AN EFFICIENT MECHANISM TO ENSURE LOCATION PRIVACY IN TELECOM SERVICE APPLICATIONS

Oliver Jorns[1], Sandford Bessler[1] and Rudolf Pailer[2]
[1]*Telecommunications Research Center Vienna (ftw.), Donau-City-Strasse 1, A-1220 Vienna;*
[2]*mobilkom austria AG & Co KG, Obere Donaustrasse 29, A-1220 Vienna*

Abstract: Location and presence information will provide considerable value to informa-
tion and communication services. Nevertheless, the users are still concerned
about revealing their position data especially to un-trusted third party applica-
tions. Furthermore, legal restrictions are effective in most countries that regu-
late processing of personal data and the protection of privacy in electronic
communications. In this paper we propose a novel privacy enhancement solu-
tion (PRIVES) which is targeted for location and presence services in the 3G
service architecture and uses cryptographic techniques well suited to run in
small devices with little computing and power resources. Once a user is
granted the permission to localize another user, the location server generates a
key used to create pseudonyms that are specific for the localized user. Passed
from the watcher to the location server via the application, these pseudonyms
identify both the watcher and the desired localized user at the location server,
but are opaque to the application. The paper presents architecture and proto-
cols of the proposed solution and discusses the performance increase in com-
parison with current implementations.

Key words: location privacy, pseudonyms, 3rd party applications, HMAC, Parlay-X web
service, hash value, hash chain, one-time password, presence.

1.   INTRODUCTION

The current next-generation service architecture enables the establish-
ment of a new class of service providers and independent software vendors
that design innovative services accessing the networks through open and
standardized interfaces and protocols such as OSA/Parlay, or SIP/SIMPLE.
A key factor for these services to be successful is their personalization, i.e.

the ability to take into account user's preferences, presence, location, etc. The existence of 3rd party service providers that are less trusted than the network operator on one side and the personalization need on the other side have lead to increasing privacy concerns and motivated us to perform research on efficient methods to protect user data.

Furthermore, the processing of personal data and the protection of privacy is regulated by law in most countries. The EU directive on privacy and electronic communications [1] mentions location data explicitly and establishes the following rules:

- Providers should minimize the processing of personal data and use anonymous or pseudonymous data where possible.
- Location data is listed as traffic data. Traffic data means any data processed for the purpose of the conveyance of a communication on an electronic communications network or for the billing thereof.
- Location data in the sense of traffic data means any data processed in an electronic communications network, indicating the geographic position of the terminal equipment of a user of a publicly available electronic communications service and may refer to the latitude, longitude and altitude of the user's terminal equipment, to the direction of travel, to the level of accuracy of the location information, to the identification of the network cell in which the terminal equipment is located at a certain point in time and to the time the location information was recorded. Location data other than traffic data is regulated to the same extent as traffic data.
- For the provision of value added services, the provider of a publicly available electronic communications service may process traffic data to the extent and for the duration necessary for such services, if the subscriber or user to whom the data relates has given his/her consent. Users or subscribers shall be given the possibility to withdraw their consent for the processing of traffic data at any time. The service provider must inform the users or subscribers, prior to obtaining their consent, of the type of location data which will be processed, of the purposes and duration of the processing and whether the data will be transmitted to a third party for the purpose of providing the value added service.
- Where consent of the users or subscribers has been obtained for the processing of location data, the user or subscriber must continue to have the possibility, using a simple means and free of charge, of temporarily refusing the processing of such data for each connection to the network or for each transmission of a communication.
- Location data may only be processed when it is made anonymous, or with the consent of the users or subscribers to the extent and for the duration necessary for the provision of a value added service.

Privacy requirements are also influenced by the type of the localizing application. We categorize localizing applications in 'Pull', 'Push' and 'Tracking', depending on the relationship between the localizing user, called 'watcher', and the user to be localized, called 'presentity'. These applications have in common, that the localization process is usually executed by some kind of middleware component, that may not be part of the trusted network provider's domain, but is operated by a 3rd party service or content provider. In general there is a relationship triangle between watcher, application and presentity (see Fig. 1).

Privacy enhancement technologies (PET) address four basic ISO requirements [2]: anonymity, pseudonymity, unlinkability and unobservability that are subject of a number of research projects dealing with address privacy, location privacy, service access privacy or authentication privacy [3].

