Optimized User Revocation for Group Signature Based Privacy-aware PKI

Wei Ren and Kui Ren
Dept. of ECE
Illinois Institute of Technology
3301 Dearborn St, Chicago, IL, USA 60616
{wren,kren}@ece.iit.edu

Wenjing Lou
Dept. of ECE
Worcester Polytechnic Institute
100 Institute Road, Worcester, MA, USA 01609
wjlou@ece.wpi.edu

Abstract

Privacy-aware Public Key Infrastructure (PKI) can maintain user access control and yet protect user privacy, which is important for many applications. The applicability of privacy-aware PKI highly relies on the performance of user revocation. The requirements of user revocation are various in general, such as subscription expiration, violation of access policy, group changing, and key exposure. To satisfy different requirements, multiple revocation approaches may interact each other. In this paper, we study how to achieve optimized user revocation cost with respect to various revocation approaches. We also propose a practical scheme Delta-RL that can fulfill an optimal overall performance on the base of extensive analysis.

1 Introduction

Conventional Public Key Infrastructure (PKI) is designed in an era when privacy is not a critical issue for business hence privacy protection is not taken into account. This situation has changed with the proliferation of the mobile devices and sensors, and the vast applications of wireless networks and ubiquitous computing. Privacy-aware PKI can protect both the privacy of users and the security of services. Some wireless networks such as vehicular networks can rely on privacy-aware PKI to provide access control and yet protect user privacy [12, 13, 14].

Recently G. Calandriello et al. [6] propose a privacy-enhancing authentication mechanism for vehicular ad-hoc networks by taking advantages of group signature and pseudonyms generated on-the-fly. K. Zeng [14] proposes a pseudonymous PKI for ubiquitous computing. X. Lin et al. [9] propose a secure and privacy-preserving protocol based on group signature and identity-based signature for vehicular communications. J. Guo et al. [8] propose a group signature based framework for vehicular communications. However, the revocation issue is not the focus of the paper and hence they all do not discuss the performance of user revocation extensively.

In this paper we focus on group signature based privacy-aware PKI as a case study. Group signature introduced by Chaum and Heyst [7], provides the authentication of the signer in certain group but protects the anonymity of the signer. Each member in the group can generate the valid signature using group secret key. Verifiers can verify the signature is from the given group with the group public key, but they do not know who signs the signature. For example, in vehicular networks privacy-aware PKI can protect the privacy of the user's location. When driving at different locations, the user sends messages signed by the group secret key to the others. The message is authenticated but others do not know who sends the message, which hence protects user's location privacy.

To apply group signature scheme in privacy-aware PKI, the efficiency of user revocation is important. In general, a user needs to be revoked due to various reasons: e.g., her subscription is expired; she violates the network access policy; she changes her group; and her group secret keys are exposed. Different schemes are selected for achieve the various user revocation requirements. It therefore raises a problem: how to design an efficient mechanism that both satisfies such diverse requirements and yet maintains optimized overall performance.

We address this problem in this paper. We first describe three schemes and analyze their performance. We then propose a new scheme - Delta-RL - with optimized performance. The Delta-RL scheme both adapts to diverse revocation requirements in practices and maintains optimized overall performance as well. To the best of our knowledge, this is the first paper addressing user revocation in group signature based privacy-aware PKI from the view point of performance evaluation and optimization.

Contributions: The contribution of this paper is as follows:

(1) We propose Delta-RL to satisfy the diverse revocation requirements by synthesizing and improving the three basic

1545-0678/08 $25.00 © 2008 IEEE
DOI 10.1109/ICDCS.Workshops.2008.95

schemes.

(2) We discover the performance optimization method between different schemes and suggest an optimized value in the Delta-RL scheme using queueing theory analysis.

Organization: The rest of the paper is organized as follows. The preliminary, assumptions and problem formulation are presented in Section II. Section III presents the proposed scheme Delta-RL. Section IV concludes the paper.

