Balancing Performance and Confidentiality in Air Index

Qingzhao Tan[1] Wang-Chien Lee[1] Baihua Zheng[2] Peng Liu[1] Dik Lun Lee[3]

[1]Pennsylvania State University, USA {qxt103, wul2, pxl20}@psu.edu

[2]Singapore Management University, Singapore bhzheng@smu.edu.sg

[3]Hong Kong University of Science and Technology, Hong Kong dlee@cs.ust.hk

ABSTRACT

Studies on the performance issues (i.e., access latency and energy conservation) of wireless data broadcast have appeared in the literature. However, the important security issues have not been well addressed. This paper investigates the tradeoff between performance and security of signature-based air index schemes in wireless data broadcast. From the performance perspective, keeping low false drop probability helps clients retrieve the information from a broadcast channel efficiently. Meanwhile, from the security perspective, achieving high false guess probability prevents the hacker from guessing the information easily. There is a tradeoff between these two aspects. An administrator of the wireless broadcast system may balance this tradeoff by carefully configuring the signatures used in broadcast. This study provides a guidance for parameter settings of the signature schemes in order to meet the performance and security requirements. Experiments are performed to validate the analytical results and to obtain optimal signature configuration corresponding to different application criteria.

Categories and Subject Descriptors

H.3.1 [**Information Storage and Retrieval**]: Content Analysis and Indexing - *Indexing methods*; H.2.7 [**Database Management**]: Database Administration - *Security, integrity, and protection*

General Terms

Security, Management, Design.

Keywords

Security, Indexing Techniques, Wireless Data Broadcast

1. INTRODUCTION

With rapid advent of wireless technology and growing popularity of smart wireless devices, there is a strong demand on pervasive data services, which facilitate wireless devices and information appliances alike to access much needed information from anywhere, anytime. Today, there are many wireless technologies (e.g., Bluetooth, IEEE 802.11, UMTS, Satellite, etc) that could be integrated to construct a seamless, pervasive information access platform. Although their goals and applications are very different, information access via these wireless technologies can be logically captured by a basic model which consists of an access point (i.e., base station or satellite) and a number of wireless channels. A client may access information via two basic approaches:

- **On-demand Access.** Through an established point-to-point wireless channel, a client submits a request to the server. The server locates the appropriate data and returns it to the client.
- **Broadcast.** Data are broadcast on a wireless channel open to the public. A Client tunes into the broadcast channel and filters out the data according to the request.

On-demand access employs a basic client-server model where the server is responsible for processing a query and returning the result directly to the client via a dedicate point-to-point channel. Alternatively, broadcast approach has the server actively pushing data to the clients. The server determines the data and its schedule to broadcast. A client listens to a broadcast channel to retrieve data based on its queries and, thus, is responsible for query processing.

Compared to on-demand access, broadcast approach is scalable and therefore very attractive to applications with a large number of clients. By broadcasting an item once, all the requests for this item from different clients will be satisfied. Broadcast has been used for TV and radio for a long time. Recently, data broadcast service also appeared. For example, the MSN Direct Service (http://direct.msn.com), based on Smart Personal Objects Technology (SPOT) and DirectBand Network, can provide weather, sports and traffic information to its customers via smart watches.

In a wireless data broadcast environment, any client with appropriate equipment can monitor the broadcast channel and log the data items being broadcast. If the broadcast data items are not encrypted, the broadcast data content is open to the public and any person can access them. Key-based encryption is a natural choice for ensuring secure access of data on air (i.e., only the subscribers who own the valid keys can decrypt the received packets to obtain the data items). Therefore, a search for broadcast data items is answered by receiving all the broadcast data items off the air and decrypting them for further processing (i.e., fil-

CIKM'05, October 31–November 5, 2005, Bremen, Germany.

tering out unwanted data items). To help alleviating the high cost of receiving, decrypting and filtering broadcast data, auxiliary information may be provided on the broadcast channel to annotate the broadcast data items. This technique is called *air indexing*. The basic idea is that, based on index information broadcast along with data items (including indexed attribute values, arrival schedule, length of data items, etc.), mobile clients are able to selectively skip unauthorized or unwanted data items by slipping into doze mode and switch back to active mode only when the data of desire arrives. This technique, substantially reducing workload and energy consumption of mobile clients, is particularly important for encrypted data broadcast. To facilitate efficient access of data on air, index information is preferred to be non-encrypted. Nevertheless, non-encrypted index information may allow unauthorized attackers to infer the data content on broadcast and therefore cause *confidentiality loss*. In this paper, we examine both performance and security issues in air indexing techniques.

Existing air indexing techniques can be roughly classified as *tree-based* and *signature-based* indexes [15]. The tree-based indexes, typically based on clustered data organizations, provide a very accurate and complete global view (in the form of index information) of data items being broadcast on air and thus are very energy efficient for clustered data items. Nevertheless, this 'complete' and 'accurate' index information, if not encrypted, causes significant confidentiality loss which is the major security issue we are concerned in this study. On the other hand, signature-based indexes are particularly good for sequentially structured media (such as a broadcast channel) and multi-attribute indexing. By naturally encoding all the indexed attributes in a bit-vector (i.e., a signature), signature-based techniques allow clients to efficiently filter out *unwanted* data items and thus improve the performance. Since signature-based techniques do not provide the most clear index information, unauthorized attackers cannot be sure of the content of data items and thus reduced the confidentiality loss. In this paper, we investigate the tradeoff between performance and confidentiality of signature-based air indexes by analysis and experiments.

