PP2db: A Privacy-Preserving, P2P-Based Scalable Storage System for Mobile Networks

Manuel Crotti, Diego Ferri, Francesco Gringoli,
Manuel Peli, and Luca Salgarelli*

University of Brescia - Italy
{firstname.lastname}@ing.unibs.it

Abstract. Reputation-based systems that handle millions of users face the problem of simultaneously supporting privacy and trust in an efficient way. In order to scale, often existing systems either sacrifice privacy to preserve trust, or vice versa. The introduction of advanced cryptographic techniques such as Group Signatures might offer a solution, but their applicability to large, distributed systems such as P2P-based ones has yet to be proved. In this paper we introduce PP2db, a privacy-preserving, scalable and distributed storage system targeted at mobile networks, specifically designed to support the anonymous but trusted exchange of Quality of Experience (QoE) information. In such case-study, QoE data is exchanged among users so as to make informed decisions on which network to select at any given time. We demonstrate that by enriching a P2P database with Group Signatures it is possible to create distributed storage mechanisms that guarantee privacy-preserving features, while offering strong trust at the group level. Furthermore, we demonstrate that the resulting architecture can scale in a realistic mobile network scenario to handle millions of users.

Keywords: Trust, anonymity, secure P2P, databases, mobile networks.

1 Introduction

Reputation-based systems have been recently proposed to drive the deployment of next-generation mobile communication services, where users with multi-interface terminals dynamically select the best available network service based on the evaluation of historical Quality of Experience (QoE) data, saved by the community [1]. QoE is an indication of how well the system meets the end user's needs, providing a measure of the end-to-end performance at the service level from the end user's perspective [2].

Two major building blocks are at the base of such vision: a storage system for historical QoE data that can scale to millions of users, typical of modern wide-area mobile networks; and a mechanism that while protecting the user's privacy when posting relevant QoE data to the community, guarantees that only people belonging to the community itself can indeed provide such data. The last issue

* This work was funded in part by the E.U. FP7 project "PERIMETER".

M. Rajarajan et al. (Eds.): SecureComm 2011, LNICST 96, pp. 533–542, 2012.

is almost an oxymoron, expressing the need for two colliding requirements: on one hand, protecting the user's privacy, possibly through anonymization, while on the other hand ensuring the community that the QoE data provided by each user can be trusted, therefore requiring some form of identification.

In this paper we present the design and evaluation of PP2db – Privacy-Preserving, Peer-to-Peer (PP^2) distributed Data Base: a mobile, distributed storage system for QoE data with privacy preservation features that aims at solving the issues described above. Besides defining the general architecture, we analyze its scalability, showing how it can scale to millions of users, making it applicable to current and future mobile networks.

While we designed PP2db with QoE-based mobile networks in mind, its flexible, scalable, P2P-based architecture makes it amenable to different applications, wherever large communities share data that needs to be trusted, while preserving the privacy of the users. Therefore PP2db can easily find applications in fields such as social networking, community networks, Internet of Things, and any large scale feedback-based application. We make our software freely available for download under an Open Source license at [3].

The rest of this paper is organized as follows. Section 2 describes the high-level requirements and goals pursued by our architecture. Section 3 introduces the two main building blocks which we used to design and implement PP2db. Section 4 describes the PP2db architecture internals, while we analyze its scalability in Section 5. Finally, Section 6 concludes the paper.

2 Rationale, Goals and Design Choices

The mobile telecommunication market is very diverse in its offerings to final users. Consumers can choose among large sets of service providers, technological means (WiFi, UMTS in its many incarnations, second generation technologies, etc.), subscription plans and add-on services. The recent introduction of community networks makes the selection of the "right" service even more complex, especially in large cities.

Several research projects have proposed in the recent past approaches to solve this issue by letting end users make informed decisions through their multi-interface devices so as to be always best connected: one example can be found in [4] or, more recently, in [1]. Most of these technologies require users to share with other users information about their Quality of Experience, i.e., indication of how well each service met their specific needs. Through the analysis of QoE data made available by other users, terminals should then be able to automate their service-selection process, always obtaining the service that better suits their needs.

