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

Ana I. González-Tablas, Benjamín Ramos, and Arturo Ribagorda

Universidad Carlos III de Madrid, 28911 Leganés, SPAIN
{aigonzal,benja1,arturo}@inf.uc3m.es

Abstract. Location tracking technologies are penetrating increasingly in industrial environments. Several challenges arise when people or mobile assets are tracked. Security is one of the main problems that location tracking poses. 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, we generalize the existent definition of location-stamp, then we formulate the new concept of path-stamp and, finally, we present a path-stamping architecture and protocol. We define a 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 some certain path enforcing a determined path-stamping policy. Our proposed solution is built on location-stamps, relative temporal authentication 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 [3]. 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 [12].

In this ubiquitous computing world, location aware applications are granted with a huge attention. Location and context awareness, along with its social and legal implications, are one of the ubiquitous computing challenges [10].

Several academic proposals have been developed in this area. See [2] for a good survey of context-aware computing research, and [5] 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. Several industrial markets for location based services have risen 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. This kind of services can be applied to such interesting areas like personal safety [1], fleet management, mobile office, field

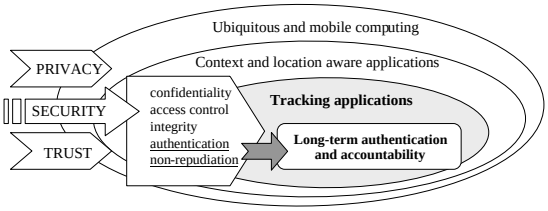

Fig. 1. Situation of the addressed problem in the context

services and field sales. Nowadays, location tracking is an standard technology to consider in industrial developments.

Security is a crucial aspect in location-aware applications, and more specifically in location tracking ones[10] [3]. Enormous efforts are being carried out to meet privacy and trust challenges in location aware systems.

Let us present the following scenario. Alice is a site inspection agent employed by some security company. She is destined to site inspect an industry complex. She is in charge of covering a certain route in some conditions (e.g. frequency, duration). She works hard and efficiently. Unfortunately a clever and lucky enemy of the company manages to get into the complex and deceives the security measures without getting noticed. Then he damages irreparably some main equipments. As a consequence, the company suffers a high amount of loses. The owner of the industry complex asks for responsibilities to the security company. The security company might question Alice behavior in order to blame her. Alice is defenseless as she has no proofs of having done correctly her duties, or having fulfilled the route assigned to her following the stipulated conditions. If she had such a proof, it could be shown that it is not her blame but site inspection procedures blame.

This example is not too far from real life and it points 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 accountability 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 in location tracking applications. Section 3 describes our solution by formulating the concepts of path, generalized location-stamp and path-stamp, and presenting our 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 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 [5]). As location technologies penetrate deeply into our society, more relevance will be granted to location tracking data. This information will be more and more considered in contracts and will increasingly affect the relation between the located entity and the verifier of its 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 these measures are not enough because data can be modified to benefit or harm any of the concerned entities. Few of existent industrial applications and academic proposals address location tracking security. Specifically, it has not been yet considered how to provide long-term authentication and accountability for location tracking information, to our knowledge.

The proposal of Kabatnik and Zugenmaier [7] 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 by them *location stamp*. Their main objective is to provide location certification for the signing of contracts. Certainly, their work has a lot in common with the problem addressed in this paper, although the main difference is that they do not certificate the location information along time, that is, the path or the whole location tracking history.

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 [13]. 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 [13]. 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. In distributed schemes the time-stamp is computed by several issuers. Une [11] realizes a deep analysis on the security of time-stamping schemes and proposes a security evaluation method and classification. Simple schemes provide absolute temporal authentication, while linking schemes provide relative temporal authentication [6].

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 [8]. 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 relative temporal authentication that can be proved. However, they do not consider at all location tracking applications. They call their solution *secure timelines*, and are based in time-stamping schemes and au-

Table 1. Comparison of related works in front of what path-stamps aim to certificate

	What	Authentication: Time	Location	History
Simple time-stamps	Existence or signing of a document	Absolute	-	-
Linked time-stamps	Existence or signing of a document	Relative	-	-
Location stamps	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	History of the states of a system	Relative	-	System states

thenticated 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 ours is the object of certification: in its case it is the history of the states of a service within a domain (or a set). The problem behind is similar, but contexts are radically different.

