Secure Real-time User Preference Collection for Broadcast Scheduling

Xuhua Ding, Shuhong Wang,Baihua Zheng,
School of Information Systems
Singapore Management University
Email: {*xhding,shwang,bhzheng*}*@smu.edu.sg*

Abstract

Efficient broadcast scheduling is essential to the performance of wireless data broadcast systems. Existing algorithms for broadcast scheduling are mostly based on the knowledge of users' data access pattern. Unfortunately, the requirement of exposing individual preference profile becomes a serious threat to user privacy. In this paper, we investigate the issue of securely collecting user access patterns in real-time for broadcast scheduling. We propose a novel secure user profile collection protocol which protects the privacy of individual users yet facilitates efficient wireless data broadcast scheduling. To address the crucial issue of power conservation in mobile devices, our scheme does not rely on expensive public key cryptography. Light computation and communication at the user end makes the scheme feasible for mobile devices with limited resource. Our theoretical security analysis shows that the proposed protocol preserves user privacy against eavesdroppers and malicious broadcast servers. Moreover, our extensive performance evaluation experiments show that the proposed scheme has low computation and communication cost.

1. Introduction

The landscape for mobile computing is rapidly changing. As wireless access becomes faster, cheaper, more reliable and ubiquitous, the wireless coverage is increasing rapidly. Information service providers (ISPs) are vying for customers by delivering or enhancing their data services wirelessly. In order to save operational cost and provide better services, most ISPs outsource their information services to a wireless telecommunication carrier.

The specific wireless technologies in use vary from application to application. However, the underlying approaches to delivering information in wireless applications are either point-to-point connection or broadcast. Different from point-to-point connection which allocates an exclusive channel for one client, broadcast approach publishes data on a public channel shared by all the clients. As a result, a simultaneous access is enabled. Due to its constant cost and high scalability, broadcast approach is the ideal dissemination approach for information services with a huge number of potential users. One typical application of wireless data broadcast is the real-time stock quote update. A broadcast server periodically broadcasts the up-to-date quotes of stocks via the wireless channel to guarantee that each user can retrieve his desired information.

To retrieve data items, a user monitors (i.e., receive and check) the broadcast channel until his desired data items arrive. The response time, i.e., the duration between a user starting to listen to the channel and receiving all his interested data items, is a common metric to evaluate the performance of a broadcast system. Broadcast performance is of paramount importance due to the fact that most users are only interested in a tiny portion of a large data set. Seemingly, a flat broadcast approach whereby all the objects share the same priority in the wireless channel can produce the best average performance for all users. Nonetheless, data items are not uniformly accessed in most, if not all, real applications. As pointed out by the *Pareto's Principle*, 20% of the data items can satisfy 80% of the requirements from the clients.

Therefore, scheduling algorithms are motivated to achieve the near optimal response time by adjusting the broadcast program according to the collective interests of current clients. Various approaches have been proposed in the literature [2, 3, 13, 14, 15]. To the best of our knowledge, all existing scheduling algorithms are based on users' access preferences on data items. They either assume that the access patterns of users are known to the server, or require the users to explicitly upload their preference profiles. Both approaches are at the cost of user privacy. On the other hand, due to the widespread public awareness on privacy, the

concerns on personal privacy are growing. In our stock quote broadcast example, the subscribers may not want to share with the broadcast server what stocks they are interested in, especially when the server is managed by an untrusted network carrier. The privacy concern dampens users's willingness to subscribe services from ISPs.

We observe that an efficient wireless information service with user privacy preservation is highly desirable, as privacy usually tops customers' concern list when they sign up for services. Unfortunately, how to protect user preference privacy in wireless broadcast scheduling has not been addressed in the literature. A naive solution is to introduce an online trusted third party (TTP). Instead of informing the broadcast server access preferences, users send their information to TTP, which gathers all the preferences and sends the accumulated result to the broadcast server. However, this approach has two fatal drawbacks. First, it is impractical to require a large amount of heterogenous mobile users to trust a common entity. Second, as in all other applications with online TTPs, the TTP naturally becomes a single point of failure in terms of both performance and security. A TTP is usually chosen as the target of attacks by adversaries. Therefore, relying on an online TTP is not an appropriate solution. Another intuitive approach is to make use of so-called "anonymizers", like Tarzan[12] or Mix[9]. This approach requires additional servers or interactions among peer nodes. Considering the broadcast servers are typically base stations of a cellular network, adding new servers will significantly change the communication infrastructure. Therefore, an anonymous network is not a satisfying approach either.

Contribution In this paper, we propose a novel user preference collection scheme which allows a broadcast server to aggregate the information needed by a scheduling algorithm, while an individual user's preferences are not exposed. Specifically, the main contribution of this paper is three-fold.

- Our study is the first addressing user preference privacy in the wireless scheduling settings.
- A secure and practical user preference collection scheme is proposed with low computation and communication cost. Neither additional server nor peer interactions are required in our solution.
- A detailed analysis on the security, communication cost and computation cost of our scheme is provided. The analysis is verified and complemented by our experiment results.