The main contributions of our study are four-fold.

- The issue of confidentiality loss in air indexing is, to the best knowledge of the authors, the first time being identified and discussed in the literature.
- The tradeoff between performance and security requirements in signatured-based air indexes are analyzed in terms of false drop and false guess probabilities of the signatures.
- Performance and security of the examined signature schemes are analyzed. Analytical model for the impact of different control parameters is studied to serve as a guidance for configuring signatures to meet the performance and security requirements of wireless data broadcast applications.
- Extensive experiments (based on simulations) are conducted to validate our analysis and to evaluate the examined signature schemes.

The rest of this paper is organized as follows. In Section 2, we present the background and related work to this study. The problem is formulated in Section 3, together with the metrics for performance and security. In Section 4, we analyze the false drop probability as well as the false guess probability and then derive analytical models for performance and security metrics in a secure wireless broadcast system. Furthermore, we present the simulation results to validate our analysis in Section 5. Finally, we conclude this paper in Section 6.

2. PRELIMINARIES

In this section, we first give an overview of the signature techniques and their application in the wireless data broadcast. Then, we briefly review some related work.

2.1 Overview of the Signature Techniques

Signature techniques have been studied extensively in information retrieval [9]. A signature is basically an abstraction of the information in a data item, which contains a set of attributes. By examining only a signature, we can determine (without checking the data content) whether a corresponding data item does not contain any searched attributes. Therefore, the signature techniques are very suitable for quickly filtering out unwanted data items.

There are a number of ways to generate signatures. Given a set of data items to be indexed by multiple attributes, the signature S_i of data item i is typically formed by first hashing each indexed attribute in the data item into a *bit string* and then *superimposing* (i.e., bitwise-OR, denoted as $\vee$) all these bit strings into a signature. Note that the size of a signature equals the size of the bit string. An example of signature generation is depicted in Figure 1, in which an attribute is hashed into a 12-bit string.

To process a query, a *query signature* S_Q corresponding to the query Q is generated similarly. The query processing is proceeded by comparing S_Q and the signatures of examined data items using bitwise-AND (denoted as $\wedge$). The signatures *match* if for every bit set in S_Q, the corresponding bit in the compared data signature S_i is also set. There are two possible outcomes of the comparison:

- $S_Q \wedge S_i \neq S_Q$: data item i does not match query Q.
- $S_Q \wedge S_i = S_Q$: a match has two possible implications:
  - *true match*: the data item is really what the query searches for; and
  - *false drop*: data item in fact does not satisfy the search criteria although the signature comparison indicates a match.

As shown in Figure 1, three queries are issued and their corresponding signatures are produced. Based on the result of $S_Q \wedge S_i$, the examined data item is not qualified for the first query, `Q=Hacker`, and hence can be discarded. However, it shows a match for both queries `Q=Security` (true match) and `Q=Mobile` (false drop). Thus, the data item has to be retrieved for further checking. Signature techniques have

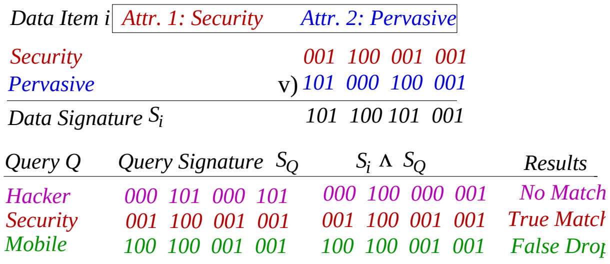


Figure 1: Signature Generation and Comparison.

been employed for air indexing [11]. The idea is to put data items into groups and generate a signature for each group. A server broadcasts a signature before its corresponding group of data items. Since the data items are periodically broadcast in the broadcast channel, a complete broadcast of the data items is called a *broadcast cycle*. Depending on size of data groups, *simple signature* and *integrated signature* air indexing schemes have been proposed. For simplicity, we assume that there is no overlap between the data items in one group for the integrated signature scheme. As shown in Figure 2 and Figure 3 respectively, each group in simple signature scheme contains only one data item while each group in integrated signature scheme contains multiple data items (two in the example). It is assumed that the hashing function H adopted to generate the signatures is known to all the clients. Therefore, when a client issues a query Q, it first generates the query signature S_Q based on H and then starts the retrieval process. Since the signatures and data items are interleaved, the client listens to the signatures to decide whether to retrieve their corresponding data items. When a signature shows a mismatch, the client, by tuning into doze mode, skips the corresponding data items to save energy. It turns back to active mode again when the next signature is broadcast to continue the search process. We assume that each data item has the same size. Thus, the arrival time of the next signature packet is predictable.

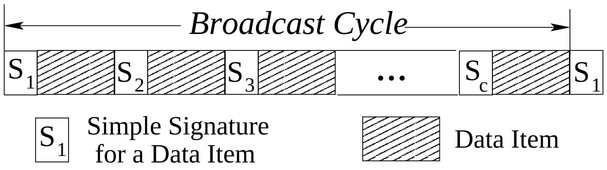


Figure 2: Signature Generation and Comparison.

