# Efficient User Revocation for Privacy-aware PKI

Wei Ren and Kui Ren
Dept. of ECE
Illinois Institute of Technology
{wren,kren}@ece.iit.edu

Wenjing Lou
Dept. of ECE
Worcester Polytechnic Institute
wjlou@ece.wpi.edu

Yanchao Zhang
Dept. of ECE
New Jersey Institute of Technology
yczhang@njit.edu

## ABSTRACT

Privacy-aware Public Key Infrastructure (PKI) can maintain user access control and yet protect user privacy, which is envisioned as a promising technique in many emerging applications. To justify the applicability of privacy-aware PKI and optimize the performance, it is highly important to ensure the efficiency of handling user revocations. In practice, user revocation can be due to various predictable and unpredictable reasons, e.g., subscription expiration, network access policy violation, group changing, secret key exposure, etc. Both predictable and unpredictable reasons can happen concurrently, which makes the design of efficient user revocation mechanism challenging. In this paper, we study how to achieve optimized user revocation cost with respect to various revocation approaches. We also propose an advanced scheme Delta-RL that ensures an optimized overall performance in terms of communication, computation and storage, as justified by the extensive analysis.

## Categories and Subject Descriptors

E.3 [**Data**]: Data Encryption-Public Key Cryptosystems

## General Terms

Design, Security, Performance

## Keywords

PKI, User Revocation, Privacy

## 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 [15, 16, 18].

Recently G. Calandriello et al. [5] 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 [18] proposes a pseudonymous PKI for ubiquitous computing. X. Lin et al. [10] propose a secure and privacy-preserving protocol based on group signature and identity-based signature for vehicular communications. J. Guo et al. [9] propose a group signature based framework for vehicular communications. However, the revocation issue is not the focus of the papers 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 the privacy of users' location.

To apply group signature scheme in privacy-aware PKI, the efficiency of user revocation is important. In practice, various predictable and unpredictable situations lead to user revocation, e.g., subscription expiration, access policy violation, group changing, keys exposure, etc. Such situations can occur concurrently so that the design of efficient user revocation mechanism is challenging. In this paper, we try to qualitatively characterize the different performance aspects of user revocation in group signature based privacy-aware PKI. We first describe and analyze three basic schemes. Atop of our qualitative analysis, we then propose a new hybrid scheme - Delta-RL. The proposed Delta-RL scheme achieves optimized performance when encountering both predictable and unpredictable user revocation situations. To our best knowledge, this is the first paper addressing user revocation in group signature based privacy-aware PKI from the view point of performance evaluation and op-

timization.

**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 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, network assumptions and problem formulation are presented in Section II. Section III analyzes the performance of the possible schemes, and presents the proposed scheme Delta-RL. In Section IV, we derive the optimization method for the proposed scheme. Finally, Section V concludes the paper.

# 2. PRELIMINARIES

## 2.1 Revocation List Based GS Scheme

We concentrate on Revocation List ($RL$) based Group Signature (GS) scheme proposed by D. Boneh et al. [3], because the revocation is our interest in this paper. The scheme 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 $\sigma$.

***Verify (gpk, RL, $\sigma$, 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 $\sigma$ on a message $M$. It returns either valid or invalid. The latter response can mean either that $\sigma$ is not a valid signature, or that the user who generated it has been revoked.

## 2.2 Privacy-aware PKI Model

We discuss two major entities in the group signature based privacy-aware PKI: one is the Trust Third Party (TTP), the other is the group members (called users). The TTP distributes the keys such as group public key *gpk* and group private keys *gsks*, and revokes the users in the group. We assume the basic key management scheme is established between the TTP and each user to secretly distribute the gsks. We concentrate on the performance optimization for the users since the users always have some resource constraints in terms of communication, computation and storage. We assume the relevant communication pattern in this paper is end to end with one hop. Besides, since different applications may have different performance metrics, we leave the flexibility to the operator to evaluate the concrete performance cost. That is, the cost can be unified (e.g., energy consumption) and denoted by the same universal notation. 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 [3]). The cost for a user to receive unit length packet is denoted by $K_c$. The operator therefore has the flexility to determine the customized price based on the performance tradeoff (such as energy consumption or delay) without changing the notation in our analysis.