The works cited above have as main goal to provide long-term authentication and accountability. The main difference between them is the particular fact or object which they want to certify (see Table 1). 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.

3.1 Path and Location-Stamps

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

Location Stamps. Zugenmaier, Kreutzer and Kabatnik [13] [7] 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 the subscriber 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, Zugenmaier *et al.*'s location stamp [13] [7] can be interpreted as a particularization of our proposed generalized location-stamp. From now on, we will use indistinctively 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 [7] and [13]). Additionally, it can be used to ascertain that something happened at a certain location prior to the issuing of the location-stamp.

3.2 Path-Stamps

Our solution certifying A's location tracking history is based in 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 of the path, becoming altogether 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.

Path-Stamp. 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.

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 set of locations in order to compute a path-stamp.

The conditions of the path-stamping policy must include the identification under which A is located and the identification/s of the authorized receiver/s of the path-stamp. A *path-stamp authorization policy* for A's path-stamps must also be specified. One example of condition specification in the path-stamping policy could be "compute a location-stamp whenever the relative distance from A's current location l_i to last A's location l_{i-1} included in the path-stamp is greater than a parameter ϵ_l that states the maximum distance between two consecutive location-stamps, 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 ϵ_t, being $1/\epsilon_t$ the minimum frequency between consecutive location-stamps". A second one could also be "compute a location-stamp whenever an entity A moves on some of the following determined and unordered set of locations: $loc_0, loc_1, loc_2, ..., loc_{n-2}, loc_{n-1}, loc_n$".

3.3 Relative Temporal Authentication by Linking Location-Stamps

Both cited examples are valid according to our definition of path-stamping policy, but first one includes explicitly time measure, on the contrary than in the second one. As the set of locations in the path are ordered, an external verifier should be able to verify 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 in an independent manner. This requirement is just addressed by relative temporal authentication [6].

Relative temporal authentication is based in one-way hash functions [9] (assuming its existence) and it has been extensively used in time-stamping linking schemes [11]. Applying a linking scheme to build the path-stamp, each location-stamp includes data from previous location-stamps. This way, temporal order of location-stamps within the path-stamp is preserved.

3.4 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 a malicious claimant of the path-stamp takes one certain (not desired) location-stamp out from the path-stamp. If the whole set of location-stamps are not cryptographically bounded and the verifier is not careful in the verification procedure, he might be deceived.

As suggested by Just [6] 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 the next path-stamps. Last location-stamp in every issued path-stamp is published, in addition to some randomly selected location-stamps just after its computation. This way we obtain an entanglement between 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 [8].

3.5 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* (B), although A could also play the role of proving her own path. The path-stamping requirements of A and B comprise the following ones, although in this work we address only the first one:

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.

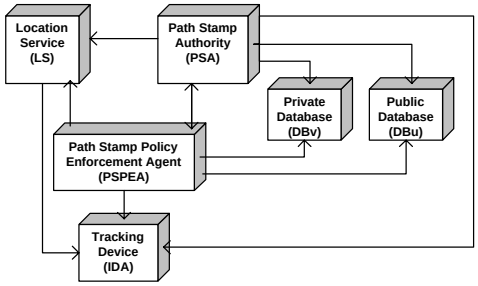

Fig. 2. Entities considered in the Path-Stamp System and their relations

We assume that entities A and B establish a commitment (or contract) that states that A must be located during some frame-time under some Path-Stamping Policy (PSP) including a Path-Stamp Authorization Policy ($PSAP$). A may choose to be located or tracked by a pseudonym which only 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 or *verifier* (V), in case of dispute, must be able of proving or verifying that A had accomplished some route according to some certain conditions.

Neither A, nor B trust each other for keeping a naïve log of A's tracking information, as both are implicated entities, so they require the services of a trusted *Path-Stamping Authority (PSA)* -see Fig. 2. B contracts a path-stamping service for A's location tracking to the PSA, that is who issues the path-stamps.

PSA also creates a *Path-Stamping Policy Enforcement Agent (PSPEA)* every time that A initiates the path-stamping service. $PSPEA$ is in charge of enforcing the PSP and consequently who requests the issuing of each location-stamp for A according to PSP and who verifies the correctness of PSA's procedures related to its requests. PSA has two databases. The first one, DB_v, is private and where location-stamps and path-stamps are stored. DB_v is accessible for authorized entities enforcing $PSAP$. The public one, DB_u, is where the linking information of some selected location-stamps is published.