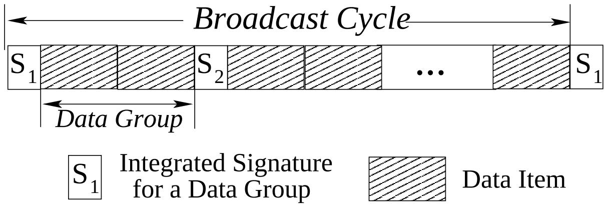


Figure 3: Signature Generation and Comparison.

2.2 Related Work

Air indexing is commonly adopted to conserve battery power in mobile clients. Several tree-based indexing techniques, such as flexible indexing and distributed tree indexing, for broadcast channels were first introduced by Imielinski et al. [6, 7]. Based on the index tree method, work presented in [3, 13] uses unbalanced indexes to improve performance for skewed data access. However, these studies concentrated on one-dimensional indexes for equality-based queries. Lee et al. have addressed general queries with a semantics-based broadcast approach [10]. Tan and Yu have developed a broadcast program that supports range queries [14]. Traditional index techniques, such as hashing [7] and signature file [4], were also applied in air indexing, along with hybrid approach [5]. Besides the design of different indexing structures for different scenarios, index organization algorithms were also studied [8]. However, none of the above techniques addresses any security issue.

There are some related work done in the networking area. For example, [12] focused on secure multicast group key management in the network and [1] proposed broadcast encryption schemes that disseminate a secret to only the privileged clients. However, key management and cryptography are not the focus of this paper and we try to address the security issue from data management aspect. Another related work in networking is *Bloom Filter* [2]. Different from the signature technique, it adopts multiple hashing functions to set the bit strings. As a result, the generation and comparison processes of the bit string become more complicated and time-consuming. It is not as suitable for the wireless broadcast systems as the signature techniques.

3. PROBLEM FORMULATION

Our study aims at revealing important and practical insight on design, deployment and administration of secure wireless data broadcast systems. We assume that data items maintained in the server are indexed by a number of attributes to facilitate efficient search. In order to construct a broadcast program, the *system administrator* chooses a hashing function H for signature generation, decides the number of attributes to be indexed in signatures, and groups data items. Our goal is *to obtain a signature configuration which minimizes confidentiality loss without introducing much performance deterioration.* In this section, we discuss the performance and security metrics, introduce parameters that affect those metrics, and describe our approach in conducting this study.

3.1 Performance and Security Metrics

Access time and *tune-in time* have been widely used as performance metrics in the studies of wireless data broadcast. The former represents the access latency and the latter estimates the energy consumption in mobile clients. Since index information consumes extra bandwidth, balance between the overhead of access time and the gain from tune-in time is extremely important to all the broadcast systems adopting air indexes. A query issued by a client may request multiple items broadcast separately. Thus, a client has to listen to the whole broadcast cycle to avoid a miss of any right answer. Therefore, the access time is only affected by the bandwidth overhead caused by the signature. The tune-in time depends on the filtering ability of signatures. As long as a signature shows a match, whether a true match or a false drop, the data items have to be downloaded and decrypted for further checking. Therefore, the false drop probability should be reduced to minimize the energy consumption on retrieving and decrypting unwanted data items.

On the other hand, the fact that one signature can match different queries provides an uncertainty which prevents attackers (also called *hackers*) from knowing the indexed attribute values of data items. The hackers scan the broadcast channel, download indexes and data items, and try to guess the encrypted content of data items from indexing information. Hence, when a hacker downloads a signature from the broadcast channel, he might start a *dictionary attack.* He uses all the attributes in his dictionary D_H to generate $|D_H|$ signatures and compares each of them with a downloaded signature. Assuming that the attacker's dictionary is comprehensive, he will find a set of matches in D_H. Among those matches, there are *correct guesses* and *false guesses*. A key challenge for the system administrator is to determine confidentiality loss by answering "how much information has been leaked to attackers?" It is important for system administrators to be able to estimate and minimize the information leakage. Thus, *information leaking degree (ILD)* is defined as the security metric in this paper. An important job of the the system administrator is to facilitate highly energy-efficient data access to only the authorized clients (maybe with a cost of some small access latency delay and bandwidth overhead), while minimizing ILD.

The tradeoff between performance and confidentiality, both of which are important to a broadcast system, is studied in this paper. The formal definitions of all the metrics are summarized as follows:

- **Access time (ACC).** The period of time from the moment a query is issued to the moment the client finishes receiving all the qualified data items. Assuming a fixed transmission rate, it is measured in byte.
- **Tune-in time ($TUNE$).** The time duration that a client has to stay in the active mode to answer a query. We use a normalized tune-in time (i.e., the ratio of the tune-in time to the whole broadcast cycle) here.
- **Bandwidth overhead (BO).** The bandwidth consumed on broadcasting signatures. Obviously, it is closely related to the signature length and the number of the groups within one broadcast cycle. It is also represented in the unit of byte in this paper.
- **Information leaking degree (ILD).** The expected number of correct guesses out of all the matched guesses obtained by an attacker. Intuitively, this depends on not only the deployed air index, but also the size and the quality of dictionaries used by attackers.

