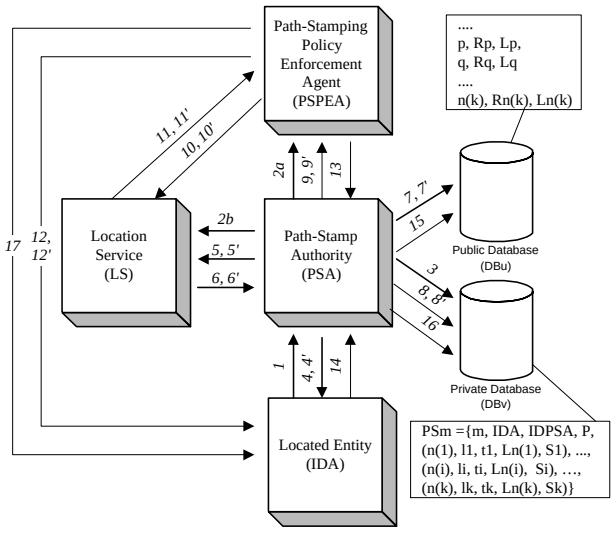

Fig. 3. Path-Stamping Architecture

First she requests to LS to locate IDA sending her $REQLOC(IDA, IDPSPEA)$, and (11) LS sends her back $LOC(IDA, l'_1)$ being l'_1 IDA's location and t'_1 time when LS's message is received. PSPEA verifies that $|l'_1 - l_1| < E_l$ and $|t'_1 - t_1| < E_t$ with E_l and E_t previously defined for example in the PSP. She stores m and verifies that $H(L_0)$ is the hash of the last published linking information in DBu. She computes again L_1 by using data included in the location-stamp, and compares it with the received one. Then, she verifies that P is the signature by A, B and PSA over the hash value of her PSP, and that S_1 corresponds to the signature by PSA of R_1; afterwards, PSPEA verifies that $LS_{m,1}$ is included in DB_v.
If every verification step has succeeded, (12) PSPEA sends IDA an acknowledge of the path-stamp initialization *ACKPSINIT (IDA, P, m, n(1),* $l_1, t_1, L_{n(1)}$*, IDPSPEA).*
Otherwise, she requests another initialization. If this second one fails again, she sends an error message to IDA and B, and asks PSA to reflect it in DB_v.
Path-Stamp Computation. As IDA moves, LS tracks her movements. Given IDA's location (requested with certain frequency by PSPEA to LS) and some other needed information, such as time, e.g., PSPEA enforces when the conditions established in PSP trigger the computation of a new location-stamp $LS_{m,i}$ with i the next location-stamp to be included in PSm. (13) At that moment, PSPEA requests to PSA a new location-stamp computation for IDA $REQLS(m, n(i-1), IDA, IDPSPEA)$. The value n(i-1) is the serial number of the last location-stamp included in PS_m.
Steps from numbered as (4) to (12) are repeated -$(4)'$ to $(12)'$-, except some minor changes, but instead of computing $LS_{m,1}$, $LS_{m,i}$ is computed and the linking information, $L_i = (R_{i-1}, H(L_{i-1}))$, is fetched from the last location-stamp issued for PS_m. Following, changes are described. In $(9)'$ PSA sends $PSPEALS(LS_{m,i}, IDPSA)$ instead of $PSPEAINIT(LS_{m,1}, IDPSA)$. In $(11)'$, PSA compares the stored m value with the received one instead of storing it, and verifies that L_{i-1} is the linking information of last location-stamp of PS_m. Also, in $(12)'$ acknowledge to IDA is omitted. Otherwise, she requests the issuing of another location-stamp.
Finalization of Path-Stamp Computation. This process repeats until the entity decides to finalize the tracking. Then (14) IDA sends PSA *REQPSFIN (IDA, P).* At this moment the PSA computes last $LS_{m,k}$, repeating steps $(4)'$ to $(12)'$, and (15) publishes its linking information
$(n(k), R_{n(k)}, L_{n(k)})$ in DB_u. (16) Finally, PSA builds the path-stamp PS_m using data in PS_m record and publishes it in the private database DB_v for authorized entities. The published final path-stamp is checked by the PSPEA.
PSm =m, IDA, IDPSA, P, (n(1), $l_1, t_1, L_{n(1)}, S_1$*), ..., (n(i),* $l_i, t_i, L_{n(i)}, S_i$*), ..., (n(k),* $l_k, t_k, L_{n(k)}, S_k$*)*
At the end, if every thing succeeds, (17) PSPEA sends IDA an acknowledge of the path-stamp finalization $ACKPSFIN(IDA, P, m, n(k), l_k, t_k, L_{n(k)}, IDPSPEA)$.
Otherwise, she requests another finalization. If this second one fails again, she sends an error message to IDA and B, and asks PSA to reflect it in DB_v.
Path-Stamping Verification Protocol. In order to verify a whole path-stamp