Such an infrastructure to store and share QoE data must satisfy several high-level requirements. The storage system should be **scalable** enough to handle the growing numbers of mobile users, where a regional service must sustain tens of millions of users. Feedback collected from users should be **protected from pollution**: for example, a malicious network operator should not be able to alter

in their favor QoE data in order to bias the users' choices. Finally, the **privacy of end users should be preserved**, both against operators and other users. In fact, posting QoE data together with identity-related information could open the door to retaliation from network operators, in case the QoE feedback is negative with respect to the services they offer. It could also expose private details: for example, stating that "the free WiFi service under the Eiffel tower in Paris is very good on Fridays" would expose one's location at a specific time.

The first and second requirements call for a distributed, scalable storage architecture. The architecture we propose is based on a Peer-to-Peer storage system.

The second and third requirements are in conflict with each other: while some form of identification is necessary to satisfy the second requirement, anonymization is paramount to achieve the third one. In order to strike a balance between the two, PP2db relies on the use of a somewhat recent set of cryptographic authentication techniques, called Group Signatures.

In the following Section we briefly describe the basics of these two fundamental building blocks we used in designing PP2db.

3 Background

3.1 XPeer

XPeer [5] is a P2P distributed database based on a hybrid P2P architecture. In XPeer data is stored and managed in XML format, and can be retrieved using XQuery [6]. The system automatically redistributes the workload over its overlay network by means of self-organizing algorithms, therefore providing for scalability.

The overlay network of the XPeer system is a tree structure. The leaf nodes are called *peers* and the inner nodes are called *superpeers*. Each peer stores a portion of the distributed database in XML format and each superpeer stores indexes for data retrieving. Figure 1 (left) depicts a two-levels overlay network. Peers, after registering with a superpeer, retrieve data following two steps: query compilation and query execution.

The **query compilation** phase involves a peer $P1$ that submits a query to superpeer $SP1$, which in turn returns to $P1$ the list of peers that possess the

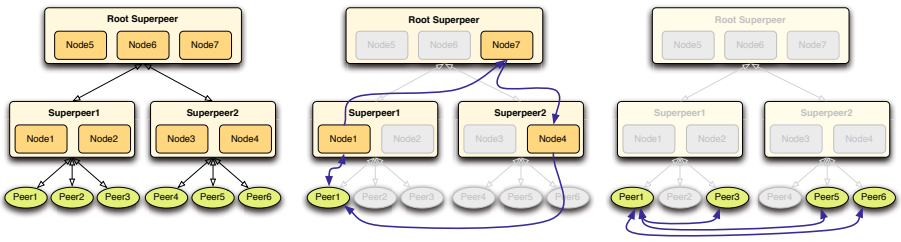

Fig. 1. XPeer Overlay Network (left), query compilation (center), query execution (right)

requested data. At this point $P1$ starts **executing the query**, which is split into sub-queries that are sent to the corresponding peers Pi. Once the results arrive, $P1$ joins them generating the requested response.

Each peer stores and maintains its own data: whenever a change is committed to the local XML, the peer sends a TreeGuide update message to the superpeer it is connected to, so that the relevant indexing information can be updated in the entire XPeer overlay.

3.2 Group Signatures

A *group signature scheme* (GS) is a relatively new digital signature scheme with enhanced privacy features [7]. Only *group members* can sign messages anonymously on behalf of the group, each one using a private and non disclosable *member secret key* ($\mathcal{MSK}$). On the contrary, everyone having access to the *group public key* ($\mathcal{GPK}$) can verify the validity of the produced *group signatures*. A trusted *group manager* holds the *group secret key* ($\mathcal{GSK}$), and is responsible for setting up the group, adding new members, revoking their membership, etc.