3.2 Control Parameters

The important parameters that affect the performance and security are listed as follows:

- **Signature length.** The number of bits in one signature (denoted by m).
- **Bit setting.** The fixed number of bits set to 1 in a bit string (denoted by w_b). Signature generation can be controlled by setting w_b between 1 and m.
- **The number of indexed attributes.** The number of attributes in a data item that contribute to the signature (denoted by u).

Due to the fact that a data signature is superimposed from the bit strings of the indexed attributes, the value of u impacts the filtering ability of the signature. In general, a signature superimposed from a small number of attributes provides a more accurate representation of the indexed item. Although this parameter is usually application dependent, it can still be adjusted subject to the system needs. In traditional information retrieval applications, the size of the signature, m, is set to a large value and the number of bits set, w_b, is carefully selected to provide a large space of hashed bit strings and minimize hash collisions. However, for secure wireless data broadcast systems, a large signature consumes too much bandwidth and extends both access latency and tune-in time. Furthermore, a larger signature may result in a higher *ILD*. Consequently, it is extremely important for the administrators to consider all the factors and choose proper control parameters.

3.3 Methodology

This study attempts to correlate performance and security requirements of the signature-based air indexing techniques for wireless broadcast systems and investigate the trade-off between them. At the first step, we conduct an analytical study that brings us a big picture of the impact of factors considered. Based on our observation, *false drop* (for authorized clients) and *false guess* (for attackers) play equally important roles in performance and security of the system. Thus, we show the *false drop probability*, which is the probability that an authorized client thinks a data item is qualified with her query but actually it is not; and *false guess probability*, which is the probability that a dictionary value shows a match with an attacked data item but actually it is not. These probabilities represent a linkage between the performance and security metrics. The combination of parameter settings can govern the false drop probability and false guess probability, which in turn can determine tune-in time, access time, bandwidth overheads and *ILD*.

Following the derived analytical model, we show the relationship between the performance and security metrics and examine various system settings. For example, we fix *bandwidth overhead* and *access time* and analyze the tradeoff between *ILD* and *tune-in time* by varying related parameters. We perform an evaluation by systematically varying signature length and bit strings. As such, we can provide a guidance to the system administrators for configuring signatures to meet the performance and security requirements of wireless broadcast systems.

4. ANALYZING SIGNATURE-BASED AIR INDEX TECHNIQUE

In this section, we first analyze both false drop probability and false guess probability for the signature schemes. The performance and security metrics are then analyzed based on these two probabilities.

4.1 False Drop and False Guess Probabilities

Without loss of generality, we assume a class of uniform hash functions are used to generate signatures. For a given application A, we use D_A to denote the combined domain of indexed attributes. The notation used in our analysis is summarized in Table 1.

4.1.1 False Drop Probability

Semantically, the *false drop probability* P_f refers to the probability that the signature of a data item matches the query signature, yet the data item actually does not satisfy the query. Given a query Q and corresponding signature S_Q, false drop probability can be *experimentally* obtained as follows. Among the total C signatures within one broadcast cycle, let C_t be the true matches, C_f be the false drops, and $C_{m'}$ be the number of signatures that do not match S_Q, i.e., $C = C_t + C_f + C_{m'}$. The false drop probability P_f is defined as the ratio of C_f to $(C - C_t)$.

$$P_f = \frac{C_f}{C - C_t} \qquad (1)$$

Since both the hash collision and superimposition of bit strings in signature generation may cause false drops, we use $P_{f,col}$ and $P_{f,sup}$ to denote the false drop probabilities caused by hash collision and superimposition, respectively. In order to analyze hash collision, *collision factor* is used

A	an application;
D_A	the combined domain of indexed attributes in A;
D_H	the hacker's dictionary;
C	number of signatures in a broadcast cycle;
C_f	number of matched signature due to false drops;
C_t	number of matched signature due to true matches;
$C_{m'}$	number of signatures that do not match;
G	number of values received from the hacker's dictionary;
G_f	number of values received due to false guesses;
G_t	number of values received due to correct guesses;
m	number of bits in a signature;
n	the size of an attribute in the unit of bit;
u	number of attributes indexed in a data item;
s	number of data items in an integrated data group;
w_b	number of 1's in an attribute's signature;
w_f	average number of 1's in a data item's signature;
P_f	false drop probability;
P_g	false guess probability;
P_s	selectivity of a query;

Table 1: Notations.

to denote the average number of different inputs hashed into the same output. Given a hashing function that generates a bit string from an attribute under an application A, $CF_{A,bstr}$ denotes the average number of attributes mapped into the same bit string by the hashing function. Similarly, the collision factor for a signature in application A, $CF_{A,sig}$ denotes the average number of data items hashed into the same signature. Therefore, given u indexed attributes contributing to the signature of a data item, there are at least $(CF_{A,bstr})^{u \cdot s}$ attribute values colliding into a signature. On the other hand, given the average number of 1's in a signature (denoted by w_f) which can be derived in terms of u, m and w_b, there can be as many as $(CF_{A,bstr})^{\mathtt{comb}(w_f,w_b)}$ data items colliding into a signature. Here $\mathtt{comb}(\cdot,\cdot)$ is the binomial function and $\mathtt{comb}(w_f,w_b)$ is the number of bit strings that possibly contribute to the signature. Thus, we have $(CF_{A,bstr})^{\mathtt{comb}(w_f,w_b)} \geq CF_{A,sig} \geq (CF_{A,bstr})^{u \cdot s}$.