Related Work

Theoretically, the problem of secure user profile collection can be solved by a secure e-voting scheme, whereby each user casts his votes on every data item and broadcast server tallies all votes. However, we argue that it is infeasible in practice. Many electronic voting schemes have been proposed, e.g. [4, 6, 7]. Basically, there are two approaches. One approach is to use mix-net, as in [1, 16, 19]. Nonetheless, it requires a number of additional mix servers between subscribers and the broadcast server so that a significant change has to be made on the underlying communication infrastructure. The other approach, as used in [4, 6, 18], is to employ homomorphic encryptions [10, 17] and threshold decryption [11]. Unfortunately, these schemes incur a heavy computation load on the voters and require the collaboration among multiple servers, which is unrealistic for a broadcast system disseminating hundreds of data items to mobile devices.

The computation paradigm of our problem has similarity to data aggregation in wireless sensor networks, where the data collected by sensors are encrypted, aggregated and sent to a sink node. However, the existing schemes, e.g. [8, 22], protect data confidentiality only against eavesdroppers. The sink node has the collection of keys and is able to decrypt individual ciphertext, which is undesirable in our scenario.

Recent work [23] by Yang et. al is very close to ours. Their scheme allows a data miner to anonymously collect data from a group of users. The basic idea is to use re-encryption, which has been widely used in mix networks. We argue that this approach is not a solution to our problem since it requires t rounds of communication between the miner and a set of t users. The incurred communication cost is too high for real-time broadcast scheduling. Moreover, re-encryption incurs multiple modular exponentiations. Considering the number of data items involved and the limited computation resource in mobile devices, the computation cost is prohibitively high.

Organization The rest of the paper is organized as follows. In Section 2, the problem is formally defined with all the challenges we are facing. The detailed secure user profile collection approach is described in Section 3. In Section 4 and Section 5, we analyze the security and performance of the proposed approach respectively. Finally, we conclude this paper in Section 6 and point out the future work.

2. Problem Formulation

In this section, we provide a detailed description of the problem settings.

System Model Throughout this paper, we consider a wireless data broadcast system consisting of three types of entities:

- Service Registration Server(SRS): SRS locates at the ISP's site and takes the charge of system initialization and service subscription. For initializing a system, SRS determines the set of data items to broadcast, based on the specific application. The size of the data set is application-dependent, varying from a few hundred to a few thousand. It is also responsible for the selection of the scheduling algorithm, used by BS in the broadcast program, and an integer range for preference specification, e.g. [0,1,. . . ,10]. A larger integer indicates a stronger interest. SRS passes the system parameters to a wireless carrier, whose base stations deal with the actual data delivery. For service subscription, it acknowledges the request from a new user via passing the user a set of parameters.
- Broadcast Server(BS): A BS is a base station operated by a wireless carrier. It periodically delivers the data set to subscribers in its domain. Before scheduling a broadcast, BS requests all its receivers to upload their preferences. With subscribers' collective preferences, BS sorts out the data set and assigns priorities to each item. Then, it broadcasts the data set accordingly. The issues of scheduling algorithm and data delivery are out of the scope of this paper.
- Subscribers: A subscriber is an end-user subscribing the broadcast service[1]. Registration at the SRS is required for a subscriber to receive data broadcast from BS. Each subscriber independently determines his own user interest profile, i.e., a list of desired items with corresponding preferences in the range defined by SRS. All preferences are independent of each other.

Trust Model We assume that SRS is fully trusted whereas BS is not. BS may take advantage of its participation in the scheme and attempt to compromise end-users' privacy. The end-users are assumed to be honest. They do not collude with malicious BS. We assume the channel between a user and BS is authentic and reliable. However, we do not assume the confidentiality of the broadcast channel. An eavesdropper is able to sniff all traffic. In addition, BS is able to obtain the subscriber's information, e.g their identities, locations and SIM card numbers. Therefore, no communication source anonymity is provided in existing networking infrastructure.

Objectives A secure user profile collection has two objectives. First, BS is able to compute the sum of all subscribers' preferences on every data item, which is required by the scheduling algorithm. Second, the protocol should preserve every subscriber's preference privacy. In other words, a user's *individual* preferences should not be exposed to either BS or other end-users. It implies that an adversary is unable to determine, except by random guessing, whether a data item is preferred by a user.

CAVEAT User preference privacy is different from user data privacy. The latter prevents the adversary from knowing what data a user obtains from the communication channel. We remark that the nature of broadcast obviously offers user data privacy.

Requirements We list below the requirements of a profile collection protocol.

- **computation cost.** Most mobile devices, e.g. cellular phones, are powered by batteries and have very limited computation capacity. A secure preference aggregation scheme should not levy the devices's resource by imposing heavy computations. This requirement implies that many cryptographic techniques based on large number operations are not suitable for common mobile devices.
- **communication cost.** Note that communications also consume CPU cycles and battery power. Moreover, in many wireless networks, a device's upload channel has narrower bandwidth compared to its download channel. The volume of traffic and the number of rounds of communications are tightly constrained. Out of this concern, interactions among subscribers should be avoided. Furthermore, it is undesirable to introduce extra entities due to both cost and policy reasons. A practical preference collection scheme should be built on top of the existing network infrastructure.
- **accuracy.** The preference aggregation protocol should not degrade the performance of the existing broadcast schedule algorithm. A broadcast service may cover hundreds of data items. Ideally, the protocol is able to count every user's preference on every data item.