## 2.3 Problem Formulation

We observe that the reason for user revocation may be one of the following: the service subscription is expired; the user violates the network access policy; the user changes group intentionally (e.g., dynamic groups [2]); 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 existing user revocation methods can be classified into two categories in general: One is based on witness [6, 14]; The other is based on Revocation List ($RL$) [3, 1, 12, 4]. 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. TTP only sends $RL$ to verifiers. 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 group secret keys, whereas the verification delay increases because the signature verification time grows linearly with the number of revoked users. 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. The related work on the certificate revocation [13, 8] cannot be applied because its inherent mechanism is different from the user revocation.

## 2.4 Notation

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

# 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

**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 $gpk$ |
| $L_{grt}$ | length of $grt$ |
| $RL$ | Revocation List |
| $URL$ | user's Revocation List |
| $TRL$ | TTP's Revocation List |
| $DRL$ | Delta Revocation List |
| $L$ | original length of $RL$ |
| $L_{RL}$ | varying length of $RL$ |
| $L_{DRL}$ | length of $DRL$ |
| $\lambda_s$ | arrival rate of the signature packet |
| $\lambda_{RL}$ | arrival rate of $RL$ packet |
| $\lambda_{DRL}$ | arrival rate of $DRL$ packet |
| $\lambda_{DRLa}$ | arrival rate of $DRL$ packet for user revocation |
| $\lambda_{DRLd}$ | arrival rate of $DRL$ packet for user revival |

(e.g., three months as a slot). One $gpk$ and a bunch of corresponding $gsks$ ($gsk$ pool), are generated for each time slot. When a new user applies to join the group, the TTP 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 the keys for all time slots that cover the subscription. As a consequence, different user may have various number of key pairs. For example, suppose each time slot has $\tau$ 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, we have:

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

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:

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

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 are 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 diverse. 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 for certain users may be inconsecutive due to the subscription, 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 users can be revoked at any time within one time slot by redistributing the keys. For example, to revoke the user $u_i$, the TTP 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 issued $gsk$ and $gpk$ are valid till to the end of the time slot. In the next time slot the new $gsk$ and $gpk$ becomes valid following the PR scheme, so the revocation persists only within one time slot. In short, to revoke user $u_i$ we have:

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

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:

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

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 is inefficient for a group with many users when the keys are re-distributed frequently, e.g., 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 for multiple time slots. Therefore, it may not be suitable for a large group with frequent revocation, highly dynamic group, or long term revocation (key exposure).

To mitigate the communication overhead, in RLS scheme the TTP revokes users by broadcasting $RL$, instead of key re-distribution. In RLS scheme when revoking the user $u_i$, the TTP 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, to revoke user $u_i$ we have:

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

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 the $RL$. In particular, the computation overhead to verify the signature grows linearly to the

length of $RL$ 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 signatures using $RL$; $L_{RL}$ is the average length of $RL$; $\lambda_s$ is the arrival rate of signature packets; $\lambda_{RL}$ 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, in particular, 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 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 details as follows:

***Key Generation and Pre-distribution***:

(1) For each time slot, the TTP 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 TTP 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 $RL$ (URL), 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 slots, so the published *grts* of the same revoked user is un-linkable. The URL will be abandoned at the end of each time slot, and obtain a new $RL$ from the TTP at the beginning of the new time slot.

(2) The TTP also maintains a $RL$ called TTP's $RL$ (TRL) 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 violates 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 TRL. The TRL 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 (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 TTP broadcasts one *Delta-RL (DRL)* packet. $DRL$ indicates the *grts* that will be added into the user's local $RL$, instead of the entire $RL$ in RLS scheme, to diminish the communication cost. This design is justified in the following analysis (1).

(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 re-distribution. Therefore, the re-distribution of the *gsks* should be invoked, and the tokens in the $RL$ will be cleared afterward. The new set of keys will be re-distributed to all the group members except for the revoked members. The observation of the existence of threshold value is justified in the following analysis (2).

*Analysis:*

(1) The $DRL$ scheme has more advantage 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. However, 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 $\lambda_s$, and the arrival rate of the $DRL$ packet is $\lambda_{DRLa}$. The signature verification cost caused by $RL$ in time span $t$ is therefore $C1 = K_v * (\lambda_s * t * (\lambda_{DRLa} * t + L)) = \lambda_s \lambda_{DRLa} K_v t^2 + \lambda_s L K_v t$. The cost of the key re-distribution 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 analysis results is depicts in Fig.1. From the graph we can find the threshold value exists.