For an application A, $\mathtt{comb}(|D_A|,u)$ is the upper-bound of the number of the data items contained in one broadcast cycle, which can be used to estimate C under simple signature scheme. We suppose a given query only has a simple attribute, then the number of matched items can be estimated by $CF_{A,sig} \cdot \mathtt{comb}(|D_A|-1,u-1)$. Among the matched items, only $\mathtt{comb}(|D_A|-1,u-1)$ items are the true matches while the rest are false drops due to the hash collision. As a result, $P_{f,col}$ can be estimated as follows.

$$\begin{aligned} P_{f,col(D_A)} &\approx \frac{\mathtt{comb}(|D_A|-1,u-1)\cdot(CF_{A,sig}-1)}{\mathtt{comb}(|D_A|,u)-\mathtt{comb}(|D_A|-1,u-1)} \\ &= \frac{\mathtt{comb}(|D_A|-1,u-1)\cdot(CF_{A,sig}-1)}{\mathtt{comb}(|D_A|-1,u)} \\ &= \frac{u\cdot(CF_{A,sig}-1)}{|D_A|-u} \qquad (2) \end{aligned}$$

Recall that in the integrated signature scheme, s data items are integrated into a group. A true match happens when at least one of the s items within a group contains the searched key. Thus, the number of true matches C_t can be estimated as follows.

$$\begin{aligned} C_t &= \sum_{i=1}^{s} \mathtt{comb}(I_t,i)\cdot\mathtt{comb}(I_f,s-i) \\ &= \mathtt{comb}(I,s) - \mathtt{comb}(I_t,0)\cdot\mathtt{comb}(I_f,s-0) \\ &= C - \mathtt{comb}(I_f,s) \qquad (3) \end{aligned}$$

where I_t equals to $\mathtt{comb}(|D_A|-1,u-1)$ and I_f equals to $\mathtt{comb}(|D_A|-1,u)$. Hence, $C - C_t = \mathtt{comb}(I_f,s)$. Consequently, $P_{f,col}$ in integrated signature scheme can be derived as below. Note that when $s=1$, Eq. (2) equals Eq. (4).

$$\begin{aligned} P_{f,col(D_A)} &= \frac{C_t\cdot(CF_{A,sig}-1)}{\mathtt{comb}(I_f,s)} \\ &\approx ((\frac{|D_A|}{|D_A|-u})^s-1)(CF_{A,sig}-1) \qquad (4) \end{aligned}$$

For a given signature S_Q of a query, for each bit belonging to S_Q that is set to one, the corresponding bit of the signature S_i of a data item/group is also set to one, a match occurs. However, if the corresponding bits belonging to S_i are actually contributed by different attributes, a false drop is caused by superimposition. The false drop probability caused by superimposition $P_{f,sup}$ has been derived in the literature of traditional information retrieval [11].

$$P_{f,sup} = \frac{\mathtt{comb}(w_f,w_b)}{\mathtt{comb}(m,w_b)} \approx (1-e^{-\frac{w_b u s}{m}})^{w_b} \qquad (5)$$

4.1.2 False Guess Probability

Suppose an attacker receives a signature S_d from the broadcast channel, he produces a set of signatures S_j where $j \in [1,|D_H|]$ from his dictionary and compares each of them with S_d. Let G be the total number of matched guesses, G_t be the correct guesses and G_f be the false guesses (i.e, $G = G_t + G_f$). The *false guess probability* P_g (from the attacker's view) can be defined as the probability that a dictionary value matches a signature of an item but actually is different from the item.

$$P_g = \frac{G_f}{|D_H|-G_t} \qquad (6)$$

Hash collision also has a direct impact on the false guess probability. Let CF_H denote the collision factor associated with D_H (usually much larger than D_A), G_f can be approximated by $(CF_H-1)\cdot G_t$. Therefore, false guess probability caused by hash collision $P_{g,col(D_H)}$ is defined as follows.

$$P_{g,col(D_H)} = \frac{G_t\cdot(CF_H-1)}{|D_H|-G_t} \qquad (7)$$

Finally, the false guess probability caused by superimposition is the same as the false drop probability caused by superimposition.

$$P_{g,sup} = P_{f,sup} \qquad (8)$$

4.1.3 Observations

According to above analysis, we have the following observations. First, P_f is closely related to 1) the signature length m, 2) the bit setting w_b, and 3) the number of indexed attributes superimposed into a data signature u. Therefore, by tuning these parameters, the system can adjust P_f. More specific, P_f is increased if we decrease m, increase u, or do both. The tuning of w_b is more complicated, which will be explored by our simulation in Section 5.

Second, $P_{f,col}$ is related to 1) the size of the application's domain D_A, 2) the number of attributes superimposed into one signature u, and 3) the collision factor $CF_{A,sig}$. Intuitively, the larger the collision factor, the larger the $P_{f,col}$.

D_A and u are also related to the number of matches in signature comparisons. $P_{f,col}$ should be made as small as possible based on the performance requirements.