The first *Provably–Secure* and *Dynamic* GS scheme was introduced by Ateniese et al. [8]. From here on we will simply refer to it as ACJT. After ACJT, *Membership Revocation* received a good deal of attention. One of the most popular techniques that offer this capability has been consolidated by Camenisch and Groth [9] and we will refer to it as CG in the following.

4 The PP2db Architecture

PP2db realizes a highly scalable, distributed storage architecture with privacy-preserving capabilities, amenable by design to support the sharing of QoE-information as described in Section 2.

A schematic representation of PP2db is given in Figure 2 (left). The P2P network at the center of the picture is accessed by users in a given group that either want to 1) anonymously upload new group data or 2) collect existing records and verify that they have been inserted by authorized users belonging to the same group. To this end the information in the P2P network is made of entangled pairs of (record, signature): thanks to this approach anyone can verify and trust a record; furthermore if, for some reason, a legitimate record is found as evidence of a user's misbehaving, the identity of the posting user can be eventually disclosed by the group manager after "opening" the associated group signature.

In PP2db the nodes that build the P2P overlay do not take part of any users' group, i.e., a SP can not add data to the network even if it is compromised. For the same reason we do not consider in the following analysis the way SPs are connected and how they mutually authenticate, and we exclude the overheads due to this kind of traffic from our investigation.

A peer connects to PP2db via a *power-up signaling* procedure. Once connected, the it can carry out the following operations, as shown in Figure 2 (left):

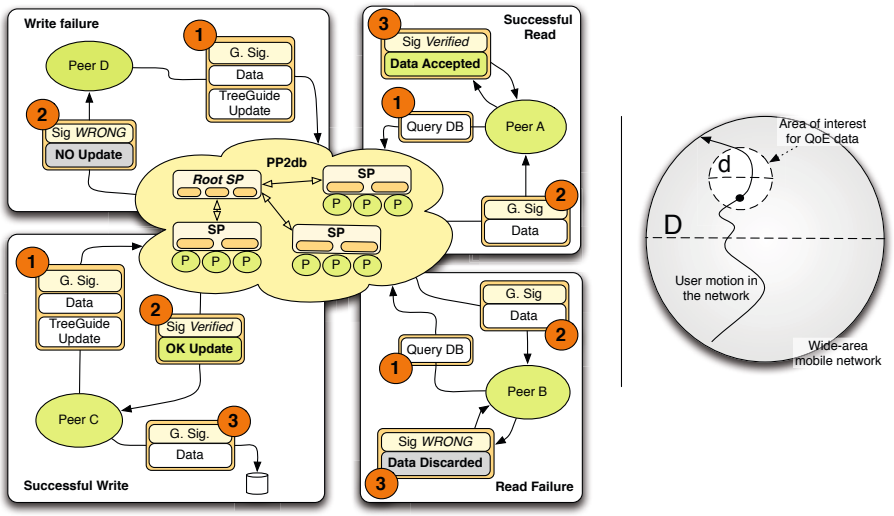

Fig. 2. Left: Schematic model of the PP2db architecture. – **Right**: QoE-enabled mobile network model adopted for PP2db.

READ. Peer A (upper right) queries the PP2db network for a given record/group, executing a *PP2db MetaSearch* operation and receiving back *PP2db Meta-SearchAnswer messages*. After following the procedure outlined in Section 3.1, the network returns (record, signature) pairs within *PP2db QueryResult messages*. The Peer will keep the data only if the group signature is valid, discarding it otherwise (lower right).

WRITE. Peer C (lower left) signs a message with its private group key and sends the pair (record, signature), together with the group name and the TreeGuide update, to a SP through the *PP2db MetaUpdate* procedure. Upon successful verification of the group signature, the SP makes the data available to the PP2db network, otherwise the TreeGuide update is discarded (upper left).