# 4. SCHEME OPTIMIZATION

## 4.1 Revocation List Reduction

It is possible that the revoked user may apply to rejoin the group, or the user is revoked temporarily. The TTP nor-

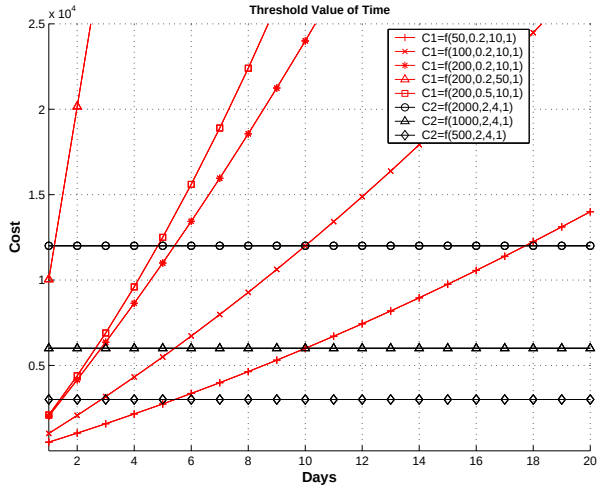


**Figure 1: Threshold value of the Time $t_{th}$. $C1 = \lambda_s\lambda_{DRLa}t^2 + \lambda_s Lt$. $C2 = K_c * (L_{gsk} + L_{gpk}) + K_s * (L_{gsk} + L_{gpk})$. $C1 = f(\lambda_s, \lambda_{DRLa}, L, K_v)$. $C2 = f(K_c, L_{gsk}, L_{gpk}, K_s)$. If $t \leq t_{th}$, then $C1 \leq C2$. If $t \geq t_{th}$, then $C1 \geq C2$.**

mally lets her rejoin the group in the next time slot by editing corresponding TTP's *RL* for that time slot. If she wants to reinstate immediately, the TTP has to distribute a new *gsk* to her. In this way such user's *grt* is still in *RL* and thus still affects the signature verification performance. Based on this observation, we suggest the TTP can remove tokens from the *RL* by sending *DRL* packet with *RL reduction*. The TTP thus can flexibly revoke and un-revoke users timely. No additional communication overhead is induced because the un-revoking information may be piggyback in user revocation packet. In particular, the user's privacy is still maintained since the token and the user's identity are un-linkable. Fig.2 depicts the outline of *Delta-RL* scheme.

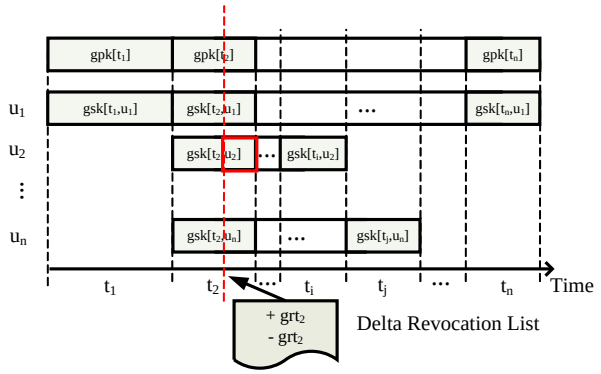


**Figure 2: *Delta-RL* scheme. Only updating information (addition or deletion) are enclosed in *DRL* packet, which has much less communication overhead than *RL* scheme. It also employ the threshold value to achieve the optimized overall performance.**

## 4.2 Modeling *RL* Length

(1) *Parameters*:

Because the overall performance is mainly determined by the length of the *RL*, we focus on the modeling of the *RL* length using queueing theory, which is the additional length in one time slot. Queueing models may be different corresponding to the different revocation policies. The revocation policies determine two major factors that affect the modeling choices. One is *the probability distribution of DRL packet's arrival*. Basically, the *DRL* is sent once a user needs to be revoked, so we model the arrival of the *DRL* packets as Poisson arrival [17]. The expectation of the arrivals of *DRL* in $t$ is $\lambda_{DRL} * t$. Alternatively, if the *DRL* is sent periodically, the arrival will be determinate distribution. In this case the arrival number of *DRLs* in $t$ is a constant value. The other factor is *the possible number of grts in the DRL packet*. In *DRL* packet, the number of *grts* for user revocation is $N_a$; the number of *grts* for user revival is $N_d$. Assume the total number of the members in the group is $N$ and revoked members is $m$, so $N_a < N - m$ and $N_d < m$. The value of $N_a$ is also further determined by the revocation policy.