Third, $P_{g,col}$ cannot be ignored since $|D_H|$ is much larger than $|D_A|$. We can choose a hash function that is sufficiently good for the smaller application domain. The attackers are expected to receive more matches (due to CF_H) and thus feel more difficult to guess the right answer.

4.2 Analytical Model

Based on the derived false drop probability and false guess probability, an analytical model for performance and security of a secure wireless broadcast system is developed.

4.2.1 Performance Metrics

Since we assume that a query can be completed only when all the signatures within one cycle are compared, the access time for a given query is one broadcast cycle. In addition, the client needs to perform an initial probe before a signature is received. Let $PROBE$ denote the initial probe time which on average is a half of the summation of a data group and its signature, and $CYCLE$ denote the time to broadcast the whole cycle, which includes both the signature segment SIG and the data segment $DATA$. Given that each group contains s items, the corresponding access time (ACC) is derived as follows.

$$\begin{aligned} ACC &= PROBE + CYCLE \\ &= \frac{m + n \cdot u \cdot s}{2} + C \cdot (m + n \cdot u \cdot s) \qquad (9) \end{aligned}$$

The tune-in time is the time a client stays active for the initial probe PT, scanning all the signatures within one cycle and retrieving all the matched groups (including both true matches and false drops). In our analysis, $PT = PROBE$. Suppose that the selectivity of a query P_s is known, tune-in time ($TUNE$) can be derived as follows.

$$\begin{aligned} TUNE &= PT + SIG + C_t \cdot n \cdot u \cdot s + C_f \cdot n \cdot u \cdot s \\ &= \frac{m + n \cdot u \cdot s}{2} + C \cdot m + C \cdot n \cdot u \cdot s \cdot P_s \\ &+ \; C \cdot n \cdot u \cdot s \cdot P_f - C \cdot n \cdot u \cdot s \cdot P_s \cdot P_f \qquad (10) \end{aligned}$$

Finally, the bandwidth overhead (BO) is given as folows.

$$BO = SIG = C \cdot m \qquad (11)$$

4.2.2 Security Metric

For a received signature S_i, suppose that a hacker finds G matches with signatures generated from his dictionary and that only G_t attributes are really contained in the attacked data item. Since ILD is defined as the ratio of G_t to G, it is obviously affected by the false guess probability. Thus, it can be derived based on P_g.

$$ILD = \frac{G_t}{G} = \frac{G_t}{G_t + P_g \cdot (|D_H| - G_t)} \qquad (12)$$

This equation reveals some interesting insights. When $P_g = 0$, the information leaking degree is 100%. To reduce information leaking to 50%, P_g needs to be raised up to $G_t/(|D_H| - G_t)$. This finding indicates a dependency between information leaking degree and the false guess probability and points out an observation, i.e., having a reasonable false drop probability, which has similar behavior as the false guess probability, is not such a bad idea for security reasons, even though low false drop probability is preferred from the performance perspective.

5. EXPERIMENTAL EVALUATION

As discussed earlier, a secure wireless broadcast system can meet different performance/security requirements by tuning the control parameters. We conduct several experiments by simulation to demonstrate the flexibility of signature-based air index. All the experiments are implemented in C language in a Unix system. As shown in Table 2, the application domain D_A has $1,000$ attribute values. We assume the application has $10,000$ data items to broadcast. Each data item is characterized by 10 indexed attribute values drawn from D_A. On the other hand, the attacker's dictionary D_H contains $10,000$ attributes which is a superset of D_A (i.e., we made a conservative assumption from the administrator's standpoint). The signature size, m, is set to 64, 128, and 256. By tuning w_b from 1 to m, the system administrator can generate different configurations of signatures.

$\lvert D_A \rvert = 1000$	$\lvert D_H \rvert = 10000$	$u = 10$
$C = 10000$	$m = 64,\ 128,\ 256$	$n = 128$
$s = 4$	$w_b = [1{,}m]$	$P_s = 0.01$

Table 2: Simulation Settings

The experimental results shown here are obtained from the average of 100 queries, each of which is based on an attribute value drawn from D_A. The false drop probability P_f and false guess probability P_g are obtained by both of experiments and analysis for validation. The tune-in time is obtained by Eq. (10). The ILD for each broadcast data item is experimentally obtained by counting the number of correct guesses and the number of total guesses. The ILD shown in the figures are the average of ILD for each attacked data item.