In this paper we study the location privacy, an issue that arises today in all 2G and 3G mobile networks that start to offer location based services. Terminal (and user) localization is done either by the user himself, equipped with a GPS receiver or, in most cases by the network, using the signal strength of a few neighboring cells listening to the terminal. The location information (expressed for example in geographical coordinates) can be used by the localized user (push/pull applications) for example to direct a call to the nearest pharmacy or taxi, or by another user (tracking applications) who may run an application that schedules a service team. In either scenario there is an application or service that processes location data and that is owned by some third party commercial organization.

Any proposed schemes to improve privacy in mobile networks clearly have strong interoperability and standardization aspects, as they have to be approved by 3GPP or IETF. The IETF "Geographic Location/Privacy (geopriv)" Working Group has defined location privacy requirements [4] in which a Location Object (LO) plays the main role: it contains the location information to be transmitted from the Location Server entity (a part of the network operator) to the location requestor (watcher) as well as access rules for different users and is itself cryptographically protected. While this proposal is general and powerful, it implies that a large data amount has to be transmitted and requires from the watcher a lot of processing power.

As the first GSM and UMTS networks start to offer location based services, the 3GPP has tried to improve privacy by tightening the access control of users and applications on location information. Thus, in a privacy enhancement specification for UMTS Release 6 [5], a complete authorization relationship between three entities: requestor (watcher), application and presentity has to be defined in the Telecom Server in order to allow access to location information.
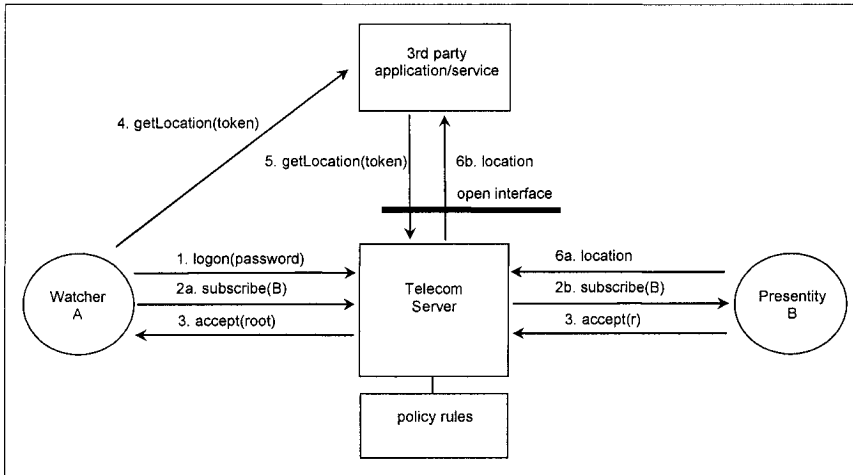
Figure 1. Architecture overview

The responsibility of the Telecom Server in Fig. 1 is to check the access rights of watcher and application (also for non-repudiation and monitoring reasons) and leads to complex processing, very large database tables and a tight coupling of all participating entities. The protection of the presentity identity is handled vaguely by proposing the use of aliases.

To overcome this situation, the approach described by Hauser et al.[6] can be used. It is based on pseudonyms which are exchanged between the watcher and the telecom service in order to make it impossible for 3rd party services to track the location of a certain presentity, store the location history or aggregate information from several services and create a profile. The difficulty in applying these schemes in practice arises from the use of public key infrastructures for signature, encryption and decryption processes which are computationally too expensive to be executed on today's mobile terminals. These considerations have motivated us to propose a more efficient Privacy Enhancement Scheme (PRIVES)[1] to be used with the architecture in Fig. 1 to protect the identity of localized users.

The rest of the paper is organized as follows: section 2 describes the architecture and the generic interactions between watcher, presentity, Telecom Server and application. Section 3 describes the mechanism used to generate pseudonyms at the watcher and at the Telecom Server. Section 4 gives preliminary implementation and performance results and section 5 concludes

---

[1] Verfahren zum Unwandeln von Target-Ortsinformation in Mehrwertinformation, pending Patent Nr.: A 363/2004

with future extensions and applicability of the proposed scheme in other scenarios.