2 Preliminaries

2.1 Group Signature and Revocation List Based Scheme

The group signature scheme generally has four algorithms [2]: group key generation, group signing, group signature verification, and open algorithm (to determine the identity of signer). A group signature scheme ***GS = (GKg, GSig,GVf, Open)*** consists of four polynomial-time algorithms [2]:

(1) The randomized group key generation algorithm ***GKg*** takes input $1^k, 1^n$, where $k \in N$ is the security parameter and $n \in N$ is the group size (for example, the number of members of the group), and returns a tuple $(gpk, gmsk, gsk)$, where gpk is the group public key, $gmsk$ is the group manager's secret key, and gsk is an n-vector of keys with $gsk[i]$ being a secret signing key for player $i \in [n]$.

(2) The randomized group signing algorithm ***GSig*** takes as input a secret signing key $gsk[i]$ and a message m to return a signature of m under $gsk[i]$ ($i \in [n]$).

(3) The deterministic group signature verification algorithm ***GVf*** takes as input the group public key gpk, a message m, and a candidate signature σ for m to return either 1 or 0.

(4) The deterministic opening algorithm ***Open*** takes as input the group manager secret key $gmsk$, a message m, and a signature σ of m to return an identity i or the symbol $\perp$ to indicate failure.

We concentrate on Revocation List (*RL*) based group signature scheme proposed by D. Boneh et al. [4], since the revocation part is our interest in this paper. It comprises three algorithms, *KeyGen*, *Sign* and *Verify*.

KeyGen(n). It is a random algorithm that takes as input a parameter n, the number of members of the group. It outputs a group pubic key *gpk*, an n-element vector of user keys $gsk = (gsk[1], gsk[2], ..., gsk[n])$, and an n-element vector of user revocation tokens *grt*, similarly indexed.

Sign (gpk, gsk[i], M). This is a randomized algorithm that takes as input the group public key *gpk*, a private key $gsk[i]$, and a message $M \in \{0,1\}^*$, and returns a signature σ.

Verify (gpk, RL, σ, M). The verification algorithm takes as input the group public key *gpk*, a set of revocation tokens (*RL*, whole elements form a subset of the elements of *grt*), and a signature σ on a message M. It returns either valid or invalid. The latter response can mean either that σ is not a valid signature, or that the user who generated it has been revoked.

2.2 Network Assumptions

We discuss two major entities in the group signature based privacy-aware PKI: one is the issuer, the other is the group members (called users). The issuer distributes the keys such as group public key *gpk* and group private keys *gsks*, and revokes the users in the group. We assume the secret channel is maintained between the issuer and each group member if the gsks are distributed online.

Considering the applications in wireless networks, we focus on the cost optimization for the profit of users instead of the issuer because the users (not issuer) always have some resource constraints in terms of computation, communication and storage. Different applications may have different performance evaluation metrics. Without loss of generality, in this paper all the cost is measured by a virtual price. For example, the cost consumption for unit storage (e.g., 1 byte) is denoted by K_s. The signature verification cost has two parts: One is the signature checking cost; The other is the revocation checking cost. The revocation checking cost to verify signatures using unit length of revocation list is denoted by K_v (because the verification cost grows linearly to the length of the revocation list [4]). The cost for a user to receive unit length packet is denoted by K_c. The pricing is determined according to the performance tradeoff between the delay, power consumption, or storage, so that the issuer can set up a customized price using the same notation.

2.3 Problem Formulation

We observe that the reason for user revocation can be one of follows: the service subscription is expired; the user violates the network access policy; the user changes group intentionally (e.g. dynamic groups [3]); or the group secret key is compromised. To perform the revocation, a straightforward way is to redistribute the group secret keys and the group public key to all the users except for the revoked users, so that the revoked users can not generate valid signature afterward. In this way the secret channel is required for key distribution and communication overhead is induced by transmitting a large number of keys.

The user revocation methods can be classified into two categories in general. One is based on witness [11], and the other is based on Revocation List (*RL*) [4, 1, 10, 5]. In witness-based schemes, every group member proves in a zero knowledge way that she knows corresponding witness to a public value. A single short public broadcast message

needs to be sent to all signers and verifiers. Witness-based schemes have a drawback: previously signed signatures cannot pass verification function after the signer is revoked (due to the update of public value), so we do not discuss these schemes. In RL-based schemes, *RL* is the list of all revoked members. Issuer only sends *RL* to verifier. When a user verifies signatures, *RL* is imported into signature verification function. Signatures from the members in the *RL* will result in the verification failure. The communication cost decreases because the length of *RL* is shorter than a batch of secret keys, whereas the verification delay increases because the signature verification time grows linearly with the number of revoked users [4]. Intuitively, some tradeoff between the computation, communication and storage exists in the design, and an optimized selection can be achieved. Therefore, the challenging problem is how to design such an optimized scheme to achieve efficient user revocation.