4.1 XPeer-Based Storage Module

The storage module of PP2db is based on XPeer for the collection and sharing of QoE data. Feedback coming from participating user terminals is converted in XML and stored in the local PP2db database of the same terminals. Each of the QoE fields that will be shared with community users is paired with its corresponding group signature that is stored in a properly reserved field of the XML schema. After each change of the local database, a TreeGuide update message is spread through the XPeer network in order to update the tree-guide of superpeers [5].

Data can be retrieved through XQuery (see Section 3.1 for details) and a user can query the XPeer network asking for a list of service providers that meets her quality standards (e.g., "PRIVACY_RATING=SECURE and not COST_RATING=EXPENSIVE").

4.2 Combining XPeer with Group Signatures

For PP2db we developed a new framework that integrates group signature services and we made it available under an Open Source license [3]. We chose Java because i) it is straightforwardly enabled on the majority of platforms that support Sun's Java Virtual Machine ii) XPeer is Java based and iii) no similar framework was released before. We opted to not bind it to any specific group signature scheme, so that new schemes can be easily adapted: for this paper we integrated the two aforementioned schemes ACJT and CG. The framework was designed to extend the Java Cryptography Architecture (JCA) [10]. To this end we reviewed the GS schemes that cannot for construction be mapped to the services already offered by the JCA and we implemented two different protocols: one for "signature" operations (**sign**, **verify**, **open**); another for group "maintenance" operations (**setup**, **join**, **revoke**).

We then reviewed the codebase of XPeer and we changed the way data is read and written: for each write operation a signature is added by the client that is pushing the data; the same signature will be verified against the group certificate for each following read operation.

5 Performance of PP2db: Scalability

The scalability of PP2db is affected by two main factors: the amount of computational resources required by each terminal that participates in the system, and the amount of network traffic generated by the architecture while in use. We start by describing the reference scenario for our evaluation, which we derived from the QoE-enabled mobile network architecture defined in [1].

5.1 Application of PP2db to a Mobile Network Scenario: A QoE-Enabled Mobile Network

Our scenario models a QoE-enabled wide area mobile network, albeit with some simplifications. We assume that users are uniformly distributed in a circle of diameter D, as shown in Figure 2 (right). While in this circle, users are randomly moving of uniform motion.

Every time a terminal powers up, its PP2db instance connects to a superpeer and gets access to the distributed storage system. Users are regularly asked to rate their mobile service experience. Such ratings are injected into PP2db in the form of Quality of Experience (QoE) reports, which represent a moving average of the ratings expressed by the user for a given service at a given location over time.

At the same time, terminals collect through PP2db both past and present QoE reports relevant for their location and produced by other users in the same geographic area. We define this "area of interest" as a circle of radius d and the user's location as the centre of this smaller circle. With this information, users are able to decide which connection is best suited to their needs, for example, whether to prefer lower price or higher reliability.

We imagine for our analysis a very simple PP2db scenario, where the hierarchical tree is made of one *Root* superpeer on top of a single layer of superpeers. All peers are then clustered around them. In accordance with our model, both peers and superpeers are uniformely distributed in the network.

Table 1. Network and system parameters used in our PP2db mobile network scenario

Symbol	*Description*	*Value*
P, SP, R	Peer, superpeer, Root	
K	# of SP whose father is R	100
D	Network diameter	1000 (Km)
$d(\leq D)$	Diameter of network portion of interest to current user for QoE purposes	1 (Km)
N	Total number of users in the mobile network	120 (Million)
P_a	Fraction of active users (powered-on terminals)	50%
n	Active users in area of interest d	60
T	Interval between two subsequent QoE writes	60 (s)
h	Fraction of users in area of interest that have relevant QoE data to share	5%, 20%, 50%

In Table 1 we show the symbols of the parameters we adopted for our model, and the values we assigned to them.