To estimate the cost of signature verification, we need to model the arrival of signature packets. We assume signature packet is also Poisson arrival at any user in the group with the rate $\lambda_s$.

(2) *Policy I - Instant Revocation and Revival*:

If the TTP broadcasts one *DRL* packet when one user needs to be revoked or revived, the $N_a$ and $N_d$ are both equal to one. The length of the *RL* is modeled as the number of the customers in queue. If arriving *DRL* includes one item to be added into the *RL*, it is looked as one customer's arrival. If arriving *DRL* includes one item to be deleted from the *RL*, it is looked as one customer's departure. The maximal number of the queue is $N$. Therefore, the length of *RL* can be modeled as $M/M/1/N$ queueing system. Recall that the $a/b/c/d$ notation in queueing theory is: the custom arrival is Poisson distribution (denoted by $M$); the service time is exponential distribution (denoted by $M$); there is only one server (denoted by 1) with limited buffer size (denoted by $N$). According to the queueing theory, the stationary length of *RL* for a long term is:

$$L_{RL} = \frac{\lambda_{DRLa}[1 + N(\frac{\lambda_{DRLa}}{\lambda_{DRLd}})^{N+1} - (N+1)(\frac{\lambda_{DRLa}}{\lambda_{DRLd}})^{N}]}{(\lambda_{DRLd} - \lambda_{DRLa})(1 - (\frac{\lambda_{DRLa}}{\lambda_{DRLd}})^{N+1})}, \quad (4)$$

where $L_{RL}$ is the stationary length of *RL*; $\lambda_{DRLa}$ is the arrival rate of the *DRL* packet with user revocation; $\lambda_{DRLd}$ is the arrival rate of the *DRL* packet with user revival.

(3) *Policy II - Instant Revocation and Delayed Revival*:

Sometimes the user revival is not as urgent as the user revocation, so one possible policy is the TTP may send one *DRL* only when one user needs to be revoked or multiple users need to be revived. This revocation policy can decrease the communication overhead of *DRL* due to the reduction of *DRL* packets. For example, only when accumulative $q$ users need to be removed from *RL*, one *DRL* packet inclosing such users' *grts* will be broadcasted. For this case we model the length of *RL* by $M/M/1/N$ queueing system with *bulk services* of number $q$. We define the state as the length of the *RL*. The number of tokens can be deleted in a batch is

$q$. Fig.3 depicts the state transition diagram. The balance

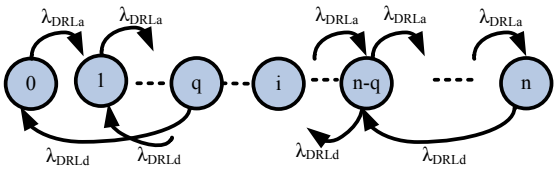


**Figure 3: *DRL* with bulk revival. *RL* is sent when one member needs to be revoked or $q$ members need to be revived.**

equation for transition diagram is:

$$\lambda_{DRLa} * P_0 = \lambda_{DRLd} * P_q \quad (5)$$

$$\lambda_{DRLa} * P_i = \lambda_{DRLd} * P_{q+i} + \lambda_{DRLa} * P_{i-1} \quad (6)$$
$$(1 \leq i \leq q-1)$$

$$(\lambda_{DRLa} + \lambda_{DRLd}) * P_i = \lambda_{DRLd} * P_{q+i} + \lambda_{DRLa} * P_{i-1} \quad (7)$$
$$(q \leq i \leq n).$$

Suppose it has the solution of the form:

$$P_n = \alpha^n P_0. \quad (8)$$

After simple substitution of the equations, the average length of the $RL$ thus is:

$$L_{RL} = \frac{\alpha}{(1-\alpha^{N+1})(1-\alpha)}, \quad (9)$$

where $N$ is the number of total users.