We consider the existence of a *Location Service* (LS), which can locate and track an entity, and that it is assumed to provide trusted and reliable location

information. We also assume that LS authenticates entities before its location, and that possible used time values are provided by a trusted time source.

3.6 Path-Stamping Protocol

Hereafter we describe the path-stamping issuing protocol (see referenced steps in Fig. 3) and the path-stamping verification protocol. We have included absolute time measure for illustrating reasons.

Path-Stamping Issuing Protocol: Initialization First, B (or A) contracts to some certain PSA the path-stamping service for A's tracking according to some PSP. The PSP is signed by the implicated entities (A, B and PSA). A is given a special device that allows her location and tracking as IDA. The unique identification of the tracking device is considered to be securely bound to IDA. For the scope of this paper the tracking device will be identified as IDA (see remarks in the Conclusions section).

1-6 Path-Stamp Request and Initialization. A requests PSA via her tracking device the initialization of a path-stamp. PSP is included in the request to select one of the several applicable policies which may exist between A and PSA. PSA verifies the correctness of IDA's request and the signatures on PSP.
PSA initiates a new $PSPEA$ securely bound to PSP, and requests IDA's tracking to LS. A new path-stamp record PS_m is initialized in DB_v for IDA including the path-stamp serial number m, IDA, PSA's identification $IDPSA$, and PSP.

7-12 First Location-Stamp Computation. The first location-stamp $LS_{m,1}$ of PS_m is computed as follows. PSA authenticates IDA, with uniqueness and timeliness guarantees [10]. Then, PSA requests the location of IDA to LS, who sends her back l_1. PSA gets the last published linking information in DB_u. It corresponds to some certain location-stamp LS_p with serial number p. So, its published linking record is (p, R_p, L_p). This information (p, R_p, L_p) is used to compute L_1, the linking information of $LS_{m,1}$ location-stamp. PSA computes L_1 and builds record R_1. $n(1)$ is $LS_{m,1}$'s serial number ($LS_{m,1} \equiv LS_{n(1)}$), t_1 is the time when it is computed. Afterwards, she computes S_1, which is the signature over R_1.

$$L_1 := (R_0, H(L_0)) \equiv (R_p, H(L_p))$$
$$R_\sharp := (m, IDA, IDPSA, PSP);\ \ r_1 := (n(1), l_1, t_1)$$
$$R_1 := (R_\sharp, r_1, L_1);\ \ S_1 := sig_{PSA}(R_1);\ LS_{m,1} := (R_1, S_1)$$

PS_m record in DB_v is updated with $LS_{m,1}$. $LS_{m,1}$ is also sent to $PSPEA$ who verifies it in following steps $13 - 17$.

13-16 First Location-Stamp Verification. First, $PSPEA$ requests to LS to locate IDA. LS sends her back l'_1 at t'_1. Then, $PSPEA$ gets LS'_p, the last published linking information at t'_1 from DB_u, and PS'_m from DB_v. $PSPEA$ verifies

