Journal of the Operations Research
Society of Japan
????, Vol. ??, No. ?, ????

# PERFORMANCE EVALUATION OF SECURE GROUP COMMUNICATION

Hiroshi Toyoizumi Matsuyoshi Takaya
*University of Aizu*

(Received November 26,2002; Revised July 22,2003)

*Abstract* There are demands for secure group communication on the internet, such as pay-per-view type broadcasting, business confidential information sharing and teleconference. Secure communication inside a groups on an open network is critical to enhance the internet capability. The public key system is not sufficient to support the group security, since it is not scalable for large groups. Some researchers propose a scalable group security model, managing several common keys for encryption and decryption sharing inside the community. In this paper, we will evaluate the performance of these group security model. Using $M/G/\infty$ queueing models and the basic queueing theory, we show how to find the optimal condition of the allocation of the common keys if the joins to group is a Poisson process. In addition, we show our optimal condition may work for more general arrival processes by using the cross covariance formula (a variant of Papangelou's formula) for the stochastic intensity of departure process.

**Keywords**: Applied probability, information technologies, Markov process, queue, telecommunication

## 1. Introduction

The basic model of secure communication is one-to-one and sharing the information between two person. One-to-one secure communication has been widely used on the internet such as, SSL (Secure Sockets Layer) [12]. We will illustrate a common procedure of these one-to-one secure communication. In the public key environment [2], two person, say Alice and Bob, have their own private and public key. When Alice wants to start a one-to-one secure communication with Bob, she creates a symmetric key, and send the key to Bob by using Bob's public key. Bob will decrypt the symmetric key with his own private key. The shared symmetric key among Alice and Bob will be used to encrypt and decrypt their communication. In general, since encryption and description using the public key system requires longer time for encryption than conventional symmetric encryption system like DES. It is appropriate to use public key system only once at the start to send the symmetric key, and then use the symmetric key to encryption and description of the information.

As the internet grows all over the world, we get the freedom to communicate with anyone, anytime, anywhere. On the internet, we can easily make a community which shares common interest. Inside the community, sometimes we need a secure communication to protect their own interest. For example, we need a secure group communication for pay TV on the internet, or sharing the business confidential information on the internet.

These secure group communication might be solved by one-to-one secure communication by specifying one sender and one receiver. However, if we use one-to-one model in group, the sender has to encrypt the information using different symmetric keys to each receivers.

When the group size is large and we need real-time encryptions, the one-to-one model will not be scalable. For example, consider an internet broadcasting company which has 10,000 subscribers. The server has to encrypt the data 10,000 times with different keys. Thus, it is impossible for streaming type real-time applications like Pay TV or teleconference.

One of the solutions to this problem is to share a common symmetric group key among the group, and use it when sending information [6] [5]. Using the group key, the sender can reduce the number of encryptions to one per data in the group. The group key might be sent to the participants of the group by using the one-to-one secure communication in advance. This group key model also has a problem to be solved. Since groups might be instable, some of the participants will leave and join the group in the future. If one participants left the group, we cannot use the same group key to keep the security of the communication. For example, a subscriber of Pay TV on the internet will quit watching the program when he/she feels it is not worth watching. If we keep using the same group key after the leave, some malicious subscriber will quit watching but keep watching the program using the group key in his/her hand. So, when the participants change in the group, we need to renew the old group key. Now, every time if one participant leaves a group of 10,000 participants, we have to encrypt the new group key 10,000 times to send them to each participants. Clearly, this model is not scalable.

In Wong [15] and RFC2627 [13], the authors introduce a concept of subgroup in the secure group communication to solve the above problem. They showed that using additional subgroup keys, they can decrease the number of encryptions of the group key, dramatically. The subgroup keys are exclusively shared in its subgroup, and used to encrypt a new group key.

In this paper, we use basic queueing theory to evaluate the number of encryptions in the subgroup model and show the optimal number of subgroups for Poisson arrivals. In addition, we show our optimal condition may work for more general arrival processes by using covariance formula for the stochastic intensity of departure process.

## 2. Secure Group Communication

Here we briefly summarize the idea of secure group communication. In the following, we write $(A)_b$, when data $A$ is encrypted by a key $b$. For simplicity, suppose we have a group of 15 subscribers, $U_1, ..., U_{15}$, and we have a key server which manages to issue group and subgroup keys.