(4) *Other Policies and Models*

Sometimes the *DRL* packet is sent once the member needs to be revoked, and the revival *DRL* is sent by certain deterministic distribution. The length of $RL$ can be modeled as the $M/G/1$ queueing system. In particular, if the *DRL* is sent periodically, we use $M/D/1$ to model the length of $RL$. The stationary length of $RL$ is:

$$L_{RL} = \frac{(\lambda_{DRLd}^2 \sigma^2 + 1)(\lambda_{DRLa}/\lambda_{DRLd})^2}{2(1-\lambda_{DRLa}/\lambda_{DRLd})}, \quad (10)$$

where $\sigma$ is the standard deviation of arrival time of *DRL* packet for revival.

The most general case is that the arrivals of *DRL* packets, for both revocation and revival, have arbitrary distribution. In this situation the $RL$ length can be modeled as the $G/G/1$ queueing system. For the sake of space, the equations for $G/G/1$ are not given here. In particular, if *DRL* is sent by beacons - the periodic broadcasting packets, the length of the $RL$ stored at a user can be modeled as $D/D/1$ queueing system. The arrivals of the *DRL* packets for the addition or deletion of $RL$ both have the deterministic distribution. Since the queue length dislikes the variance [17], the $D/D/1$ queueing system has the shortest average $RL$ length.

## 4.3 Optimal Time for Key Re-distribution

When the length of $RL$ grows to a threshold size, the overall cost will be larger than the overall that of the key re-distribution. The TTP then re-distributes the *gsks*, and thus the *grts* for current time slot in the $RL$ are cleared, which largely diminishes the signature verification cost. After the clearance if new users need to be revoked, the TTP sends new *DRLs*. According to Poisson arrival of the signature packets, in time span $T_{th}$ the revocation checking cost is:

$$\lambda_s * T_{th} * K_v * (L_{RL} + L), \quad (11)$$

where the $\lambda_s$ is the arrival rate of signature packets; $T_{th}$ is the threshold time value; $K_v$ is the unit revocation checking cost; $L_{RL}$ is the average additional length of $RL$ in one time slot; $L$ is the original length of $RL$ at the beginning of time slots.

The storage cost for $RL$ is:

$$K_s * (L_{RL} + L), \quad (12)$$

where $K_s$ is the unit storage cost; $L_{RL}$ is the average additional length of $RL$ in current time slot; $L$ is the original length of $RL$ at the beginning of time slots.

The broadcasting times of the *DRL* in time span $T_{th}$ is $\lambda_{DRL} * T_{th}$, so the communication cost of broadcasting *DRL* is:

$$\lambda_{DRL} * T_{th} * K_c * L_{DRL}, \quad (13)$$

where $\lambda_{DRL}$ is the arrival rate of *DRL* packets; $T_{th}$ is the threshold time value; $K_c$ is the unit communication cost; $L_{DRL}$ is the length of *DRL* packets.

Therefore, the total cost due to $RL$ is the summation of the Eq. 11, 12, and 13. That is:

$$\lambda_s T_{th} K_v (L_{RL}+L) + K_s (L_{RL}+L) + \lambda_{DRL} T_{th} K_c L_{DRL}. \quad (14)$$

If Eq. 14 $\leq$ Eq. 2, we have:

$$T_{th} \leq \frac{K_c(L_{gsk} + L_{gpk}) + K_s(L_{gsk} + L_{gpk} - L_{RL} - L)}{\lambda_s K_v (L_{RL} + L) + \lambda_{DRL} K_c L_{DRL}}, \quad (15)$$

where $L_{RL}$ is given by the Eq. 4, 9, or 10 according to the different revocation policies. Therefore, $T_{th}$ is the upper bound time of the distribution of *DRL*. After that, key re-distribution should be triggered and $RL$ will be cleared.

If the policy I is used in specific applications, the length of *DRL* almost equals $L_{grt}$ (token length). According to the aggregative property of Poisson arrival [17], we have $\lambda_{DRL} = \lambda_{DRLa} + \lambda_{DRLd}$. Also, $L_{RL} = m * L_{grt}$. More specifically, if the D. Boneh et al. scheme [3] is used, we have $L_{grt} = 0.5 * L_{gsk}$ and $L_{gpk} = 1.5 * L_{gsk}$. We also assume $L << L_{RL}$. Substitute Eq. 15, we have:

$$T_{th} \leq \frac{K_c L_{gsk} + 1.5 L_{gsk} + K_s(2.5 L_{gsk} - 0.5 m L_{gsk})}{0.5 \lambda_s K_v m L_{gsk} + 0.5(\lambda_{DRLa} + \lambda_{DRLd}) K_c L_{gsk}} \quad (16)$$