Table 1 summarizes the notations used in the following description.

1 Without causing ambiguity, we use subscribers and users alternatively in the rest of this paper.

Notation	Description
U_x	User with identity $x \in \mathcal{Z}^+$
w_x	User U_x's secret key
$\mathcal{D}$	the set of data items to broadcast
N	the cardinality of $\mathcal{D}$, i.e. the number of items to broadcast
D_i	the i-th data item, $i \in [1, N]$
N_u	the number of users in BS's domain
$[0, 1, \cdots, s]$	the integer range of a preference on a data item
t	the threshold group size, i.e. the number of users in each group
G_i	the i-th threshold group of t users
$\|m\|$	the bit length of an integer m
$\mathcal{H}_K()$	a one-way collision-resistant keyed hash function with the secret key K

Table 1. Notations

3. Secure Preference Aggregation Protocol

The basic idea of our approach is similar to the jigsaw puzzle. Each individual piece of a jigsaw puzzle seems random, whereas the outcome from a correct assembly of them is expressive. In our approach, a group of users share a public data using a threshold secret sharing scheme. Each user's key share is kept secret by herself and serves as the encryption key to encrypt his preferences. The server collects users' preferences and aggregates user preferences on all data items to broadcast.

3.1. System Setup and User Registration

SRS is responsible for system setup and user registration. For an information service, SRS initializes $\mathcal{D} = \{D_1, D_2, \ldots, D_N\}$, the set of data items to broadcast. SRS also selects $[0, \ldots, s]$, the range of the user preference on a data item. It picks a security parameter $t \in \mathcal{Z}^+$ as the threshold group size, and selects a system parameter $Y \in \mathcal{Z}^+$ satisfying $Y > t \cdot s$. Then, it selects the smallest prime p satisfying $p > Y^{N+1}$. To use a t-out-of-t threshold secret sharing scheme [21], it initializes a $(t-1)$-degree polynomial $f(x) = \sum_{i=0}^{t-1} b_i x^i \bmod p$, where $b_1, b_2, \cdots, b_{t-1} \in_R \mathcal{Z}_p^*$ and $b_0 = 0$. A collision-resistant keyed hash function [5] $\mathcal{H}_K()$ is selected, where K denotes the hash key. SRS keeps $b_1, \ldots, b_{t-1}$ and K secret and all others are public.

Once a user subscribes to an information service at SRS, it receives a unique random $x \in_R \mathbb{Z}_p^*$ as its identity, a secret key $w_x = f(x)$, and the hash key K. The user keeps w_x and K secret and lets x be public. As an option, a user may download $Y, Y^2, \ldots, Y^{N-1}$ from SRS in order to save computation cost in future protocol executions.

3.2. The Protocol

The preference aggregation protocol consists of three steps.

1. Advertising: BS divides all users in its domain into groups of size t. It broadcasts to all users their membership together with the next time slot for preference uploading.
2. Reporting: Upon receiving the advertisement from BS, all users report their encrypted preferences to BS concurrently.
3. Aggregation: On receiving all t encrypted preferences from the same group, BS aggregates them and computes the sum of preferences for each data item.

Advertising Algorithm 1 describes the details of advertising, which is executed by BS when re-scheduling of the broadcast program is demanded[2]. Let N_u denote the number of users in BS's domain. They are partitioned into groups of size t. For user U_x, BS assigns c_x as U_x's membership token, which enables the aggregation of preferences from the same group. Finally, it broadcasts the results to all subscribers.

Algorithm 1 Advertising (By BS)

Input: threshold group size t, all users' identities;
Output: all users' group membership;
Procedure: (Executed by BS)

1: Divide all users into groups of size t. The following steps are with respect to every group.
2: Pick a new integer $G \in \mathbb{Z}^+$ as the identity of this group. Let the group members' identities be denoted by $x_1, x_2, \cdots, x_t$.
3: **for** $i = 1; i \leq t; i++$ **do**
4: calculate c_{x_i}:

$$c_{x_i} = \prod_{1 \leq k \leq t; k \neq i} \frac{x_k}{x_k - x_i} \bmod p \qquad (1)$$

5: output $< x_i, c_{x_i}, G >$
6: **end for**
7: Broadcast $\langle x_i, c_{x_i}, G \rangle$, for all $1 \leq i \leq N_u$ and T to all users, where T is the expected data broadcast time.

2 When the re-scheduling process is triggered is out of the scope of this paper. Generally, a base station can re-schedule the broadcast program periodically or based on the real application requirements.

Reporting The Reporting algorithm shown in Algorithm 2 below is executed independently by every user. Consider a user U_{x_i} in a threshold group G. Let a_j denote U_{x_i}'s preference on data item D_j, for $1 \leq j \leq N$. U_{x_i} first cumulates $a_1, \ldots, a_N$ into v_{x_i} in order to save communication cost. Note that a large portion of the preferences are zeroes. Therefore, the computation of v_{x_i} is not expensive. v_{x_i} is then padded and encrypted. The ciphertext z_{x_i} is sent to BS.