given PS_m, the verifier first has to validate all the signatures S_i. Then, he requests the PSA the linking data of location-stamp p and n(k). He has to generate values L_i, using data from the path-stamp, and compare them with L_i values received in the path-stamp. Again, he requests to PSA the path-stamp record in DB_v, and compares its content with the received path-stamp. Last, he verifies that the calculated linking information of location-stamp n(k) has been published in DB_u.
Another issue is the verification of the enforcement of PSP. For this problem, we propose that PSPEA be a secure authenticated code, so its reliability can be proved before path-stamps are issued. PSPEA would sign the two initialization and finalization acknowledges that she sends to located entity, and these records can be used to verify the first and last location-stamps in the path-stamp chain independently from the PSA.

4 Conclusions

In this position paper we have shown that the long-term authentication and accountability of location tracking history information or path of an entity is an unresolved problem. In order to address this problem we have proposed the concept of path-stamps, and presented a path-stamping architecture and protocol. Our solution is build using location-stamps, linking schemes for relative temporal authentication, and path-stamp entanglement.
However, some remarks on our proposal and further work must be made. The architecture that we propose is strongly centralized. This feature could be some way inadequate in ubiquitous and computing environments, so in the future this has to be enhanced by considering a distributed architecture and protocol. The linear linking schemes applied have two main drawbacks. These are first the efficiency, as the verifier has to compute same data than the issuer, and, second, the huge quantity of information that the issuer has to store for clients disponibility. Some more advanced linking schemes could be studied in the future.
An issue that we have not addressed in this paper, but crucial to the success of location tracking certification, is the differences between authenticating a device (or a general entity) and authenticating some certain person. Zugenmaier, Kreutzer and Kabatnik address this problem for GSM terminals in [15]. This must be incorporated to the path-stamping protocol too.
Another issue that must be addressed is how much an implementation of a path-stamping system would cost, and whether industry would find it worthy. The path-stamping model we propose must be mapped to real location aware systems and to a possible universal location system that integrate these.

References

1. Applewhite, A.: What Knows Where You Are? Personal Safety in the Early Days of Wireless. IEEE Pervasive Computing 1:4 (2002) 4-8

2. Buswell, C.: Surveillance and Nurses: the Use and Misuse of Electronic Monitoring. Research for Nursing Practice 1:2 (1999)
3. Chen, G., Kotz, D.: A Survey of Context-Aware Mobile Computing Research. Dartmouth Computer Science Technical Report TR2000-381 (2000)
4. Davies, N., Gellersen, H.-W.: Beyond prototypes: Challenges in Deploying Ubiquitous Systems. IEEE Pervasive Computing 1:1 (2002) 26-35
5. Haber, S., Stornetta, W.S.: How to Time-Stamp a Digital Document. Journal of Cryptology. 3:2 (1991) 99-111
6. Hightower, J., Borriello, G.: Location Systems for Ubiquitous Computing. IEEE Computer, August 2001 (2001) 57-66
7. Just, M.: Some Timestamping Protocol Failures. In Proc. of Internet Society Symposium on Network and Distributed System Security (1998)
8. Kabatnik, M., Zugenmaier, A. Location Stamps for Digital Signatures: a New Service for Mobile Telephone Networks. In Proc. of ICN 2001, Colmar, France (2001)
9. Maniatis, P., Baker, M.: Secure History Preservation through Timeline Entanglement. In Proc. of the 11th USENIX Security Symposium 2002, San Francisco, CA, USA (2002).
10. Menezes, A.J., van Oorschot, P.C., Vanstone, S.A. Handbook of Applied Cryptography. Ed. CRC Press 1997.
11. Rockhold, J.: The Business of Where. Wireless Review (2001)
12. Satyanarayanan, M.: Pervasive Computing: Vision and Challenges. IEEE Personal Communications 8:4 (2001), 10-17
13. Une, M.: The Security Evaluation of Time Stamping Schemes: The Present Situation and Studies. IMES Discussion Papers Series 2001-E-18 (2001)
14. Weiser, M. The Computer of the 21st Century. Scientific American 265:3 (1991) 66-75
15. Zugenmaier, A., M., Kreutzer, Kabatnik, M.: Enhancing Applications with Approved Location Stamps. In Proc. of the IEEE Intelligent Network 2001 Workshop (IN2001), Boston, MA, (2001)