2.   SERVICE INTERACTIONS

In this section we describe shortly the service interactions between the system entities in Fig. 1: watcher, presentity, telecom server and 3rd party application. Basically, the watcher starts by establishing a trust relation with the presentity using a subscription/notify message pattern. We assume that the presentity accepts a subscription to his/her location information only, if the watcher is known and trusted. Alternatively, rules may be predefined and stored in form of policies in the Telecom Server. Both approaches can be combined with a group management system.

Subscription/acceptance messages precede authentication of the users to the Telecom Server. Other messages are needed to query the status of subscriptions, which are stored and forwarded when the users go online:

- `getBuddies()` returns the list of subscribed and accepted presentities
- `getPendingWatchers()` returns the list of watchers waiting an accept message for that presentity
- `getPendingSubscriptions()` returns the list of presentities that did not send an accept so far.

Returning to the general operation in Fig. 1, the accept message is mediated by the server which calculates a "root" value r and sends it to the watcher (step 3). The root is the initial value of a chain of one-time passwords (pseudonyms, tokens) that are subsequently sent in localization requests to the 3rd party application. These pseudonyms are used to identify the presentity in the (standardized) API methods between the 3rd party application and the Telecom Server (see Fig. 1, step 5). The pseudonyms sent by the application to the Telecom Server are used to authenticate and authorize the localization request and to retrieve the real user identity. The Telecom Server then retrieves presence or location information of the presentity and returns this data to the application.

3.   USE OF HASH VALUES FOR AUTHENTICATION
AND AUTHORIZATION

In order to reduce the calculation complexity on the watchers's and the presentity's device, we propose PRIVES, a scheme which authenticates and authorizes the watcher on the basis of hash values.

A hash value is the result of a hash function H which is a one-way function that it is easy to compute but computationally infeasible to invert. A hash function $y = H(x)$ is defined as a function for which the effort of computing x, given the output y consisting of n bit, is 2^{n-1}. Hash functions are also required to be collision resistant, that means, finding two different inputs x and x' such that $H(x) = H(x')$ requires an effort of $2^{n/2}$. The most commonly used hash functions are MD5 [7] by Ronald Rivest and SHA-1 [8] by NIST.

For repeated interactions such as requesting location information periodically, we use hash values that are calculated from the previous hash value. The idea to use hash values for authentication which are based on a chain of computations of hash values was first published by Lamport [9]. Starting with the shared secret the first computed hash value is used as input for the calculation of the next hash value and so on. After the client received n (number of hashes to compute) it calculates the n-1 hash value (h^{n-1}) on the basis of the shared secret and sends it back to the server. The server calculates the next hash value of this received one ($h(h_{n-1})$) and compares it with the nth computed hash value computed on the basis of the shared secret (h^n(shared_secret)). Equality of these values proves that the client owns the right shared secret. Now, the server decrements the value of n by one and stores this value. The next time the client authenticates, the server receives the hash value h^{n-2} (see Fig. 2).
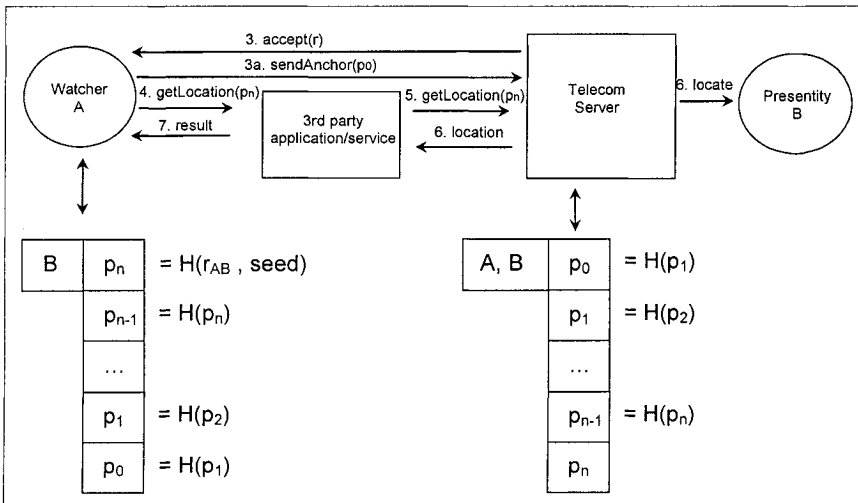

Figure 2: System Architecture based on Lamports Hash chains

The strength of security is reached because of the fundamental one-way property of hash values. The problem with a hash chain is that it has to be used in the reverse direction, i.e. first the nth value, then the n-1 value, etc., implying that n-values have to be generated first - a complex operation in small devices.