A participant $U_i$ has its own public key $O_i$ and secret key $S_i$ (or symmetric key if both sides has already negotiated). The key server initially generate a group key $G(0)$. The group key $G(0)$ is encrypted by $O_i$, and $(G(0))_{O_i}$ is sent to $U_i$ on an open network. Each user uses his own secret (or symmetric) key to decrypt $G(0)$. Thus, participants can send the information encrypted by $G(0)$ to share them inside the group.

Now let us consider a scenario;

1. First, $U_{15}$ leaves the group.
2. Then, a new participant $U_{16}$ joins the group.

As pointed out in the previous section, when the participants changed, we need to renew the group key and send them to each participants with encryption. We will estimate the number of encryptions of new group keys. First, we will consider the case when there is no subgroup, and then we will evaluate the case with subgroups.

### 2.1. Without subgroups

Assume $U_{15}$ leaves the group. Since $U_{15}$ knows $G(0)$, the key server has to generate new group key $G(1)$ to keep the security inside the rest of the group $\{U_1, ..., U_{14}\}$. The new group key $G(1)$ is encrypted by each participant's open key to send it to them. Thus, we need the following encryptions;

$$\{(G(1))_{O_i}\}_{i=1,...,14}. \tag{2.1}$$

Let $A_{15}$ be the number of encryptions required for the leave of $U_{15}$, then we have

$$A_{15} = 14. \tag{2.2}$$

Now, assume a new participants $U_{16}$ subscribes to join the group. To protect the information shared by the group before $U_{16}$ joins, the key server generates another new group key $G(2)$ instead of $G(1)$. Since $U_{16}$ does not know $G(1)$, we can use $G(1)$ to encrypt $G(2)$. Together with encryption to $U_{16}$, we need the encryptions as

$$\left\{(G(2))_{G(1)}, (G(2))_{O_{16}}\right\}.$$

Thus, letting $B_{16}$ be the number of encryptions at the join of $U_{16}$, we have

$$B_{16} = 2. \tag{2.3}$$

### 2.2. With subgroups

Here, we divide the group into 3 subgroups: $SG_1 = \{U_1, ..., U_5\}$, $SG_2 = \{U_6, ..., U_{10}\}$, $SG_3 = \{U_{11}, ..., U_{15}\}$. Initially, the key server generates and sends subgroup keys $(SG_j(0))_{j=1,2,3}$ to members of each subgroups. For example, since $U_6$ belongs to $SG_2$, $U_6$ has the keys $(G(0), SG_2(0), S_6)$, but does not have the keys of other subgroups like $SG_1(0)$ and $SG_3(0)$.

Now suppose $U_{15}$ leaves the group. To keep the security, not only renew the group key $G(0)$ to $G(1)$, but the subgroup key $SG_3(0)$ should also be renewed to $SG_3(1)$. Since $U_{15}$ does not know the keys of other subgroup $SG_1$ and $SG_2$, we can send the new group key $G(1)$ to $SG_1$ and $SG_2$ using their subgroup keys. Thus we need the following 2 encryptions to $SG_1$ and $SG_2$.

$$\left\{(G(1))_{SG_1(0)}, (G(1))_{SG_2(0)}\right\}. \tag{2.4}$$

Next, we should consider to send the new group key $G(1)$ to the members of the subgroup $SG_3$. $U_{15}$ knows the old subgroup key $SG_3(0)$, so it is not appropriate to send $G(1)$ encrypted by $SG_3(0)$. First, we generate a new subgroup key $SG_3(1)$, and send $SG_3(1)$ encrypted by each member's public key. Thus we need the following 4 encryptions to the members of $SG_3$.

$$\{(SG_3(1))_{O_i}\}_{i=11,...,14}. \tag{2.5}$$

Then, using this new subgroup key $SG_3(1)$, we can encrypt the new group key $G(1)$ as

$$(G(1))_{SG_3(1)}. \tag{2.6}$$

Thus, summing up (2.4), (2.5) and (2.6), we can get the total number of encryptions required for the leave of $U_{15}$,

$$A_{15} = 7. \tag{2.7}$$

Compare to (2.2), the number of encryptions is decreasing when we introduce the concept of subgroup.