Algorithm 2 Reporting (by user U_{x_i})

Input: U_{x_i}'s preference set: $\{a_1, a_2, \ldots, a_N\}$, membership token c_{x_i}, group identity G, schedule time T, secret key w_{x_i}, hash key K.

Output: z_{x_i} as U_{x_i}'s reply to BS

Procedure:

1: Compute:
$$v_{x_i} = \sum_{j=1}^{N} a_j Y^{j-1} \tag{2}$$
$$z_{x_i} = v_{x_i} + c_{x_i}^{-1} w_{x_i} \mathcal{H}_K(T||G) \bmod p \tag{3}$$
2: Send $\langle x_i, z_{x_i} \rangle$ to BS.

Aggregation When BS receives the reports from all users, it starts Aggregation step as depicted in Algorithm 3. BS first evaluates Equation 4, which returns the aggregated preferences for all users from the same threshold group. Essentially, $\Gamma = \sum_{j=1}^{N} r_j Y^{j-1}$, where r_j is exactly the sum of all group members' preferences on data item D_j. Thereafter, by evaluating Equation 5 and Equation 6, the sums of preferences for all data items are recovered.

We show the correctness of the protocol. Since $U_{x_1}, U_{x_2}, \cdots, U_{x_t}$ are in the same threshold group G_i, we have $\sum_{k=1}^{t} c_{x_k} w_{x_k} = 0 \bmod p$ by virtue of the (t, t) threshold secret sharing scheme, which distributes the value 0. Therefore,

$$\begin{aligned}
\Gamma &= \sum_{k=1}^{t} z_{x_k} \bmod p \\
&= \sum_{k=1}^{t} v_{x_k} \bmod p + \mathcal{H}_K(T||G_i) \sum_{k=1}^{t} c_{x_k} w_{x_k} \bmod p \\
&= \sum_{k=1}^{t} v_{x_k} \bmod p \\
&= \left(r_1 + r_2 Y + r_3 Y^2 \cdots + r_N Y^{N-1}\right) \bmod p \\
\because\ p &> Y^{N+1} \text{ and } Y > s{\cdot}t \text{ and } r_j < s{\cdot}t,\ \forall j \in [1, N] \\
\therefore\ \Gamma &= r_1 + r_2 Y + r_3 Y^2 \cdots + r_N Y^{N-1}
\end{aligned}$$

where r_j is exactly the sum of $U_{x_1}, \cdots, U_{x_t}$'s preferences on data item D_j. The above computation justifies the requirements for selecting p and Y in Section 3.1.

Algorithm 3 Aggregation (by BS)

Input: threshold groups G_i, parameters c_x for all users $U_x \in G_i$,All z_x reported by all users;

Output: $R_1, R_2, \ldots, R_N$ as the collective preferences on $D_1, D_2, \ldots, D_N$ respectively;

Procedure:

1: Let $R_1, \ldots, R_N$ be all 0.
2: **for** each threshold group G_i, $1 \leq i \leq \lfloor N_u/t \rfloor$. Let their members be denoted by $U_{x_1}, U_{x_2}, \cdots, U_{x_t}$ **do**
3: BS computes
$$\Gamma = \sum_{k=1}^{t} z_{x_k} \bmod p \tag{4}$$
4: **for** $j = 1; j \leq N; j{+}{+}$ **do**
5:
$$r_j = \Gamma \bmod Y; \tag{5}$$
$$\Gamma = (\Gamma - r_j)/Y; \tag{6}$$
$$R_j = R_j + r_j \tag{7}$$
6: **end for** (8)
7: **end for**
8: Output $R_1, R_2, \cdots, R_N$ as the cell's collective preferences.

4. Security Analysis

In this section, we conduct both a theoretical discussion and an experimental simulation to analyze the information security of the proposed preference aggregation approach. Without loss of generality, the adversary fixes its target: a victim U_1 's preference on certain data item[3] D_j. Let U_1 and $U_2, \cdots, U_t$ form a threshold group. We use $a_1, a_2, \cdots, a_t$ to denote $U_1, \cdots, U_t$'s preferences on D_j respectively. Since a user is only interested in a small subset of $\mathcal{D}$, the adversary will claim his success if he correctly determines whether U_1 has interests in D_j. Namely, the adversary guesses whether $a_1 = 0$ or $a_1 \neq 0$. Note that this type of attack is much stronger than those attempting to determine the exact value of a_1. In the following, our discussion focuses on the privacy with respect to a_1. The conclusion is applicable to any other user preferences.

Notion of Privacy The notion of privacy is defined as the amount of related information revealed by the system to the adversary. Formally, let A be the discrete random variable for U_1's preference on the targeted data item. $A = 0$ iff $a_1 = 0$ and $A = 1$ iff $a_1 \neq 0$. Let $H(A)$ be the entropy of A. Let $H'(A)$ be the entropy of A based on the adversary's observation after mounting attacks. Note that $H'(A)$ captures the information obtained by an adversary regarding to A. Let ρ denote the security strength of our scheme with re-

3 Note that the preferences on different data items are completely independent from each other.

spect to a_1. It is defined as

$$\rho = H(A) - H'(A)$$

The semantic of ρ is the magnitude of information leakage regarding to whether $a_1 = 0$. It shows that approximately ρ-bit information is exposed to the adversary by the system.