Several of the values we assigned to the parameters were derived using the statistical data for modern 3G networks taken from [11]. We consider in our model a relatively large country ($D = 1000km$) with four operators, each of them with 30 million subscribed users (total $N = 120M$). At any given time, half of the users are active. As we have already noted, the users are uniformely distributed in the network circle, and move of randomly uniform motion at a speed uniformly distributed between 0 and $30km/h$ ($0 - 8.3m/s$). A couple of simple formulas let us derive the number of active users in each "area of interest" at any given time ($n = 30$), and the number of powerups/downs in such area ($694 \cdot (d/D)^2$).

We conservatively assume that each user writes to the PP2db storage system updated QoE data every $T = 60sec$, regardless of where they are. Each user will request QoE data any time it crosses a new area of interest. In our experiments, we consider three cases regarding the fraction of users in each area of interest that can respond with relevant QoE data to a query. We identify such parameter with h, and consider for it values of 5, 20 and 50%.

Table 2. PP2db message size and overheads for each basic operation

Symbol	*Description*	*Value*
M_1	PP2db power-up/down signalling message size	1.5 (kB)
M_2	PP2db MetaUpdate message size	36 (kB)
M_3	PP2db MetaSearch message size	3.7 (kB)
M_4	PP2db MetaSearchAnswer message size	2 (kB)
M_5	PP2db QueryResult message size	5 (kB)
	Power up and connect to SP	$8 \cdot M_1 + 2 \cdot M_2$
	Power down	$2 \cdot M_1 + M_2$
	TreeGuide update	$2 \cdot M_1 + 2 \cdot M_2$
	Query Compilation	$(K+1) \cdot M_3 + K \cdot M_4$
	Query Execution	$h \cdot (2 \cdot M_1 + M_5)$

Messages exchanged among PP2db-enabled nodes are either for signalling or exchanging meta-data. We classify them according to the procedures defined in Section 4. We computed their size by tracing an active PP2db network in our laboratory, and report them with symbols M_1 through M_5 in Table 2. Finally, simple formulas link the parameters and message sizes expressed above to the total traffic generated by the QoE storage system for each of the read/write operations defined in Section 4.

5.2 Computational Overheads

We have analyzed all the primitives that we implemented in the PP2db Group Signature Java framework to profile their computational costs. The space allowed for this paper does not allow us to report on our findings. In extreme summary, all operations that are executed by mobile terminals are independent on the number of users in the system, therefore **PP2db can scale indefinitely in size with respect to the computational burden imposed on each mobile terminal**. Please refer to [12] for more details.

5.3 Network Traffic Overheads

The characterization of the message overheads introduced by the various group signatures schemes that PP2db implements is fully described in [12]. Here, for space constraints, we just use the numerical values that were derived in that technical report.

For group signatures we suppose that every mobile user belongs to the same GS group, because we are interested in the study of the performance of our system in the whole network of a particular mobile operator. For this evaluation, we used signatures of 1024 bits equivalent security. In terms of revocation (GC signature scheme), we considered a base value of 5% of users revoked in a solar year, i.e., 5% of the subscribers to a given operator will switch to another one each year.

Table 3. Scalability of PP2db in three privacy-preserving configurations

Privacy protection	$h = 5\%$	$h = 20\%$	$h = 50\%$
	Bandwidth Mb/s (overhead)		
PP2db with no security	1.79	1.93	2.22
PP2db with ACJT GS	1.81 (1.18%)	1.98 (2.59%)	2.30 (3.60%)
PP2db with CG GS	1.82 (1.68%)	1.98 (2.59%)	2.30 (3.60%)

Applying the message overhead of [12] to the parameters and formulas described in Tables 1 and 2, we obtain the amount of bandwidth occupied by PP2db messages in any given QoE area of interest (diameter d), as shown in Table 3.

The first result worth commenting is that the overhead introduced by PP2db alone is quite sustainable by modern mobile networking infrastructures, generating only up to 2.22 Mb/s of traffic in a network area of a 1km radius when no privacy protection scheme is activated. When 5% of users have relevant QoE data, this value drops to 1.79 Mb/s.