Table 1: The number of encryptions

| | $A_{15}$(Leave) | $B_{16}$ (Join) | Total |
|---|---|---|---|
| Without SG | 14 | 2 | 16 |
| With SG | 7 | 4 | 11 |

Now let us suppose $U_{16}$ joins the group. Assume the new participant $U_{16}$ will join to the subgroup $SG_3$. As in the previous section, since we can reuse the old group key $G(1)$, we have

$$\left\{(G(2))_{G(1)}, (G(2))_{O_{16}}\right\}. \tag{2.8}$$

Thus, only 2 encryptions are required to renew the group key. However, since we need to send the new subgroup key $SG_3(2)$ to the members of $SG_3$, we need 2 more encryptions:

$$\left\{(SG_3(2))_{SG_3(1)}, (SG_3(2))_{O_{16}}\right\}. \tag{2.9}$$

Together with (2.8) and (2.9), we need

$$B_{16} = 4, \tag{2.10}$$

for sending the new keys to each participants when $U_{16}$ joins the group. Note that compare to (2.3), the number of encryptions is increased because of the additional subgroup keys.

### 2.3. Encryptions and subgroups

We summarize the results of our example in Table 1. We can find the total number of encryptions decreases, but the number of encryptions at the join increases, due to the additional subgroup keys. If we use subgroups, we can reduce the number of encryptions of the group key, but at the same time the number of encryptions to send the subgroup keys is increased.

Thus, to see the effect of the subgroups, we need to take into account the number of participants and the number of subgroups. To do this, we need to establish a queueing model. By using the queueing theory, we will see how to estimate the number of encryptions in the next section.

## 3. Queueing Model

We make a queueing model to deal with the secure group communication. Note that the subgroup keys are used only for the delivery of the group key, but not for the communication inside the group. Thus, we assume the subgroup can be made independent of the member's attribute. Of course, we can use the subgroup keys for the communication inside the subgroup, if the members of the subgroup shares some special interest (e.g. a company is a group and a department is its subgroup). However, in the following, we assume the subgroup is purely for the delivery of the group keys.

Let $U_n$ be the $n$-th participant of the group, $T_n$ be the join (arrival) time of the $U_n$, and $S_n$ be the sojourn time of $U_n$ in the group. We assume the point process of the new participant's join is Poisson process with rate $\lambda$. Also, assume the sojourn time $S_n$ has independent and identical distribution $F(x) = P[S_n \leq x]$ with its mean $E[S_n] = 1/\mu$. There is no limit of the number of participants in the group.

**Remark 3.1** *The assumption of Poisson arrival is not always valid for secure group communications. In general, arrival processes highly depend on the contents. However, it is*

*known that the aggregation of independent rare events can be a Poisson process. So, if the decision of joining and leaving the group is independent to other users, it is safe to assume Poisson arrival. For example, Poisson arrival may be assumed for subscription to stored information like music or videos, and membership service to regular TV programs. Also, we will discuss the possibility to extend our solution for general arrival cases (see Section 5).*

We divide the group into $N$ subgroups, $(SG_i)_{i=1,...N}$. Let $L_i(t)$ be the number of members of subgroup $SG_i$, and $L(t)$ be the number of the participants to the group at time $t$.

When a new participant joins the group, the participants will be assigned to a subgroup with equal probability independent of any other event (Bernouilli trial). Thus, let $J_n$ be the index of the new participant $U_n$, then we have

$$P\{J_n = i\} = P\{U_n \in SG_i\} = \frac{1}{N}. \tag{3.1}$$

It is well-known that if we divide the Poisson process with Bernouilli trial, each stream is also independent Poisson process ([11] P.69). Thus, the arrivals to each subgroup can be regarded as the independent Poisson process with rate $\lambda/N$, and $L_i(t)$ is the number of customers at time $t$ of the $M/G/\infty$ queue with its service time distribution $F(x)$ (for example, see [8]). The customers of this queueing system receive service immediately at the arrival and leave the system when the service is finished.

In the equilibrium state, the steady state distribution of the number of members $L_i(t)$ of the $M/G/\infty$ queue at arbitrary time is the Poisson distribution with mean $\lambda/(\mu N)$ [8]. Since the arrival rate to the subgroup $SG_i$ is $\lambda/N$, we have

$$P\{L_i(t) = n\} = \frac{1}{n!}\left(\frac{\lambda}{\mu N}\right)^n e^{-\lambda/(\mu N)}. \tag{3.2}$$

Since the arrival stream to each subgroup is independent, $\{L_i(t)\}_{i=1,...,N}$ has independent and identical Poisson distribution.

The mean number of members in a subgroup is obtained by

$$E[L_i(t)] = \frac{\lambda}{N\mu}, \tag{3.3}$$

and the mean number of participants to whole group is obtained by

$$E[L(t)] = \sum_{i=1}^{N} E[L_i(t)] = \frac{\lambda}{\mu}. \tag{3.4}$$

## 4. The Number of Encryptions

We use the queueing model $M/G/\infty$ to estimate the number of encryptions.