The One-Time Password (OTP) System [10] is similar to the solution of Lamport. Although both guarantee a high level of security, they fail in practical implementations because OTPs have to be used in reverse order of their creation, that is the ith OTP p_i is the $(n-i)^{th}$ value of the hash chain which is not feasible on mobile devices as our measurement results clearly show.

This means that all OTPs could be computed at once and stored in the user's terminal, which is probably infeasible in mobile devices that usually have only a very restricted amount of memory available.

Alternatively the OTP could be computed on demand from the root r, meaning that for every OTP the hash chain has to be partially rebuilt. On demand computation of the OTP p_i requires n-i applications of the hash function with the result that for using all OTPs of a hash chain of length n, a mobile device would have to compute hashes $n*(n+1)/2$ times. The computational effort of calculating one OTP has the order $O(n)$ and the on demand calculation of all OTPs of a hash chain of length n increases with the order $O(n^2)$, which also restricts the applicability on mobile device platforms.

We use in PRIVES the keyed hash scheme called HMAC [11] that overcomes the computational problems of the former schemes and still guarantees high security without the need of computation of a large number of hash values in advance. HMAC is based on MD5 or SHA-1 and allows us to create a hash value from a previous one, using in addition the password as a secret key shared between the watcher and the Telecom Server.

3.1 Hash-chain generation and initialization

The only information watcher A needs is the root r_{AB}, which is a random number initially created by the Telecom Server after a successful subscription. All messages are synchronous (request - response), so that watchers have to refresh periodically the subscription status. As shown in Fig. 3, the hash values are created in parallel by the watcher and the Telecom Server from the previous value h_{n-1} and watcher's password by applying the HMAC function.
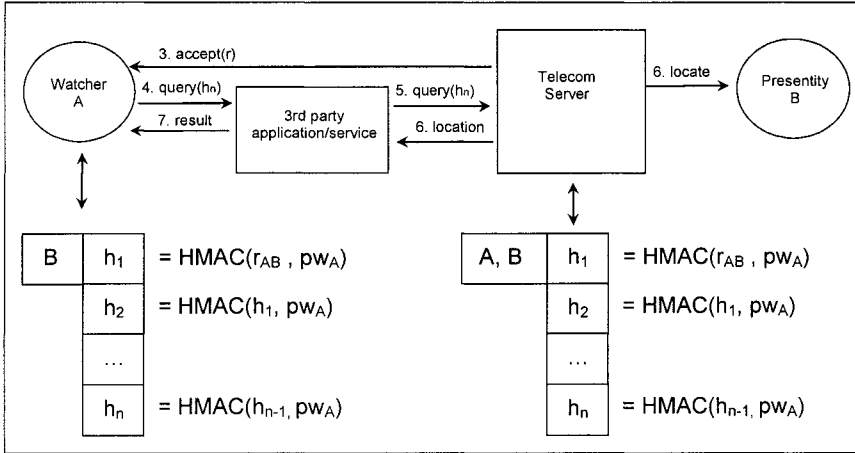
Figure 3. Creation of chained hash values based on HMAC

After subscription, the watcher sends the first hash value as argument of a location request to the 3rd party application. The request is forwarded via the standardized interface to the Telecom Server where the real identity of the presentity and its location is determined.

The Telecom Server maintains a hashed access control list, such that exactly one access to a database table is required to resolve authorization rights, whereas an implementation based on standard procedures would have to first fetch the presentity data from a user table and then to look up the watcher (and/or application) in an 'authorized watcher' table and to check the allowed operations. Thus, the performance gain for the access control look-up is at least a factor of 2.

Finally, if authorizations have to be revoked, the Telecom Server has just to delete the stored hash value from the access control list, leading to a new subscription at the watcher by the time the next localization requests arrives.

3.2 Protocol error handling

The correct operation of PRIVES relies on the synchronization of hash value creation at watcher and server side. However, in case of erroneous transmission channel, unexpected crashes of client applications due to loss of battery power, synchronization is lost.