Quite surprisingly, such an overhead remains well under control even when privacy protection is enabled and a large fraction of users ($h = 50\%$) responds to PP2db queries: in this case the use of group signatures introduces an extra 3.60% overhead, generating 2.3 Mb/s of PP2db traffic. Note that in the case of CG, the impact of handling revocation for 5% of the users is next to null, amounting to an extra 0.5% of overhead only in the case of $h = 5\%$. Note that such sustainable bandwidth figures were obtained by forcing users to rate QoE every minute ($T = 60$sec), which is really an inflated estimate.

The system presents no other visible limits since all relations in the model are linear. Therefore, **PP2db introduces manageable network overheads when used in privacy-preserving QoE storage architectures for modern mobile networks**.

6 Conclusions

In this paper we presented PP2db, a privacy-preserving, scalable storage system for mobile networks. We designed it to support an emerging requirement of modern multi-operator, multi-interface mobile network architectures, such as the one described in [1], where there is the need to store QoE data in a scalable and privacy-preserving way, while ensuring trust at the group level.

Our analysis shows that PP2db scales quite well to support such requirements in modern mobile networks with millions of users, even when its overhead are evaluated in highly dynamic ($h = 50\%$) and densely populated environments. As far as we know, PP2db is the first system to combine strong trust at the group level through Group Signatures, anonymity and distributed storage systems in a highly scalable architecture.

Although PP2db was designed with these expressed targets in mind, its features make it amenable to many other scalable storage applications where privacy must be coupled with trust, such as online social services, community services, media-sharing applications, and, in general, new distributed applications in contexts such as the Internet of Things.

We make PP2db available under an Open Source license at [3].

References

1. PERIMETER - User-centric paradigm for seamless mobility in future internet. STREP, EU FP7 Grant No. 224024, http://www.ict-perimeter.eu/
2. Architecture & Transport Working Group. Tripleplay Services Quality of Experience (QoE) Requirements. TR-126, DSL Forum (December 2006)
3. PP2db: A Privacy-Protected, P2P-based Scalable Storage System for Mobile Networks, http://www.ing.unibs.it/ntw/tools/pp2db
4. Andersson, K.: Always Best Served and Managed. Technical report, Lulea Univ. of Technology (2007)
5. Sartiani, C., Manghi, P., Ghelli, G., Conforti, G.: XPeer: A Self-Organizing XML P2P Database System. In: Lindner, W., Fischer, F., Türker, C., Tzitzikas, Y., Vakali, A.I. (eds.) EDBT 2004. LNCS, vol. 3268, pp. 456–465. Springer, Heidelberg (2004)
6. XQuery 1.0: An XML Query Language, http://www.w3.org/TR/xquery
7. Chaum, D., van Heyst, E.: Group Signatures. In: Davies, D.W. (ed.) EUROCRYPT 1991. LNCS, vol. 547, pp. 257–265. Springer, Heidelberg (1991)
8. Ateniese, G., Camenisch, J., Joye, M., Tsudik, G.: A Practical and Provably Secure Coalition-Resistant Group Signature Scheme. In: Bellare, M. (ed.) CRYPTO 2000. LNCS, vol. 1880, pp. 255–270. Springer, Heidelberg (2000)
9. Camenisch, J., Groth, J.: Group Signatures: Better Efficiency and New Theoretical Aspects. In: Blundo, C., Cimato, S. (eds.) SCN 2004. LNCS, vol. 3352, pp. 120–133. Springer, Heidelberg (2005)
10. Sun Microsystem. Java Cryptography Architecture Reference Guide for JavaTM Platform Standard Edition 6
11. Tonesi, D., Salgarelli, L., Sun, Y., La Porta, T.F.: Evaluation of signaling loads in 3GPP networks. IEEE Wireless Communications 15(1), 92–100 (2008)
12. Ferri, D.: Secure P2P Storage Systems: Techniques and Architectures. Technical report, University of Brescia, M.Sc. Thesis (2010)