Let $G(t)$ be the group key and $(SG_1(t), ..., SG_N(t))$ be the subgroup keys at time $t$. In the following, we assume the functions $G(t)$ and $SG_i(t)$ are right-continuous. Note that $G(t)$ has jumps at arrivals $\{T_n\}$ and departures $\{D_n = T_n + S_n\}$. Also, the function $SG_i(t)$ has jump at the arrival and departure of the subgroup.

### 4.1. Leaving the group

Let us consider the case when a participant $U_n$ leaves the group at the time $D_n$. Since $U_n$ has been a member of $SG_{J_n}$, two keys, the group key $G(D_n-)$ and subgroup key $SG_{J_n}(D_n-)$, should be renewed. However, unlike the joining case shown in Section 4.2, $U_n$ knows $G(D_n-)$ and $SG_{J_n}(D_n-)$. So, we should follow the procedure below to renew the two keys.

1. The key server generates a new group key $G(D_n)$ and a new subgroup key $SG_{J_n}(D_n)$.
2. The key server encrypts the new subgroup key with the public key $O_k$ of each member in the subgroup $SG_{J_n}$,
$$\{(SG_{J_n}(D_n))_{O_k}\}_{k=1,\ldots,L_{J_n}(D_n+)},$$
and send them to each member who remains in the subgroup $SG_{J_n}$ immediately after the time $D_n$. Here, $k$ is the index of members of $SG_{J_n}$ at $D_n+$.
3. The key server can use the subgroup keys to encrypt the new group key $G(D_n)$,
$$\{(G(D_n))_{SG_i(D_n)}\}_{i=1,\ldots,N},$$
and send them to the each participants.

Letting $A_n$ be the number of encryptions required for the leave of $U_n$, we have
$$A_n = L_{J_n}(D_n+) + N, \tag{4.1}$$
where $L_{J_n}(D_n+)$ is the number of members of $SG_{J_n}$ immediately after the leave of $U_n$[1]. In general, if we have $M$ layers and $N_m$ subgroups in the $m$-th layer, we have
$$A_n = L_{J_n}(D_n-) + \sum_{m=1}^{M} N_m. \tag{4.2}$$

### 4.2. Joining the group

Let us assume a new participant $U_n$ joins the group at the time $T_n$ . Since $U_n$ joins a subgroup $SG_{J_n}$, the group key $G(T_n-)$ and the subgroup key $SG_{J_n}(T_n-)$ should be renewed at the time $T_n$ . Under the assumption that $U_n$ does not know two key $G(T_n-)$ and $SG_{J_n}(T_n-)$, we can renew them according to the following procedure.

1. The key server generates a new group key $G(T_n)$ and a new subgroup key $SG_{J_n}(T_n)$.
2. The key server encrypts the new group key with the old group key,
$$(G(T_n))_{G(T_n-)},$$
and send it to the $L(T_n-)$ participants who are already in the group just before the arrival of $U_n$.
3. The key server encrypts the new subgroup key with the old subgroup key,
$$(SG_{J_n}(T_n))_{SG_{J_n}(T_n-)},$$
and send it to the $L_i(T_n-)$ members who are already in the subgroup just before the arrival of $U_n$.
4. The key server use the public key of $U_n$ to encrypt as
$$\{(G(T_n))_{O_n}, (SG_{J_n}(T_n))_{O_n}\}$$
and send it to $U_n$.

Let $B_n$ be the number of encryptions at the join of $U_n$. Then, we have
$$B_n = 4, \tag{4.3}$$
which is independent of the number of subgroups and members.

In general, if we make more layers of sub-subgroups inside a subgroup, we have
$$B_n = 2M, \tag{4.4}$$
where $M$ is the number of layers.

[1]When a customer leaves a subgroup and the subgroup becomes empty, we do not need to encrypt a new group key with the subgroup key. However, for simplicity, we assume that the group key is encrypted with the subgroup key even in such cases.

### 4.3. Optimal number of subgroups

Now we consider a problem to find the optimal number of subgroups, which minimizes the mean number of encryptions.

From (4.3) and (4.1), we can see the number of encryptions depends on the number of subgroups and the number of members to each subgroups. Especially, the number of encryptions at the join is constant, so it is sufficient to estimate the one for the leave to get the optimal number of subgroups. Thus, we will find the number of subgroups $N^{min}$, which minimizes the mean number of encryptions at the $n$-th leave, $E[A_n]$. Note that $A_n$ is the burst workload and critical performance index for the key server.