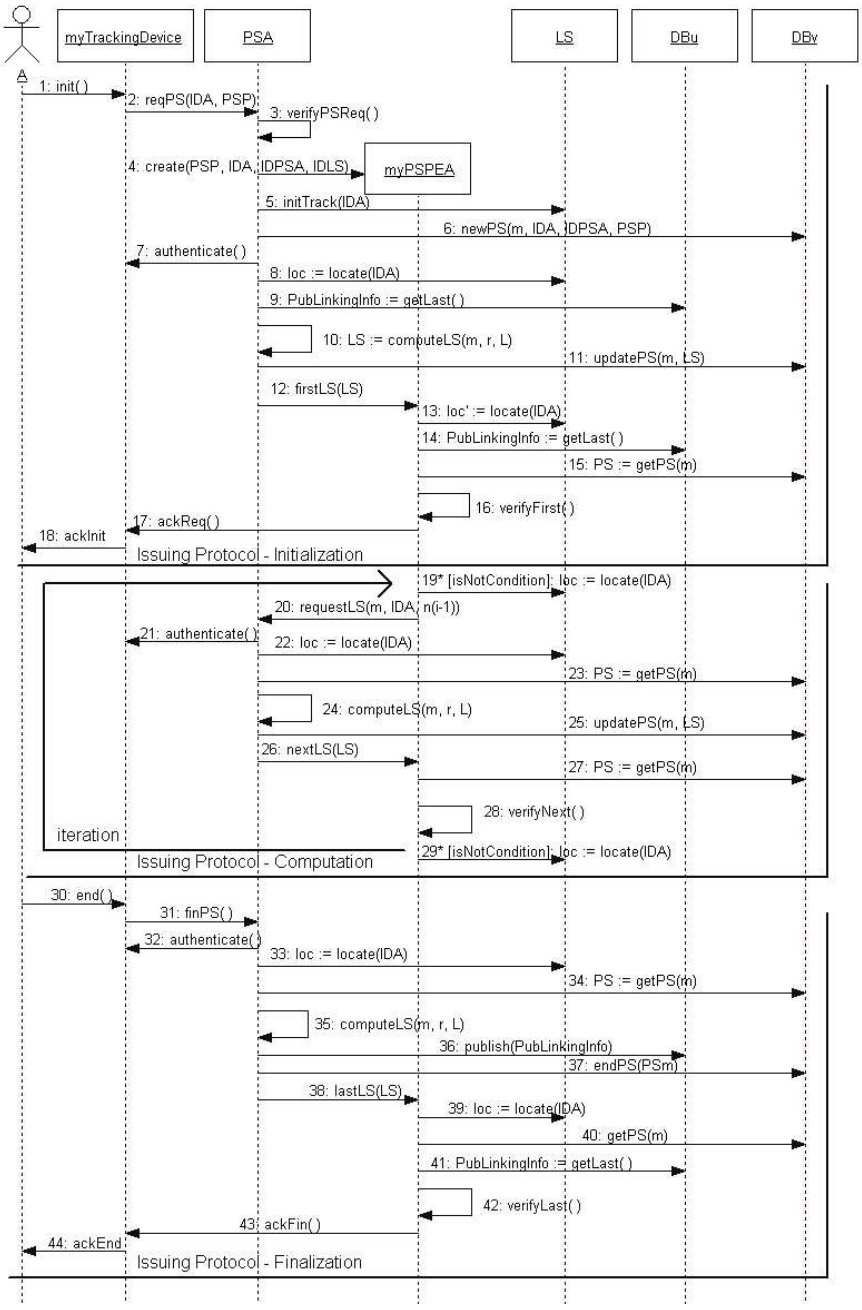

Fig. 3. Path-Stamping Issuing Protocol

that differences between location and time values (l_1, t_1) included in the received location-stamp $LS_{m,1}$, and (l'_1, t'_1) are less than certain values ϵ_l and ϵ_t defined in PSP. $PSPEA$ stores m and data from $LS_{m,1}$ that will be needed in following verifications of the next location-stamps. $PSPEA$ verifies that the linking information $(R_0,\ H(L_0))$ included in $LS_{m,1}$ is the same that the linking information in LS'_p obtained from DB_u. H is a hash function. Then, $PSPEA$ verifies the signatures over PSP and that this PSP corresponds to one of the path-stamping policies securely bound to it. $PSPEA$ verifies also that S_1 corresponds to the signature by PSA over R_1. Afterwards, $PSPEA$ verifies that $LS_{m,1}$ is included in PS_m record obtained from DB_v.

17-18 Path-Stamp Request and Initialization Acknowledge. If every verification step has succeeded, $PSPEA$ sends IDA an acknowledge of the path-stamp initialization. Otherwise, $PSPEA$ requests another initialization. If this second one fails, $PSPEA$ sends an error message to IDA and B, and asks PSA to reflect it in DB_v.

Path-Stamping Issuing Protocol: Computation

19-20 Path-Stamp Policy Enforcement. $PSPEA$ requests from time to time IDA's location. With this data and some other needed information, e.g. time, $PSPEA$ enforces the conditions in PSP that trigger the computation of the next location-stamp $LS_{m,i}$ to be included in PS_m. At that moment, $PSPEA$ requests to PSA the issuing of a new location-stamp keeping (l_i, t_i) to compare afterwards them with values in $LS_{m,i}$.

21-26 Location-Stamp Computation. Steps from 7 to 12 are repeated (correspond to steps from 21 to 26 in Fig. 3) with minor changes for every new location-stamp that must be included in PS_m. In (23) the linking information $L_i = (R_{i-1}, H(L_{i-1}))$, is obtained from $LS_{m,(i-1)}$, the last location-stamp issued for PS_m in contrast to step 9.