After simple transformation and assuming $\lambda_{DRLa} + \lambda_{DRLd} \ll \lambda_s$, we have:

$$T_{th} \leq \frac{K_c + 1.5 + 2.5 K_s - 0.5 K_s m}{0.5 \lambda_s K_v m} \quad (17)$$

or

$$T_{th} \leq \frac{\frac{\alpha + 5\beta}{m} - \beta}{\lambda_s}, \quad (18)$$

where $\alpha = (2K_c + 3)/K_v$ and $\beta = K_s/K_v$; $\lambda_s$ is the arrival rate of signature packets; $m$ is the number of revoked users.

Note that, the Eq. 18 shows $T_{th}$ is inversely proportional to the number of revoked users and arrival rate of the signature packets. It also provides a method to estimate the threshold value. Fig. 4 depicts the expected time of key re-distribution corresponding to the number of revoked users for given arrival rate of signature packets. We assume the communication cost is about three orders of magnitude more than the computation cost [11]. We find that the key re-distribution time is short (only one or two hours) if the signature verification traffic rate is about 4 packets per minute and even if the number of revoked users is small ($\leq 11$). In particular, the key re-distribution time will be shorter while the signature verification occurs more frequently or revoked users become more.

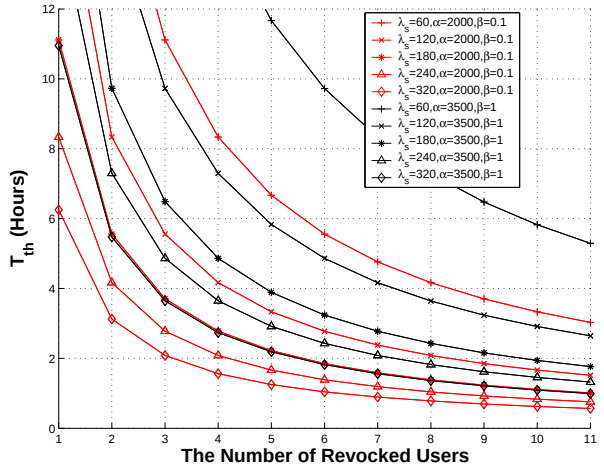


**Figure 4: The optimal time for key re-distribution**

## 5. 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 based revocation accordingly. Delta-RL can also achieve optimized overall performance. The communication cost in Delta-RL is lower than in basic revocation list scheme. For performance optimization, we proofed a threshold value for performance optimization exists between the revocation list scheme and timely revocation scheme. Based on this observation, we derived an upper bound time for revocation list distribution in Delta-RL, which is inversely proportional to the number of revoked users and arrival rate of signature packets.

## 6. ACKNOWLEDGMENTS

This research is supported in part by ERIF, IIT and National Science Foundation Grants CNS-0626601, CNS-0716306, and CNS-0716302.

## 7. 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, H. Shi, and C. Zhang. Foundations of group signatures: The case of dynamic groups. In *Proc. of CT-RSA 2005*, 2005.

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

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

[5] 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.

[6] J. Camenisch and A. Lysyanskaya. Dynamic accumulators and application to efficient revocation of anonymous credentials. In *Proc. of Crpto'02*, pages 61–76, Auguest 2002.

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

[8] D. Cooper. A more efficient use of delta-crls. In *Proc. of 2000 IEEE Symposium on Security and Privacy (S&P'00)*, pages 190–202, May 2000.

[9] 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.

[10] 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.

[11] G. Mathur, P. Desnoyers, D. Ganesan, and P. Shenoy. Ultra-low power data storage for sensor networks. In *IPSN '06: Proceedings of the fifth international conference on Information processing in sensor networks*, pages 374–381, New York, NY, USA, 2006. ACM.

[12] 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.

[13] M. Naor and K. Nissim. Certificate revocation and certificate update. *IEEE Journal on Selected Areas in Communications*, 18(4):561–570, April 2000.

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

[15] 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.

[16] 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.

[17] S. M. Ross. *Introduction to Probability Models*. Academic Press, ninth edition, 2006.

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