From (4.1), the expectation of $A_n$ can be obtained by

$$E[A_n] = N + E[L_{J_n}(D_n+)]. \tag{4.5}$$

It is well-known in the queueing theory that if a system allows only discontinuous changes of size (plus or minus) one at the arrival and departure, then the probability distribution of the number of customers in the system seen by the arrivals is equal to the one left behind by departure (see [8], P176). Thus,

$$P[L_{J_n}(D_n+) = k] = P[L_{J_n}(T_n-) = k]. \tag{4.6}$$

Further more, by Poisson Arrivals See Time Average (PASTA)([14] p.294), Poisson arrivals see the same distribution of the number of customers in the system at arbitrary time, i.e.,

$$P[L(t) = k] = P[L_{J_n}(T_n-) = k]. \tag{4.7}$$

Thus, by (3.2), $P[L_{J_n}(D_n+) = k]$ is also Poisson distribution and its mean is obtained by $\lambda/(N\mu)$. Hence, we have

$$E[A_n] = N + \frac{\lambda}{N\mu}. \tag{4.8}$$

By a simple calculation, we can minimize $E[A_n]$ when $N = (\lambda/\mu)^{1/2}$. So, the optimal number of subgroups can be obtained by

$$N^{min} = \left(\frac{\lambda}{\mu}\right)^{1/2} = (E[L(t)])^{1/2}. \tag{4.9}$$

Thus, the optimal number of the subgroups is the square root of the expected number of the participant to the whole group.

In addition, from (4.1) and the fact that $P[L_{J_n}(D_n+) = k]$ is also Poisson distribution, we can obtain the distribution of the number of encryptions by

$$\begin{aligned} P[A_n^{min} = k] &= P[L_i^{min}(t) = k - N^{min}] \\ &= \frac{1}{\{k - (\lambda/\mu)^{1/2}\}!}\left(\frac{\lambda}{\mu}\right)^{\{k-(\lambda/\mu)^{1/2}\}/2} e^{-(\lambda/\mu)^{1/2}}, \end{aligned} \tag{4.10}$$

for $k \geq N^{min} = (\lambda/\mu)^{1/2}$. Also, the mean optimal number of encryptions is

$$E[A_n^{min}] = 2(E[L(t)])^{1/2}. \tag{4.11}$$

Here in Figure 1, we compare the no-subgroup model and the optimal subgroup model. We can see the clear advantage of the subgroup model in terms of the number of encryptions.

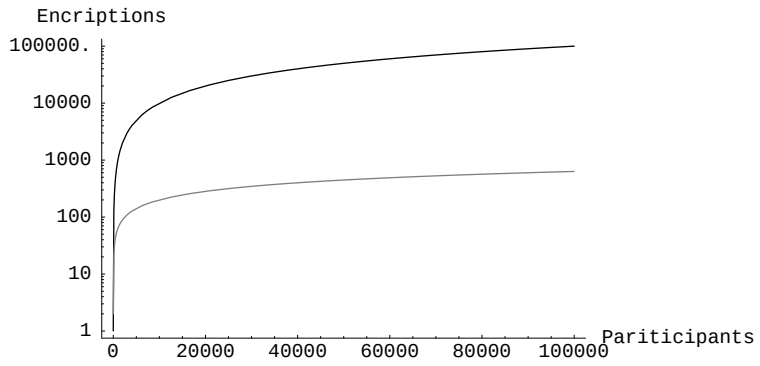


Figure 1: Log plot of the mean number of encryptions vs the number of participants: the upper line is the case of no subgroup, while the lower line is the case of the optimal number of subgroups.

Since the mean number $C$ of encryptions in a unit interval is given by

$$C = \lambda(E[A_n] + E[B_n]) = \lambda(E[A_n] + 4), \tag{4.12}$$

we have the optimal mean number of encryptions in a unit time $C^{min}$ as

$$\begin{aligned} C^{min} &= \lambda(E[A_n^{min}] + 4) \\ &= \lambda(2E[L(t)]^{1/2} + 4). \end{aligned} \tag{4.13}$$

Finally, consider the group has $M$ layers of subgroups. Let $N_i^{min}$ be the optimal number of subgroups in $i$-th layer. We can easily obtain the optimal number,

$$N_1^{min} = ... = N_M^{min} = \left(\frac{\lambda}{\mu}\right)^{1/M} = (E[L(t)])^{1/M}. \tag{4.14}$$