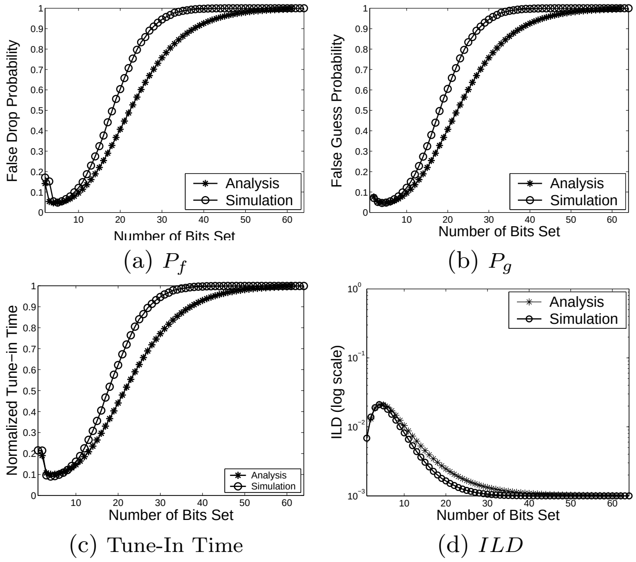
5.1 Validating Analytical Results

We have performed a series of experiments to validate our performance and security analysis. In our experiments, we assume each data group in the integrated signature scheme has 4 data items. By fixing signature size m to $64, 128$ and 256, we vary the number of bit settings, w_b, to observe its impact on false drop probability, false guess probability, tune-in time, and ILD. As for the bandwidth overhead, integrated signatures obviously consume much less bandwidth than the simple signatures scheme because the number of integrated signatures generated is smaller.

Due to the space limitation, here we only show simple signature with $m = 64$ and $m = 256$ in Figure 4 and Figure 5, representing small and large bandwidth overhead allowed, respectively.

The readers should note that the analytical value of false drop rate shown here is approximated by $P_f(D_A) = P_{f,col}(D_A) + P_{f,sup}(D_A)$. The false guess probability is similarly obtained by $P_g(D_H) = P_{g,sup}(D_H) + P_{g,col}(D_H)$. Since the collision factor in our analysis, $(CF_{A,bstr})^{u \cdot s}$, is estimated by its lower bound, the shown analysis values represent lower bound of the real values.

From the figures, we can observe that while the analytical results do not accurately approach the simulation results all the time, their curves show consistent trends. Most importantly, they are close to each other at critical points. This is very useful for quick tuning of control parameters because the analytical results are easier to obtain than simulation results(it took days for us to obtain our simulation results). Also shown in the figures is that the tune-in time shows an opposite trend to ILD as we vary w_b. Thus, it's very important for the administrator to be able to efficiently decide the best signature configurations that meet specific performance and security requirements. For example, if an application requires a higher security, the administrator can raise the false guess probability higher by choosing a larger w_b. For an application to facilitate better energy conservation at clients, a lower false drop probability can be obtained by setting a smaller w_b. The simulation results of the integrated signature scheme provide the same hints and therefore are ignored here due to the space limitation. In order to demonstrate the behavior of the metrics under different settings of m, we have performed extensive simulations. Due to the space limitation, we only show the results for $m = 256$ in Figure 5. All the trends in the figures are similar with those in Figure 4 ($m = 64$). It is worth noting that the minimal false drop probability is smaller while the minimal tune-in time is not.



(a) P_f

(b) P_g

(c) Tune-In Time

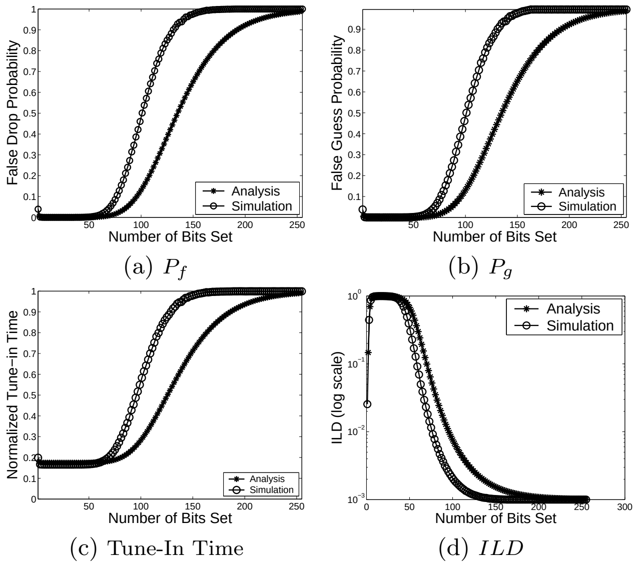
(d) ILD

Figure 4: Performance/Security Metrics of Simple Signature Scheme ($m = 64$)

This is because the filtering ability of signature is improved with a larger m which reduces the false drop probability. However, a signature with a larger m also incurs more bandwidth overhead and hence increases the tune-in time. Based on the simulation results, the energy saving owing to the decreased false drop probability is not sufficient to offset the overhead incurred due to the length of the signatures. That's why the minimal tune-in time for $m = 256$ is even longer than that for $m = 64$. Similarly, a bigger m results in a decreased false guess probability and thereby the maximal ILD is increased. As a result, getting a larger signature does not necessarily guarantee a minimal tune-in time, as well as a maximal ILD.

5.2 Balancing Performance and Security

Ideally, a secure wireless data broadcast system should provide an efficient data access with minimal information leakage. In other words, it requires the false drop probability to be small but false guess probability to be large. Since both P_f and P_g share the same trend, it is impossible to optimize both security and energy at the same time. However, a system administrator can still reach a balance on tune-in time and ILD by properly tuning w_b.