Adversary In our adversary model, we consider a malicious BS, since it is more powerful than eavesdroppers. BS may exploit its participation in protocol execution. It is clear that from Algorithm 3, BS only obtains two pieces of information relevant to a_1: the user's report z_1, which is directly sent by U_1; and the sum of preferences $r = \sum_{i=1}^{t} a_i$ which is computed by BS in Equation 5.

We first prove in Lemma 4.1 that the adversary does not obtain additional information about users' preferences, except the sum. For the purpose of clarity, suppose that there exist $2t$ users in BS's domain, who are divided into two threshold groups G_1 and G_2. Lemma 4.1 and its proof also apply to multiple threshold groups. We regard the protocol as a batch encryption scheme, which takes a set of users' preferences $\{v_1, \cdots, v_{2t}\}$ as input and outputs $\{z_1, \cdots, z_{2t}\}$. Its secret key is in fact the $(t-1)$-degree polynomial $f(x)$ over $\mathbb{Z}_p$.

Let $\{x_1, \cdots, x_{2t}\}$ be the users' identities respectively. Let $\mathcal{T}_f$ denote the transcript of the protocol execution using $f(x)$. In specific, $\mathcal{T}_f = \{w_1, \cdots, w_{2t}, \mathcal{H}_K(T||G_1), \mathcal{H}_K(T||G_2)\}$. Let $\mathcal{V}_f$ denote the adversary's *view* of the protocol execution. In specific, $\mathcal{V}_f = \{z_1, \cdots, z_{2t}, \sigma_1, \sigma_2\}$, where $\sigma_1 = \sum_{i=1}^{t} z_i \bmod p = \sum_{i=1}^{t} v_i$ and $\sigma_2 = \sum_{i=1}^{t} z_{t+i} \bmod p = \sum_{i=1}^{t} v_{t+i}$.

Lemma 4.1 *Given a transcript $\mathcal{T}_f$ and a view $\mathcal{V}_f$ with respect to a polynomial $f(x)$ and a set of preference inputs $\{v_1, \cdots, v_{2t}\}$, the adversary is able to construct another $(t-1)$-degree polynomial $f'(x)$ over $\mathbb{Z}_p$ and another set of inputs $\{v'_1, \cdots, v'_{2t}\}$, such that (1) $\mathcal{V}_{f'} = \mathcal{V}_f$; (2) $\mathcal{T}_f$ and $\mathcal{T}_{f'}$ are indistinguishable.*

Proof: For the threshold group G_1, the adversary randomly chooses the inputs $\{v'_1, \cdots, v'_t\}$, such that $\sum_{i=1}^{t} v'_i = \sum_{i=1}^{t} v_i$. Then it picks $\bar{h} \in_R \mathbb{Z}_p^*$, and computes $w'_i = (z_i - v'_i)c_{x_i}^{-1}\bar{h}^{-1} \bmod p$, for all $i \in [1, t]$. Clearly, for $1 \le i \le t$,

$$z'_i = v'_i + \bar{h}w'_i c_{x_i} = z_i \bmod p$$

Now the adversary constructs $f'(x)$. Use $(x_1, w'_1), \cdots, (x_t, w'_t)$ as t points and calculate a $(t-1)$-degree polynomial $f'(x)$ by Lagrange Interpolation so that $f'(x_i) = w'_i$. It is clear to observe that $f'(0) = 0$ since $f'(0) = \sum_{i=1}^{t} w'_i c_i = \bar{h}^{-1} \sum_{i=1}^{t} (z'_i - v'_i) = \bar{h}^{-1}(\sum_{i=1}^{t} z_i - \sum_{i=1}^{t} v_i) = 0$.

Thus, set $w'_i = f'(x_i)$ for all $i \in [t+1, 2t]$. Then, the adversary selects a random $\hat{h} \in_R \mathbb{Z}_p^*$, satisfying $z_i - \hat{h}w'_i c_{x_i} \bmod p < Y^{N+1}$, for all $i \in [t+1, 2t]$. Compute $v'_i = z_i - \hat{h}w'_i c_{x_i} \bmod p$, for all $i \in [t+1, 2t]$. Thus,

$$\sum_{i=t+1}^{2t} v'_i = \sum_{i=t+1}^{2t} z_i - \hat{h} \sum_{i=t+1}^{2t} w'_i c_{x_i} \bmod p$$

Since $\sum_{i=t+1}^{t} w'_i c_{x_i} = f'(0) = 0$, $\sum_{i=t+1}^{2t} v'_i = \sum_{i=t+1}^{2t} z_i = \sigma_2$. Therefore, the $\mathcal{V}_{f'} = \mathcal{V}_f$.

$\mathcal{T}_{f'} = \{w'_1, \cdots, w'_{wt}, \bar{h}, \hat{h}\}$, which is indistinguishable from $\mathcal{T}_f$ when the hash function $\mathcal{H}_K(\cdot)$ is modelled as a pseudo-random number generator.
□

The Lemma above shows that the protocol execution does not reveal additional information about $\{v_1, \cdots, v_{2t}\}$ except the view $\mathcal{V}_f$, which exposes the sum of v_i-s. Therefore, BS obtains the sum of t user's preferences on all N data items.