2.4 Notation

Table 1 lists the notation used in the rest of the paper.

Table 1. Notation

gsk	group secret key
gpk	group public key
grt	group member revocation token
K_s	unit storage cost
K_c	unit communication cost
K_v	unit computation cost for revocation check
L_{gsk}	length of *gsk*
L_{gpk}	length of *gpk*
RL	Revocation List
L_{RL}	varying length of *RL*
λ_s	arrival rate of the signature packet
λ_{RL}	arrival rate of *RL* packet

3 The Proposed Schemes

3.1 Periodic Revocation (PR)

One reason of user revocation is the expiration of the user's service subscription, which occurs frequently and regularly. Once the subscription consumes away, the user's *gsk* should be invalidated. We design PR scheme for this purpose. The entire service providing time is divided into several time slots (e.g., three months as a slot). One *gpk* and a bunch of corresponding *gsks* (pool), are generated for each time slot. When a new user applies to join the group, the issuer distributes corresponding keys according to her service time slots. That is, selects the *gpk* and one *gsk* from the pool for each slot, and distributes all the keys for all the time slots that cover the subscription. As a consequence, different user may have different number of key pairs. For example, suppose each time slot has τ days. The subscriber u_i pays for n_i time slots of services (namely, $n_i * \tau$ days), and thus obtains n_i pairs of *gsk* and *gpk*. At the end of each service time slot, the user is automatically revoked due to the invalidation of the keys. In short, the scheme is described as follows:

$$u_i \longleftarrow \langle gsk[t_j, u_i], gpk[t_j] \rangle, t_j \in [1, n_i].$$

Where u_i is the user i; $gsk[t_j, u_i]$ and $gpk[t_j, u_i]$ are u_i's keys in the time slot t_j; n_i is the number of total service time slots.

If the keys are deployed off-line upon the user's subscription, the total cost for a user is as follows:

$$K_s * (L_{gsk} + L_{gpk}) * n_i \quad (1)$$

Where K_s is the unit storage cost; L_{gsk} is the length of *gsk*; L_{gpk} is the length of *gpk*; n_i is the number of the time slots covering the service time.

Discussions: The length of the time slot depends on the security policy and performance tradeoff. If the time slot is shorter, the exposed *gsk* can be used for a shorter time. If the time slot is longer, the required keys is less so that the storage cost is smaller. Also, to decrease the total number of required keys, the length of the time slot may be variant. Learning from the previous security statistics, we may heuristically differentiate some "safer" duration from others. We choose a longer time slot span in the "safer" duration to save the total number of required time slots, as well as the amount of required keys. In addition, the time slots can be inconsecutive in the service time, e.g. January, and from June to August each year.

3.2 Timely Revocation (TR)

In PR scheme the multiple keys are distributed off-line upon the user's subscription. It is efficient for the regular revocation due to the subscription expiration, but it cannot resolve the requirement that the *gsk* must be revoked timely, e.g., some users violate the access policy, group changing, or key exposure. To address these situations, the TR scheme is designed.

In TR scheme the users can be revoked at any time within one time slot by redistributing the keys. For example, to revoke the user u_i, the issuer broadcasts a new *gpk* and distributes new *gsks* to all the group members except for u_i. Similarly, multiple users can be revoked simultaneously. The new issuing *gsk* and *gpk* are valid till to the end of the time slot. In the next time slot the new *gsk* and *gpk* start to validate following the PR scheme, so the revocation persists

only within one time slot. In short, the scheme is described as follows:

$$u_j \longleftarrow \langle gsk[j], gpk \rangle, j \neq i, j \in [1, N].$$

Where u_j is the user j; i is the index of the revoked user; N is the number of the total users.