Therefore, each hash value has to be stored persistently on the user's device after a location request. Each time the application starts, the stored hash value and the user's password are used for calculation of subsequent hash values.

If for any reason the Telecom Server cannot process a received hash value, it returns coordinates (0, 0) as result, which indicates an error. To recover from the error, the client sends a new `subscribe` message for the respective presentity. Since the watcher subscribed this presentity already, the server computes and issues a new root which reinitializes the hash values and allows the watcher to request the position again.

The use of hash values in PRIVES increases also the efficiency of the Telecom Server that in normal case has to check if the watcher, and the 3rd party application are authorized to locate the presentity .

4.　　　PROTOTYPE REALIZATION

In order to validate the architecture and measure the performance of the system, we implemented in the lab a "Telecom Server" that obtains real location information from a Parlay-X location service. Thus, several mobile phones (the presentities) can be localized. The watchers are currently implemented on a J2ME Personal Profile 1.0 [12] platform. A simple application is being currently implemented for demonstration purposes: it tracks the user and calculates periodically the distance between successive retrieved locations of that user. The scenario is a bit more complex than that described in Fig. 3: it starts with the watcher (which is identical with the presentity in this case) sending the application one single startRoute() request with the pseudonym as parameter. The application requests from the Telecom Server to be notified with the new position of the user every T minutes (using the pseudonym as parameter). This architecture has the advantage that the application can communicate much more efficiently with the Telecom Server than the wireless client and it relieves the client from transmitting a high amount of data. When the terminal leaves a certain geographical area, the total (direct) distance is summed up, the user is notified and the service session terminates.

4.1　　　Performance and security considerations

The performance gain is determined by the following operations:
- Hash value calculation at watcher and server
- Checking a certain hash-value at the server

One reason for using hash techniques is their computational efficiency that makes them suitable for today's wide spread mobile devices. Our performance measurements were undertaken for MD5, SHA-1, MD5/HMAC and SHA-1/HMAC, carried out on a mobile emulator of the J2ME Wireless Toolkit 2.0 with preset VM speed emulation of 100 byte codes/millisecond.

All results in Table 1 and Table 2 are mean values based upon calculation of 100 hash values. Our performance analysis first concentrated on an implementation of the OTP (one time password) System designed to allow only up to n=10 authentications before the system needs to be reinitialized. This would require n*(n+1)/2=55 hash value calculations (see Table 1).

Table 1. Mean time in ms for single hash value calculations

function	mean time
MD5	51ms
SHA-1	139ms
MD5/HMAC	164.1ms (0.164sec)
SHA-1/HMAC	448ms (0.448sec)

To keep computing effort low, n has to be small which however results in frequent re-initializations. From Table 2 we see that an implementation based on chained hash calculations as it is done in the OTP System is not feasible on mobile devices with low processing power.

The calculations based on MD5/HMAC and SHA-1/HMAC take longer than those based on corresponding MD5 and SHA-1, but since the hash value can be safely calculated from a previous one (n=1), the HMAC procedure is fast enough for our purposes.

Table 2. One-Time Password scheme and expected mean time needed for calculation

function	Total computation time for 10 authentications	Total computation time for 100 authentications
MD5	2190ms (2.19sec)	205003ms (~3.4min)
SHA-1	6677ms (6.677sec)	701950ms (~11.7min)

Our performance analysis first concentrated on an implementation of the hash chain scheme that calculates OTPs in reverse order. The implementation does not store the calculated hash values, but has to partially rebuild the hash chain for each authentication. Storing n 128 bit (MD5) or 160 bit (SHA-1) values for each buddy requires a memory amount that may not be available on every today's mobile devices. A preset hash chain length of n=10 allows 10 authentications before re-initialization. The first authentication requires the computation of 10 hash values, the next 9 and so on. This means that the largest delay for getting the hash chain is determined by the first authentication and grows linearly with n.

To keep computing effort low and authentication delays acceptable, n has to be small which however results in frequent re-initializations. Our calculations show that an implementation based on reverse chained hash calculations (without storing the hash chain) is not feasible on mobile devices with low processing power.

The time needed to calculate a hash value based on MD5/HMAC and SHA-1/HMAC takes longer than with MD5 and SHA-1 (see Table 1). But since only one HMAC value has to be calculated per authentication and each hash value can be safely calculated from a previous one (n=1), HMAC is better suited. We also see from Table 1 that the HMAC calculation takes about three times longer than the underlying hash function. This means that for n=4 the HMAC scheme is already faster than the reverse order scheme.