**Remark 4.1** *In general, the encryption workload using public keys is longer than the one using secret keys. Set the encryption time using secret key to be a unit time, and the encryption time using public key to be $\alpha$. Then, as similar to (4.8), we have*

$$E[W_n] = N + \alpha\frac{\lambda}{N\mu}, \tag{4.15}$$

*where $W_n$ is the workload required for the encryptions at the leave. Thus, the optimal number of subgroups to minimize the total workload can be obtained by*

$$N^{min} = \left(\frac{\alpha\lambda}{\mu}\right)^{1/2} = (\alpha E[L(t)])^{1/2}. \tag{4.16}$$

## 5. General Arrival Processes

In the previous sections, we obtained the optimal number of subgroups assuming Poisson arrival of users. In this section, we discuss the possibility of extension of our result to more

general arrival processes. Since $L(D_n+) = \sum_{i=1}^{N} L_i(D_n+)$, and the subgroup to be joined is determined independently to anything else, we have

$$E[L_{J_n}(D_n+)] = \frac{E[L(D_n+)]}{N} \tag{5.1}$$

Thus, for general arrivals, the optimal condition (4.9) can be replaced by

$$N^{min} = (E[L(D_n+)])^{1/2}. \tag{5.2}$$

Hence, to show that (4.9) still works for general arrivals, we need to estimate the difference between the event average $E[L(D_n+)]$ and the time average $E[L(t)]$ for general arrival processes. More precisely, let $A^{min}$ be the number of encryptions for the optimal case $N = E[L(D_n+)]^{1/2}$, and let $A^P$ be the number of encryptions when we set $N = E[L(t)]^{1/2}$. We will show the relative difference between $A^{min}$ and $A^P$ can be small for the large group, i.e.,

$$\frac{E[A^P] - E[A^{min}]}{E[A^{min}]} \to 0, \text{ as } E[L(t)] \to \infty. \tag{5.3}$$

Since $E[A^P] = E[L(t)]^{1/2} + E[L(D_n+)]/E[L(t)]^{1/2}$ and $E[A^{min}] = 2E[L(D_n+)]^{1/2}$, we have

$$\frac{E[A^P] - E[A^{min}]}{E[A^{min}]} = \frac{\left\{E[L(t)]^{1/2} - E[L(D_n+)]^{1/2}\right\}^2}{2E[L(D_n+)]^{1/2}E[L(t)]^{1/2}}. \tag{5.4}$$

Now, recall the definition of the (conditional) stochastic intensity of point processes [9]. Let $T_n$ be the time of $n$-th join and $A(t)$ be the number of joins in $(0, t]$, then the stochastic intensity of $A(t)$ given $L(t)$ is defined by

$$\lambda(t) = \lim_{s \to 0} \frac{E[A(t+s) - A(t)|L(t-)]}{s} = \lim_{s \to 0} \frac{P[A(t+s) - A(t) = 1|L(t-)]}{s}. \tag{5.5}$$

Intuitively, $\lambda(t)$ is the stochastic intensity conditioned by the number of users in the group just before arrivals. Note that the ordinary stochastic intensity is defined to be conditioned by the history of system up to $t$, $\mathcal{F}_t = \sigma(L(s-), A(s); s \le t)$ [1], but the above definition is sufficient for our purpose. By using the heuristic argument similar to [7], for a small $s$

$$\begin{aligned} E\left[L(t-)|A(t,t+s] = 1\right] &= \frac{E\left[L(t-)1_{\{A(t,t+s]=1\}}\right]}{P\left[A(t,t+s] = 1\right]} \\ &= \frac{E\left[L(t-)E[1_{\{A(t,t+s]=1\}}|L(t-)]\right]}{P\left[A(t,t+s] = 1\right]}. \end{aligned}$$

Letting $s \to 0$ on both sides, we have

$$E\left[L(T_n-)\right] = \frac{E\left[L(t-)\lambda(t)\right]}{\lambda}, \tag{5.6}$$

where $\lambda = E[\lambda(1)] = E[A(1)]$ is the intensity of $A(t)$. The equation (5.6) is known to be Papangelou's formula [7, 1]. Using (5.6), we can obtain the so-called cross-covariance equation [9, 10];

$$E[L(T_n-)] - E[L(t)] = Cov\left[L(t-), \frac{\lambda(t)}{\lambda}\right]. \tag{5.7}$$

Note that (5.7) holds for all processes which has the stochastic intensity.