The cost for key re-distribution for a user in TR is:

$$K_c * (L_{gsk} + L_{gpk}) + K_s * (L_{gsk} + L_{gpk}) \quad (2)$$

Where K_c is the unit communication cost; L_{gsk} is the length of the *gsk*; L_{gpk} is the length of the *gpk*; K_s is the unit storage cost.

3.3 Revocation List Scheme (RLS)

In TR scheme the revocation persists only to the end of the time slot since the *gpk* is different in the new time slot. It may not be efficient for a group with many users (a large group) in the scenarios that the keys are re-distributed frequently, e.g, the scenarios that multiple users need to be revoked asynchronously in one time slot, or multiple users need to be revoked in different time slots, or the revocation needs to persist multiple time slots. Therefore, it may not be suitable for a large group in the situations such as frequent revocation, highly dynamic group, or long term revocation (key exposure).

To mitigate the communication overhead, in RLS scheme the issuer revokes users by broadcasting *RL*, not key re-distribution. In RLS scheme when revoking the user u_i, the issuer adds u_i's group revocation token grt_i into *RL* and broadcast *RL*. The users correctly maintain the *RL* locally. When a user verifies received signature, the *RL* is imported into the verification function. If the function returns invalid, the signature is either a invalid signature or the user who generates it has been revoked. Once the user receives a new *RL*, her old *RL* will be abandoned. In short, the *RL* scheme can be described as follows:

$$* \longleftarrow \langle RL \rangle, grt_i \in RL, i \in [1, N].$$

Where i is the index of the revoked user; * means all the users.

In RLS scheme the system performance is mainly determined by the length of *RL*. In particular, the computation overhead to verify the signature grows linearly to the length of the *RL* stored at a user. The total costs for a user in the duration t is:

$$K_s * L_{RL} + \lambda_{RL} * t * K_c * L_{RL} + \lambda_s * t * K_v * L_{RL} \quad (3)$$

Where K_s is the unit storage cost; K_c is the unit communication cost; K_v is the revocation checking cost for verifying the signature using *RL*; L_{RL} is the average length of *RL*; λ_s is the arrival rate of signature packets; λ_{RL} is the arrival rate of *RL* packets.

3.4 Delta-RL (DRL) Scheme

In RLS scheme while the number of invoked users grows larger, the length of *RL* increases. Once a user needs to be revoked, the entire *RL* has to be sent, which raises a large amount of communication overhead. Also, the time for signature verification grows longer. To mitigate the communication overhead, we suggest to broadcast the *RL* that includes only additional revoked users. To restrain the signature verification delay, we propose a optimized threshold time to shorten *RL*. Moreover, the PR, TR, or RLS scheme is only appropriate respectively for single revocation situation, but in real applications the comprehensive scheme is required, which needs to synthesize three basic schemes coordinately. We therefore propose a comprehensive scheme with *Delta-RL* by taking the advantages of the PR, TR and RLS, and in particular, by reinforcing an optimized design. The procedures of the *Delta-RL* scheme are described in detail as follows:

Key Generation and Pre-distribution:

(1) For each time slot, the issuer generates a set of keys including multiple *gsks* and one *gpk*. Each set of keys is only valid for one time slot.

(2) When a user applies for the service, the issuer distributes multiple key pairs covering her subscription. Each key pair consists of *gsk* and *gpk* that are only valid for one time slot. Different subscribers may have different number of key pairs, as the service spans may be different.

Revocation List Distribution:

(1) Each user maintains a *RL* locally, called user's *RL*, which is the list of revoked user's *grts*. The *grt* is the identity of the revoked user, but the genuine identity of the user is still unknown to others. The *grt* for each user varies in different time slot, so the published *grts* of the same revoked user is un-linkable. The users' *RL* will be abandoned at the end of each time slot, and obtain a new *RL* from the issuer at the beginning of the new time slot.

(2) The issuer also maintains a *RL* called issuer's *RL* for each time slot, because the revoked users may be different in various time slots. If a user is detected to be compromised, or severely violate access policy, she will be revoked for a long term sustaining several time slots. To do it, her current *grt* will be broadcasted in the revoking time slot, and her other *grts* for the future time slots will be recorded in the issuer's *RLs*. The issuer's *RL* for certain time slot is distributed either to all the group users at the beginning of each time slot, or to the new user upon her subscription (of course, together with the key distribution), which is always executed off-line.