From a security point of view HMAC is secure enough given the (not so high) sensitivity of the data. If higher secrecy of data is required, SHA-1/HMAC should be preferred over MD5/HMAC, since collisions in compressing functions of MD5 have already been found [12]. In case processing power is critical, MD5/HMAC will be the better choice because it is computed approximately three times as fast as SHA-1/HMAC.

In general, it is not possible for an intruder to calculate easily hash values by eavesdropping the previous HMAC hash value. Examinations on HMAC in [11] show that finding a collision (guessing the input values that result in the same hash value) for hash values of l =128 bit length would require $2^{l/2}$ messages for each given key. It can be assumed that this is improbable to occur. As [11] further states, using the same password an attack would require approximately 250.000 years.

5.   CONCLUSIONS AND FURTHER RESEARCH

In this work we propose PRIVES, a scheme that allows a third party application to receive and process user location information from a network operator without being able to identify the localized user. Monitoring users, building of location profiles or aggregation across different applications becomes impossible, a fact that would increase the user acceptance for location based services. Since the presentity grants the watcher the access to location information no additional security responsibility has to be taken by the LBS to check the authorization of the watcher. Furthermore, PRIVES retrieves data tuples from access control lists in an efficient way.

As further research directions, we will investigate, whether PRIVES can be extended to other location operations, such as triggered location or periodical notifications, or to other privacy relevant services, like presence.

6.   ACKNOWLEDGEMENT

This work has been done at the Telecommunications Research Center Vienna (http://www.ftw.at) within the project "Service Platforms beyond OSA", and partially funded by the Austrian Kplus Program.

REFERENCES

1   Directive 2002/58/EC of the European Parliament and of the Council concerning the proc-
    essing of personal data and the protection of privacy in the electronic communications sec-
    tor (Directive on privacy and electronic communications), 12 July 2002, Official Journal
    of the European Communities L 201/37
2   "Common Criteria for Information Technology Security Evaluation , Part 2: *Security func-
    tional requirements*, January 2004, Version 2.2, CCIMB-2004-01-002, aligned with ISO
    15408, http://www.commoncriteria.de/it-security_english/ccinfo.htm
3   RAPID - Work Package 2 - Stream 6: *PETs in Infrastructure*, FP5 IST Roadmap project,
    RAPID – IST-2001-38310: Roadmap for Advanced Research in Privacy and Identity
    Management, http://www.ra-pid.org/
4   J. Cuellar, J. Morris, D. Mullignn, J. Peterson and J. Polk, Geopriv Requirements, IETF
    RFC 3693, Feb. 2004
5   3GPP, *Enhanced User Support for Privacy in Location Services*, 3GPP TR 23871 V 5.0.0
6   C. Hauser, M. Kabatnik, *Towards Privacy Support in a Global Location Service*, Proceed-
    ings of the IFIP Workshop on IP and ATM Traffic Management (WATM/EUNICE 2001),
    Paris, 2001
7   Ronald L. Rivest: *The MD5 Message-Digest Algorithm. RFC 1321*, April, 1992.
8   National Institute of Standards and Technology: *Secure Hash Standard*, Federal Informa-
    tion Processing Standards (FIPS) Publication 180-2, 2002
9   Leslie Lamport, *Password Authentication with Insecure Communication*, Communications
    of the ACM, vol. 24(11), 1981, pp. 770-772.
10 N. Haller, C. Metz, P. Nesser, M. Straw, *A One-Time Password System*, RFC 2289, 1998
11 Mihir Bellare, Ran Canetti, Hugo Krawczyk, Message Authentication using Hash Func-
    tions   –   The   HMAC   Construction,   CryptoBytes,   Vol.   2,   No.   1,   1996,
    http://www.cs.ucsd.edu/users/mihir/papers/hmac-cb.pdf
12 Hans Dobbertin: *Cryptoanalysis of MD5 Compress*, Announcement on Internet, May,
    1996, http://citeseer.ist.psu.edu/dobbertin96cryptanalysis.html
13 J2ME Personal Profile, Version 1.0, http://java.sun.com/products/personalprofile/index.jsp