Hereafter, we proceed to analyze the information leakage from knowing the sum of preferences for D_j. In other words, we analyze the inference of $r = \sum_{i=1}^{t} a_i$ with respect to a_1. For example, when $r = 0$, a_1 must be zero since all preferences are non-negative. To facilitate the discussion, we assume that in average every user is interested in n data items, and $n << N$. Therefore, A follows the distribution below:

$$\Pr(A = 0) = 1 - \frac{n}{N} \quad \text{and} \quad \Pr(A = 1) = \frac{n}{N}$$

Consequently,

$$H(A) = -\frac{n}{N}\log(\frac{n}{N}) - (1 - \frac{n}{N})\log(1 - \frac{n}{N})$$

Let random variable R represent $\sum_{i=1}^{t} a_i$, which takes on the integer set $\{0, 1, \cdots, st\}$. Because the adversary obtains the information $R = r$ during every protocol execution, $H'(A)$ is computed as the conditional entropy of A given R. Specifically,

$$H'(A) \triangleq H(A|R) = \sum_{r=0}^{st} \Pr(R = r) H(A|R = r)$$

where

$$H(A|R = r) = \\ - \sum_{i \in \{0,1\}} \Pr(A = i|R = r) \log(\Pr(A = i|R = r))$$

Hence,

$$\rho = H(A) - \sum_{r=0}^{st} \Pr(R = r) H(A|R = r) \qquad (9)$$

Next, we proceed to evaluate ρ. The challenge here is how to compute $\Pr(R = r)$ and $\Pr(A = 0|R = r)$. To evaluate $\Pr(A = 0|R = r)$, we introduce another random variable R' representing $\sum_{i=2}^{t} a_i$, i.e. $R = R' + a_1$. R' takes on the integer set $\{0, \cdots, s(t-1)\}$. Hence,

$$\begin{aligned} \Pr(A = 0|R = r) &= \Pr(A = 0, R = r)/\Pr(R = r) \\ &= \Pr(A = 0, R' = r)/\Pr(R = r) \end{aligned}$$

$\because A$ and R' are independent random variables

$$\therefore \Pr(A = 0|R = r) = \Pr(A = 0)\Pr(R' = r)/\Pr(R = r)$$

Since $\Pr(A = 1|R = r) = 1 - \Pr(A = 0|R = r)$, we are able to evaluate Equation 9, provided that $\Pr(R = r)$ and $\Pr(R' = r)$ are available.

Unfortunately, it is prohibitively complex to determine the probability distribution of R and R' by using the multinomial distribution and integer partition theories. Instead, we have to acquire the distribution of R and R' from the observation of 100,000 experiments. The distributions are shown in Figure 1.

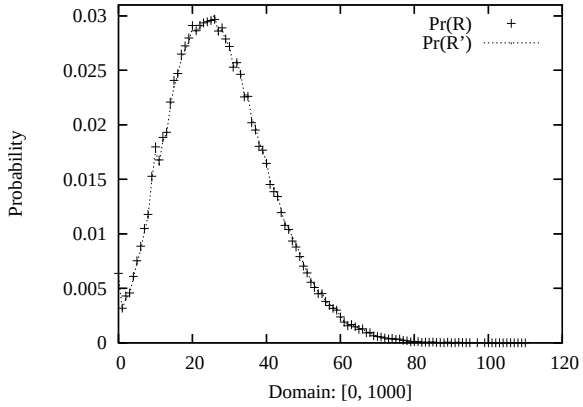

Figure 1. The distribution functions of R and R': $t = 100$, $N = 1000$, $n = 50$, $s = 10$

We ran the protocol using different threshold group size t. After obtaining the distribution of R and R', we compute ρ based on Equation 9. The results are shown in Table 2. For instance, it shows that our scheme only reveals approximately 0.0071-bit information when using $t = 100$ and $n = 50$. Moreover, it is evident that ρ decreases when t increases. It implies that a larger t will reduce the information leakage. The intuitive reason behind this is that when a threshold group is formed by more users, there are more randomness introduced and more possible partitions of the preference sum. Thus, inferring a_1 from the sum is less effective.

t	ρ ($n = 50$)	ρ ($n = 200$)
10	0.0625	0.0533
30	0.0249	0.0197
50	0.0146	0.0137
80	0.0086	0.0087
100	0.0071	0.0072
130	0.0057	0.0067
200	0.0037	0.0055

Table 2. The information leakage with different t and n ($N = 1000$).

In short, we model the notion of privacy of our scheme as the amount of exposed information regarding to a user's interest. We have shown that our scheme reveals insignificant information to the malicious broadcast servers when t is reasonably large.

Discussion on (t, t) Access Structure

In our scheme, all subscribers in BS's domain are divided into groups of size t. For each group, only when t reports from its members are available, can BS successfully run the Aggregation algorithm to compute the preference sum. Such a requirement is called a *t-out-of-t access structure*, or (t, t) *access structure*. One may observe that it does not provide reliability since if one member fails, the rest $t-1$ reports cannot be taken into account by BS. However, we argue that a flexible access structure will allow BS to mount *cross-set attacks* as explained below.