Timely Revocation using Delta-RL:

(1) If any user needs to be revoked, the issuer broadcasts one *Delta-RL (DRL)* packet including her *grt*. *DRL* indicates the only *grts* that will be added into the user's local *RL*, instead of the entire *RL* in RLS scheme, to diminish the

communication cost.

(2) While verifying the signature, the user imports her *RL* into the verification function. The signature signed by the user whose revocation token is in the *RL* will result in failure.

(3) While the length of *RL* grows longer, it will cost longer time for the signature verification and more storage overhead. If the *RL* length reaches to a threshold value, the cost caused by *RL* is larger than the cost of keys redistribution. Therefore, the re-distribution of the *gsks* should be invoked, and the tokens in the *RL* will be cleared thereafter. The new set of keys will be re-distributed to all the users except for the revoked users. The DRL scheme is outlined in Fig. 1.

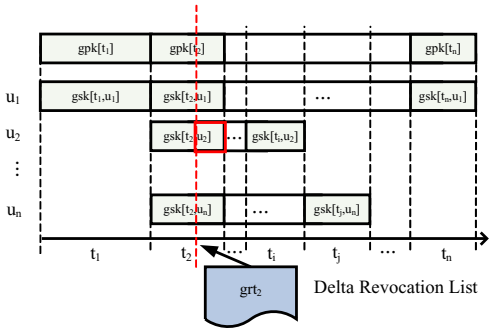

Figure 1. DRL scheme

3.5 Analysis

(1) The *DRL* scheme has more advantages than *RL* scheme in terms of communication overhead.

Justification: Suppose the number of total users is N. In the revocation packet RL_i, x_i $(1 \leq i \leq m)$ users are revoked, which compose a set $(Set(i) = U_{i1}, U_{i2}, ..., U_{ix_i})$. The number of total *RL* packets are m. The number of total revoked users is $X = \sum_{i=1}^{m} x_i \leq N$. Let $\Gamma(Set(i))$ be the members in $Set(i)$, so $\Gamma(Set(i)) = x_i$. In *RLS* scheme the communication overhead for RL_i and $RL_{i+1}(1 \leq i \leq m-1)$ is $C_1 = K_c * (x_i + x_{i+1})$ because it includes the total revoked users each time, but in *DRL* scheme the communication overhead for revoking same users is $C_2 = K_c * x_i + K_c * \Gamma(Set(i+1) - Set(i) \cap Set(i+1))$. $\Gamma(Set(i+1) - Set(i) \cap Set(i+1)) \leq x_j$, so $C_1 \leq C_2$. For all i, we have same results.

(2) There is a threshold value that the revocation list based method has less advantages than re-distribution method.

Justification: The length of *RL* grows with the time elapsing. The verification cost grows linearly with the length of the *RL*. Suppose the length of original *RL* is L, the arrival rate of the signature packet is λ_s, and the arrival rate of the *DRL* packet is λ_{DRLa}, thus the verification cost caused by *RL* in time span t is $C1 = K_v * (\lambda_s * t * (\lambda_{DRLa} * t + L)) = \lambda_s \lambda_{DRLa} K_v t^2 + \lambda_s L K_v t$. The cost of the key redistribution is Eq. 2, $C2 = K_c * (L_{gsk} + L_{gpk}) + K_s * (L_{gsk} + L_{gpk})$. Therefore, $\exists t_{th}$, s.t. $t \geq t_{th} \Rightarrow C1 \geq C2$ and $t \leq t_{th} \Rightarrow C1 \leq C2$. t_{th} is the threshold value.

The simulation result is depicted in Fig.2, which justifies our analysis.