Now, we consider the reverse process. Let $D(t)$ be the left-continuous counting process of leaves, i.e. the number of leaves during $[0,t)$. By reversing the sample path of process $L(t)$, the leaves of original process are considered to be joins to the reversed process. Define the reversed stochastic intensity $\lambda_D(t)$ of $D(t)$ by

$$\lambda_D(t) = \lim_{s\to 0} \frac{E[D(t) - D(t-s)|L(t+)]}{s}. \tag{5.8}$$

Intuitively, $\lambda_D(t)$ can be regarded as the stochastic intensity conditioned by the number of users left behind at the leave. Then by using (5.7) for this reversed process, we have

$$E[L(D_n+)] - E[L(t)] = Cov\left[L(t+), \frac{\lambda_D(t)}{\lambda}\right], \tag{5.9}$$

since $E[D(1)] = E[A(1)] = \lambda$ for stationary systems. For simplicity, let $L = E[L(t)]$ and $C = Cov\left[L(t+), \lambda_D(t)/\lambda\right]$. Then, by using (5.9), we can rewrite (5.4) as

$$\frac{E[A^P] - E[A^{min}]}{E[A^{min}]} = \frac{\left\{1 - (1 + C/L)^{1/2}\right\}^2}{2(1 + C/L)^{1/2}}. \tag{5.10}$$

Since in general $|Cov(X,Y)| \le Var[X]^{1/2} Var[Y]^{1/2}$, we have

$$|C/L| \le \frac{Var[L(t)]^{1/2}}{L} Var[\lambda_D(t)/\lambda]^{1/2}, \tag{5.11}$$

where we used the fact $Var[L(t+)] = Var[L(t)]$. Note that the second term $\lambda_D(t)/\lambda$ has already been normalized and stable relative to $L$, so the first term should be estimated. Now we assume the following;

$$\frac{Var[L(t)]^{1/2}}{L} \to 0 \text{ as } L \to \infty. \tag{5.12}$$

This assumption is valid if there is some kind of independence among the users in the group. For example, when the group is consisted by $M$ independent subgroups, then we can show (5.12) by taking $M \to \infty$. Under the assumption (5.12), using the fact $C/L \to 0$ as $L \to \infty$ in (5.10), we can obtain (5.3).

**Remark 5.1** *Even for a small group, if the covariance appeared on the right hand side of (5.9) is equal to* 0 *or at least sufficiently small, then (4.9) is a good engineering solution. This can be attained by the following cases;*

1. *A group with frequent joins and leaves, which results a large* $\lambda$ *and small (5.9).*
2. *If the arrival process is simple and the sojourn time of each user is independent, then* $E[L(D_n+)] = E[L(T_n-)]$, *and we have* $Cov[L(0-), \lambda(0)] = Cov[L(0+), \lambda_D(0)]$. *In this case, when the arrival stream is stable relative to* $L(t)$, *then we have a small* $Cov[L(0-), \lambda(0)/\lambda]$. *The condition* $Cov[L(0-), \lambda(0)] = 0$ *is known to be the lack of bias assumption (LBA), and is intensively studied to find the processes with LBA in the context of ASTA (arrivals see time-average) [3, 9, 4, 10, 7].*

Table 2: The number of encryptions

| Number of SGs | $E[A_n]$ (leave) | $C$ (encryption rate) |
|---|---|---|
| No SG | 10000 | $3334 \times 10^3$ |
| 10 | 1010 | $338 \times 10^3$ |
| 100 | 200 | $68 \times 10^3$ |
| 1000 | 1010 | $338 \times 10^3$ |

## 6. Numerical Example

Let us consider the pay TV on the internet for an example. Suppose we can expect the mean number of the participants is $10,000$. Assume each participants will remain the group 30 minutes on the average. Thus, we have $\mu = 1/30$, $\lambda = \mu E[L] = 1000/3$ for the $M/G/\infty$ queue. By (4.9), we can estimate the optimal number of subgroups,

$$N^{min} = 10000^{1/2} = 100. \tag{6.1}$$

As described in the previous section, the expected number of encryptions at the leave $E[A_n^{min}]$, which corresponds the burst workload for the key server, can be obtained by

$$E[A_n^{min}] = 200. \tag{6.2}$$

By (4.13), the average encryption rate $C^{min}$ is

$$\begin{aligned} C^{min} &= \frac{1000}{3}(2 \cdot 100 + 4) \\ &= 68000. \end{aligned} \tag{6.3}$$

We summarize the result of the optimal case and compare it with the cases of the non-optimal number of subgroups in Table 2. As you see in Table 2, for the large group such as the multicast group of the size $10,000$, the number of the subgroup has the great significance in terms of the performance of the key server.