If BS obtains a preference sum r for a user set G and another sum r' for another set G', it computes $r - r'$, which is the difference of preferences for users in $G - G'$. The privacy for users in $G - G'$ is compromised when the size of $G - G'$ is small. In particular, if BS is allowed to compute the sum for any $t - 1$ users out of t users, the difference between two user's preference can be easily obtained by BS. Since a user's preference is not uniformly distributed across all data items, such a difference reveals their interests. A (t, t) structure ensures that there is no overlap between any legitimate user groups.

5. Performance Analysis

Computation Cost In order to minimize the computation load on wireless devices, our protocol avoids expensive public key cryptographic techniques. Our approach only incurs modular multiplications and additions. $Y, Y^2, \cdots, Y^{N-1}$ are computed by the system administrator when initializing the system. A user down-

loads them on the basis of her personal needs and stores them on her device. During protocol executions, the computation load for a user is the evaluation of Equation 2 and Equation 3. Since a user is only interested in n data items, it needs n multiplication and addition operations in evaluating Equation 2. We observe that since all a_i-s are small numbers bounded by s, the computation is cheap despite the fact that some Y^i-s are relatively large numbers. To evaluate Equation 3, the user executes two modular multiplications and one hash function. Therefore, the user totally evaluates 2 modular multiplications of $N(\log_2 t + \log_2 s)$ bits, 1 hash function, and n regular multiplications. To avoid huge modulus due to large N, one approach is to divide the whole data set into subsects, and run the same protocol over each subsect.

To show the performance on mobile devices, rather than PCs in our experiments, we use the results from E. Savas et.al. [20]. They measured the computation cost of modular multiplications by software and hardware on ARM processors, a popular processor in PDAs, with results shown in Table 3. This is a conservative estimation of the performance as modern mobile devices usually have a faster CPU than ARM 80MHz. In our experimental settings, $N = 1000, s = 10$, the modulus p could be as long as a few kilo-bits. To avoid using a huge modulus, we split the whole data set into smaller sets of size 100, so that the modulus is around 1024bits. According to Table 3, it may only take a number of milliseconds for an old-fashion mobile device to execute the protocol in our experimental setting.

Precision (bit)	Hardware(μs) (80MHz)	Software (μs) (on ARM with Assembly)
224	5.9	33.2
256	6.6	42.3
1024	61	570

Table 3. Execution time of hardware and software implementations of the $GF(p)$ multiplication[20]

In Algorithm 3, the broadcast server evaluates t modular multiplications in Equation 4. Since there are N_u/t groups of users, the broadcast server totally needs N_u modular multiplications to calculate all aggregated preferences on all data items. Considering the fact that modular operations and division operations have negligible cost for a regular PC, we dismiss this portion of cost for Equation 5 and 6. In our experiment, we implemented the broadcast server on a PC with 1GHz CPU and 1GB memory. It costs $14\mu s$ for a 1024-bit modular multiplication. Therefore, the computation load on the broadcast server side is only in milliseconds.

Communication Cost The protocol in Section 3 only requires one round communication. The data sent by a user in Algorithm 2 has the same size of the modulus p. Therefore, the total number of bits to send is $\log_2 p$ which is $O(N(\log t + \log s))$ bits. Note that for a preference collection scheme without privacy preservation, the communication cost is $O(n \log N + n \log s)$, which is much smaller than our cost. Therefore, the communication cost is mainly the price paid for privacy protection. We argue that transmission of a few thousand bits is still affordable for modern mobile devices.

Selection of t A larger t entails higher computation and communication cost. Both costs grow linearly with the bit length of t. However, considering N is not a small integer, attentions should be paid to select an appropriate t. Table 2 shows the relation between t and security strength. It is visualized in Figure 2 below. We observe that when t is around 100, increasing t does not provide significant enhancement on security. Therefore, we recommend to choose a 6- or 7-bit integer for t.

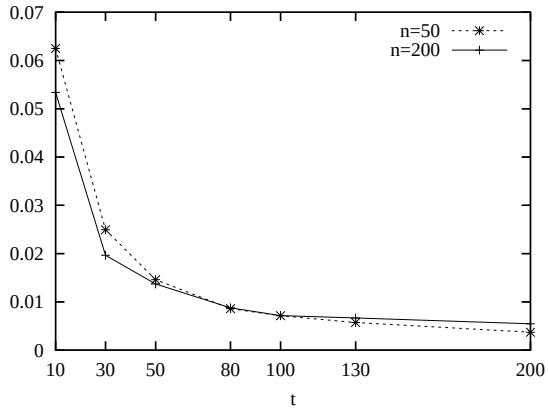

Figure 2. The relation between t and $\rho = H(A_0) - H'(A_0)$ ($N = 1000$)

6. Conclusion and Future Work

In this paper, we propose a new secure user profile collection protocol. Without compromising user privacy, the proposed scheme meets the demands by all broadcast scheduling algorithm. A broadcast server is

able to schedule a data broadcast based on collective preference profile, while each individual user's personal interests remain secret. From the protocol execution, the adversaries, either eavesdroppers or malicious broadcast servers, obtain insignificant amount of information. Our scheme does not involve expensive large number operations, such as modular exponentiations. With low computation and communication cost, our scheme is feasible for light-weight mobile devices with scarce computational and communication resource.