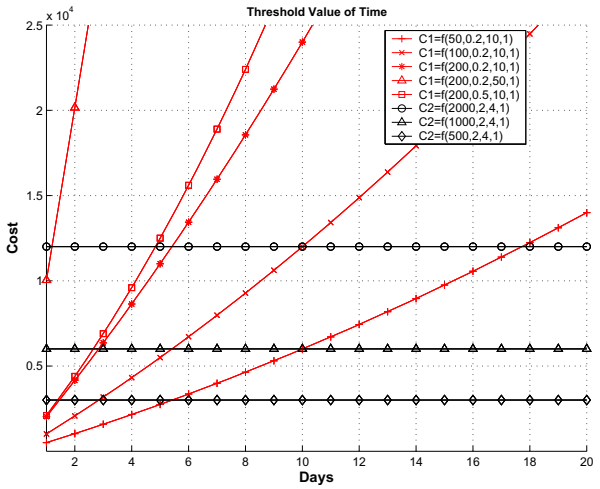

Figure 2. Threshold value of the Time t_{th}. $C1 = \lambda_s \lambda_{DRLa} t^2 + \lambda_s L t$. $C2 = K_c * (L_{gsk} + L_{gpk}) + K_s * (L_{gsk} + L_{gpk})$. $C1 = f(\lambda_s, \lambda_{DRLa}, L, K_v)$. $C2 = f(K_c, L_{gsk}, L_{gpk}, K_s)$. If $t \leq t_{th}$, then $C1 \leq C2$. If $t \geq t_{th}$, then $C1 \geq C2$.

4 Conclusion

We proposed a scheme Delta-RL that can satisfy various revocation requirements. For different revocation requirements, Delta-RL provides periodic revocation, timely revocation, or revocation list approach accordingly. Delta-RL can also achieve optimized overall performance. The communication overhead is lower in DRL scheme than in basic revocation list scheme. We proofed the threshold time exists in revocation list based scheme, and employed such value for performance optimization.

References

[1] G. Ateniese, D. Song, and G. Tsudik. Quasi-efficient revocation in group signatures. In *Proc. of Financial Cryptography (FC'02), LNCS 2357*, pages 183–197, 2002.

[2] M. Bellare, D. Micciancio, and B. Warinschi. Foundations of group signatures: formal definitions, simplified requirements, and a construction based on general assumptions. In *Proc. of Eurocrypt'03*, pages 614–629, 2003.

[3] M. Bellare, H. Shi, and C. Zhang. Foundations of group signatures: The case of dynamic groups. In *Proc. of CT-RSA 2005*, 2005.

[4] D. Boneh and H. Shacham. Group signatures with verifier-local revocation. In *Proc. of ACM CCS'04*, pages 168–177, 2004.

[5] E. Bresson and J. Stern. Efficient revocation in group signatures. In *Proc. of PKC'01, LNCS 1992*, pages 190–206, 2001.

[6] G. Calandriello, P. Papadimitratos, J.-P. Hubaux, and A. Lioy. Efficient and robust pseudonymous authentication in vanet. In *VANET '07: Proceedings of the fourth ACM international workshop on Vehicular ad hoc networks*, pages 19–28, New York, NY, USA, 2007.

[7] D. Chaum and E. van Heyst. Group signature. *Proc. of Eurocrypt'91*, 547:257–265, 1991.

[8] J. Guo, J. Baugh, and S. Wang. A group signature based secure and privacy-preserving vehicular communication framework. In *Proc. of MOVE workshop in IEEE INFOCOM'07*, pages 103–108, May 2007.

[9] X. Lin, X. Sun, P.-H. Ho, and X. Shen. Gsis: A secure and privacy-preserving protocol for vehicular communications. *Vehicular Technology, IEEE Transactions on*, 56(6):3442–3456, Nov. 2007.

[10] T. Nakanishi and N. Funabiki. Verifier-local revocation group signature schemes with backward unlinkability from bilinear maps. In *Proc. of ASIACRYPT'05, LNCS3788*, pages 533–548, 2005.

[11] L. Nguyen. Accumulators from bilinear pairings and applications. In *Proc. of CT-RSA'05, LNCS 3376*, pages 275–292, 2005.

[12] P. Persiano and I. Visconti. An anonymous credential system and a privacy-aware PKI. In *Proc. of Australasian Conference on Information Security and Privacy (ACISP'03)*, 2003.

[13] M. Raya and J.-P. Hubaux. The security of vehicular ad hoc networks. In *Proc. of ACM Workshop on Security of Ad Hoc and Sensor Networks (SASN'05)*, 2005.

[14] K. Zeng. Pseudonymous pki for ubiquitous computing. In *Proc. of EUROPKI 2006*, 2006.