Next, we will show the advantage of our optimal solution in the sense of distribution of the number of encryptions as well as its mean. Since the number of encryptions for non-optimal cases is also Poisson distribution as discussed in Section 4.3, we can compare the distribution among the different number of subgroups. In Figure 2, we show the distribution of optimal solution and non-optimal cases. Of course, there is a chance to have a large number of encryptions in the optimal case, but as you can see in the Figure 2 the large number of encryption happens less likely compared to the non-opimal cases.

## 7. Conclusion

Introducing the subgroup concept into the group communication such as Pay TV, we can make a scalable secure group communication. Further, we showed that the number of subgroups should be the square root of the expected number of participants, since it minimizes the number of encryptions at the key server.

Dynamic control of the subgroup and the efficient subgrouping considering network topology might be the future work.

## Acknowledgement

We thank the referees for helpful comments, especially the suggestion to consider Section 5.

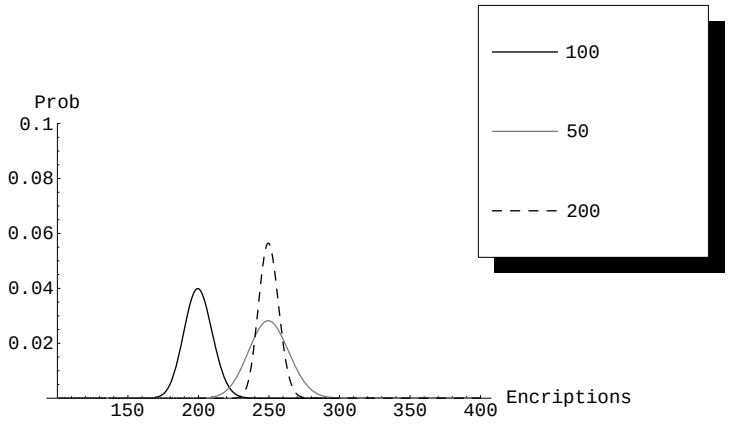


Figure 2: Probability distribution of the number of encryptions at the leave: Each line corresponds to the different number of subgroups. The optimal solution is 100 in this case.

## References


[1] F. Bacceili and P. Bremaud: *Elements of Qeueing Theory* (Springer-Verlag, 1994).

[2] W. Diffie and M. E. Hellman: New directions in cryptography, *IEEE Transactions on Information Theory*, **22- 6** (1976) 644–654.

[3] M. El-Taha and Jr. S. Stidham: *Sample-path Analysis of Queueing Systems* (Kluwer's international press, 1999).

[4] P. Glynn, B. Melamed, and W. Whitt: Estimating customer and time averages, *Operations Research.*, **41** (1993) 400 – 408.

[5] H. Harney and C. Muckenhirn: Group key management protocol (gkmp) sepcification, *The Internet Engineering Task Force RFC 2093*, (1997).

[6] H. Harney and C. Muckenhirn: Group key management protocol (gkmp) architecture, *The Internet Engineering Task Force RFC 2094* (1997).

[7] R. Kannupratti P. Bremaud and R. Mazumdar: Event and time averages: a review, *Advances in Applied Probability*, **24** (1992) 377 – 411.

[8] L. Kleinrock: *Queueing Systems Vol. 1* (John Wiley and Sons, 1975).

[9] B. Melamed and D. Yao: The Asta Property, *Advances in Queueing Theory, Methods and Open Problems*, J.H. Dshalalow, Ed., (CRC Press, 1995), 195–224.

[10] B. Melamed and W. Whitt: On arrival tha see time averages: a martingale approach, *Journal of Applied Probability*, **27** (1990) 376 – 384.

[11] S. M. Ross: *Stochastic Processes* (John Wiley and Sons, 1996).

[12] Stephen A. Thomas: *SSL and TLS Essentials: Securing the Web* (John Wiley and Sons, 2000).

[13] D. Wallner, E. Harder, and R. Agee: Key management for multicast: Issues and architectures, *The Internet Engineering Task Force RFC 2627* (1999).

[14] R.W. Wolff: *Stochastic Modeling and the Theory of Queues* (Princeton-Hall, 1989).

[15] C.K. Wong, M. Gouda, and S.S. Lam: Secure group communications using key graphs, *IEEE/ACM Transactions on Networking*, **8-1** (2000) 16–30.

Hiroshi Toyoizumi
University of Aizu
Performance Evaluation Lab.,
Tsuruga, Ikki-machi, Aizu-Wakamatusu,
Fukushima, Japan 965-8580
E-mail: toyo@u-aizu.ac.jp.