Nonetheless, our construction has two drawbacks. First, our scheme is built on top of (t, t) threshold secret sharing, which is not fault tolerant. A malfunction of one user will fail the preference aggregation for the related threshold group. However, it is demanded to have such rigid group access structure as pointed out in Section 3. It is an open problem how to harmonize security and reliability. Second, our scheme is vulnerable to collusion attacks by a malicious BS and a subscriber. Knowing the secret key K will expose more information about user preferences, even though the adversary is still unable to correctly determine a preference with an overwhelming probability.

Acknowledgement

We would like to thank Wei Wei for her help on implementation work. We are also grateful to anonymous reviewers for their constructive comments.

References

[1] M. Abe. Universally verifiable mix-net with verification work independent of the number of mix-servers. In *Advances in Cryptology -Eurocrypt'98*, pages 437–447, 1998.

[2] S. Acharya, R. Alonso, M. Franklin, and S. Zdonik. Broadcast disks: Data management for asymmetric communications environments. In *Proceedings of the ACM SIGMOD International Conference on Management of Data (SIGMOD'95)*, pages 199–210, May 1995.

[3] S. Acharya, M. Franklin, and S. Zdonik. Disseminating updates on broadcast disks. In *Proceedings of the 22nd VLDB Conference*, pages 354–365, September 1996.

[4] O. Baudron, P. Fouque, D. Pointcheval, G. Poupard, and J. Stern. Practical multi-candidate election system. In *Proceedings of the 20th Annual ACM Symposium on Principles of Distributed Computing (PODC'01)*, 2001.

[5] Mihir Bellare, Ran Canetti, and Hugo Krawczyk. HMAC: Keyed-hashing for message authentication. Technical Report 2104, February 1997.

[6] J. Benaloh. *Verifiable Secret-Ballot Elections*. PhD thesis, Yale University, 1987.

[7] J. Benaloh and D. Tuinstra. Receipt-free secret-ballot election (extended abstract). In *Proceedings of 26th Annual Symposium on Theory of Computing (STOC'94)*, pages 544–553, 1994.

[8] C. Castelluccia, E. Mykletun, and G. Tsudik. Efficient aggregation of encrypted data in wireless sensor networks. In *Proccedings of the 3rd Annual International Conference on Mobile and Ubiquitous Systems: Networks and Services (MobiQuitous'05)*, July 2005.

[9] D. Chaum. Untraceable electronic mail,return address, and digital pseudonyms. *Communications of the ACM*, 24(2):84, 1981.

[10] J. D. Ferrer. A provably secure additive and multiplicative privacy homomorphism. In *Proceedings of the 5th International Conference on Information Security (ICIS'02)*, pages 471–483, 2002.

[11] P. Fouque, G. Poupard, and J. Stern. Sharing decryption in the context of voting or lotteries. In *Financial Crypto'00*, pages 90–104, 2000.

[12] M. J. Freedman and R. Morris. Tarzan: A peer-to-peer anonymizing network layer. In *Proceedings of the 9th ACM Conference on Computer and Communications Security (CCS'02)*, pages 193–206, November 2002.

[13] S. Hameed and N. H. Vaidya. Efficient algorithms for scheduling data broadcast. *ACM/Baltzer Journal of Wireless Networks (WINET)*, 5(3):183–193, 1999.

[14] Q. L. Hu, D. L. Lee, and W.-C. Lee. Optimal channel allocation for data dissemination in mobile computing environments. In *Proceedings of the 18th International Conference on Distributed Computing Systems (ICDCS'98)*, pages 480–487, May 1998.

[15] T. Imielinski, S. Viswanathan, and B. R. Badrinath. Data on air - organization and access. *IEEE Transactions on Knowledge and Data Engineering (TKDE)*, 9(3):353–372, May-June 1997.

[16] M. Jakobsson. A practical mix. In *Advances in Cryptology - Eurocrypt'98*, pages 448–461.

[17] P. Paillier. Public-key cryptosystems based on composite degree residue classes. In *Advances in Cryptology - EuroCrypto'99*, pages 223–238, 1999.

[18] D. Pointcheval. Self-scrambling anonymizers. In *Proceedings of the 4th International Conference on Financial Cryptography*, pages 259 – 275, 2000.

[19] K. Sako and J. Kilian. Receipt-free mix-type voting scheme - a practical solution to the implementation of a voting booth. In *Advances in Cryptology - Eurocrypt'95*, pages 393–403, 1995.

[20] E. Savas, A.F. Tenca, and C.K. Koc. A scalable and unified multiplier architecture for finite fields $GF(p)$ and $GF(2^m)$. In *Proceedings of the Second International Workshop on Cryptographic Hardware and Embedded Systems*, pages 277 – 292, 2000.

[21] Adi Shamir. How to share a secret. *Communications of the ACM*, 22(11):612–613, Nov 1979.

[22] D. Wagner. Resilient aggregation in sensor networks. In *Proceedings of the 2nd ACM workshop on Security of ad hoc and sensor networks*, pages 78 – 87, 2004.

[23] Z. Yang, S. Zhong, and R. Wright. Anonymity-preserving data collection. In *Proceeding of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining (KDD'05)*, pages 334 – 343, 2005.