27-28 Location-Stamp Verification. With some differences from steps 15−16, $PSPEA$ verifies each of the issued location-stamps. $PSPEA$ compares the stored m value with the serial number included in received $LS_{m,i}$ and verifies that possible location and time differences are within ϵ_l and ϵ_t intervals. $PSPEA$ verifies that the linking information in received $LS_{m,i}$ is correctly computed from data of previous location-stamp (stored in last verification, step 16). $PSPEA$ verifies also PSP and that PS_m has been updated with $LS_{m,i}$. Otherwise, the issuing of another location-stamp is requested.
Step 13 is not considered in this phase because, in this case, $PSPEA$ already knows which location and time values should be considered. Step 14, in a similar manner, is also omitted because $PSPEA$ already knows which linking information should be used. Acknowledge to IDA is also omitted until finalization phase.

Path-Stamping Issuing Protocol: Finalization

29-31 Path-Stamp Finalization Request. The process described in the computation phase repeats until entity A decides to finalize her tracking and sends a finalization request to PSA via her tracking device.

32-38 Last Location-Stamp Computation and Path-Stamp Finalization. PSA computes $LS_{m,k}$, the last location-stamp comprised in PS_m. Then, PSA publishes $(n(k), R_{n(k)}, L_{n(k)})$ in DB_u. Finally, PSA ends PS_m path-stamp publishing it in DB_v, and also sends it to $PSPEA$.

39-42 Last Location-Stamp and Path-Stamp Verification. $PSPEA$ verifies the published final path-stamp in a similar way that in the other phases.

43-44 Path-Stamp Finalization Acknowledge. If every verification succeeds, $PSPEA$ sends IDA an acknowledge of the path-stamp finalization. Otherwise, $PSPEA$ should request another path-stamp finalization. If this second one fails, $PSPEA$ must send an error message to IDA and B, and ask PSA to reflect it in DB_v.

Path-Stamping Verification Protocol In order to verify a whole path-stamp PS_m, the verifier first has to validate all the signatures S_i. Then, he has to request to PSA the published linking information of location-stamps with serial numbers p and $n(k)$. The verifier has to generate the linking information of each location-stamp of PS_m. He has to use the published linking data of location-stamp LS_p to compute the first one, and follow the verification of the linking chain using data from the path-stamp. The verifier has to compare the calculated linking values with L_i values within the received path-stamp. He also has to request to PSA the path-stamp record in DB_v, and compare it with the received path-stamp. Last, he has to verify that the calculated linking information of location-stamp $LS_{n(k)}$ has been published in DB_u.

Another issue is the verification of the Path-Stamp Policy (PSP) enforcement. For this problem, we propose that the $PSPEA$ 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 which it 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 availability. 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. Chen, G., Kotz, D.: A Survey of Context-Aware Mobile Computing Research. Dartmouth Computer Science Technical Report TR2000-381 (2000)
3. Davies, N., Gellersen, H.-W.: Beyond prototypes: Challenges in Deploying Ubiquitous Systems. IEEE Pervasive Computing 1:1 (2002) 26-35
4. Haber, S., Stornetta, W.S.: How to Time-Stamp a Digital Document. Journal of Cryptology. 3:2 (1991) 99-111
5. Hightower, J., Borriello, G.: Location Systems for Ubiquitous Computing. IEEE Computer, August 2001 (2001) 57-66
6. Just, M.: Some Timestamping Protocol Failures. In Proc. of Internet Society Symposium on Network and Distributed System Security (1998)
7. Kabatnik, M., Zugenmaier, A. Location Stamps for Digital Signatures: a New Service for Mobile Telephone Networks. In Proc. of ICN 2001, Colmar, France (2001)
8. Maniatis, P., Baker, M.: Secure History Preservation through Timeline Entanglement. In Proc. of the 11th USENIX Security Symposium 2002, San Francisco, CA, USA (2002).
9. Menezes, A.J., van Oorschot, P.C., Vanstone, S.A. Handbook of Applied Cryptography. Ed. CRC Press 1997.
10. Satyanarayanan, M.: Pervasive Computing: Vision and Challenges. IEEE Personal Communications 8:4 (2001), 10-17
11. Une, M.: The Security Evaluation of Time Stamping Schemes: The Present Situation and Studies. IMES Discussion Papers Series 2001-E-18 (2001)
12. Weiser, M. The Computer of the 21st Century. Scientific American 265:3 (1991) 66-75
13. 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)