(a) P_f

(b) P_g

(c) Tune-In Time

(d) ILD

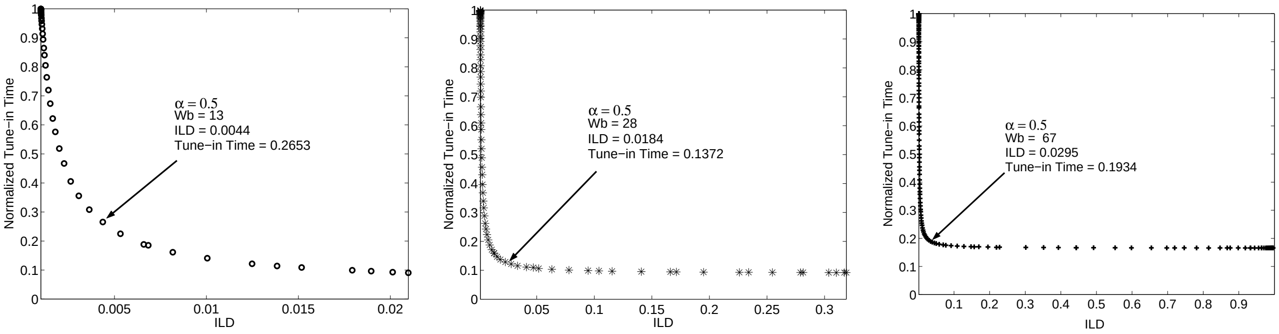
Figure 5: Performance/Security Metrics of Simple Signature Scheme ($m = 256$)

Given a fixed m, without loss of generality, we can obtain a signature configuration that optimizes a normalized cost, which equals to $\alpha\cdot\frac{TUNE}{MAX(TUNE)}+(1-\alpha)\cdot\frac{ILD}{MAX(ILD)}$. Here α, ranging over $[0, 1]$ assigns different weights to performance and security. The larger α is, the more important the tune-in time is considered. In this case, the administrator would like to decrease the tune-in time while increasing ILD (which is the price he has to pay). For example, when security and performance share the same importance to a secure system, $\alpha = 0.5$. As a result, the corresponding w_b that achieves the optimal balance can be derived by either theoretical analysis or experiments.

The relationship between ILD and the tune-in time based on the tuning of w_b is shown in Figure 6. From these figures, we can observe a trade-off between the tune-in time and ILD. While we cannot obtain the minimal ILD and the minimal tune-in time at the same time, an optimal configuration corresponding to different values of α (i.e., representing different emphasis of performance and security) can be obtained. Based on our simulations, optimal w_b settings with respect to different signature configurations (i.e., signature sizes and schemes) and α, along with the corresponding tune-in time and ILD are summarized in Table 3. As such, once an application criteria is set (by assigning a proper α value), the system administrator is able to set a suitable value for w_b to balance both performance and security. Our analysis has provided useful and valuable insights to facilitate this task.

6. CONCLUSION

Air indexing is an important technique to facilitate energy conservation of mobile clients in wireless broadcast system. However, the crucial security issues on air indexing have not been discussed. This paper, to the best of our knowledge, is the first research effort to address both performance and security issues in wireless data broadcast systems.



(a) $m = 64$ (b) $m = 128$ (c) $m = 256$

Figure 6: ILD vs *Tune-in Time* (Simple Signatures, Simulation Result)

α	m	Simple Signature			Integrated Signature		
		w_b	ILD	$Tune$	w_b	ILD	$Tune$
0.25	64	16	0.0026	0.4053	9	0.0039	0.9793
	128	30	0.0092	0.1875	12	0.0051	0.7756
	256	72	0.0163	0.2159	19	0.0101	0.4125
0.50	64	13	0.0044	0.2653	4	0.0054	0.7212
	128	26	0.0184	0.1372	7	0.0090	0.4404
	256	67	0.0295	0.1934	15	0.0173	0.2574
0.75	64	10	0.0082	0.1613	1	0.0085	0.4705
	128	23	0.0333	0.1153	4	0.0153	0.2716
	256	63	0.0500	0.1817	12	0.0281	0.1750

Table 3: Optimal Configurations

In this paper, we argue that signature-based air index is an ideal technique to meet the performance and security requirements of applications because the tradeoff between performance and security metrics can be properly tuned by system administrators. We define a security metrics called information leaking degree to measure confidentiality loss in air indexes and analyze both security and performance metrics in terms of a number of controllable parameters. The analysis provides much insight and guidance about tuning the system. We also conduct a series of simulation-based experiments to validate our analysis and to show that optimal configurations can be easily obtained to meet various performance and security requirements.

This is a new research direction which deserves more effort from the research community. We are developing new air indexing techniques for secure wireless data broadcast and performing more detailed analysis. Both of the performance and security aspects of wireless data broadcast will be further exploited in our future study.

Acknowledgment

Wang-Chien Lee and Qingzhao Tan were supported in part by NSF Grant IIS-0328881.
Peng Liu was partially supported by NSF CCR-0233324 and Department of Energy Early Career PI Award.

7. REFERENCES

[1] S. Berkovit. How to broadcast a secret. In *Proc. of Eurocrypt'91*, pages 536–541, Brighton, UK, 1991.

[2] B. Bloom. Space/time trade-offs in hash coding with allowable errors. *Comm. of ACM*, 13(7), July 1970.

[3] M. Chen, P. S. Yu, and K. Wu. Indexed sequential data broadcasting in wireless mobile computing. In *Proc. of the 17th International Conference on Distributed Computing Systems*, pages 124–131, Baltimore, MD, USA, 1997.

[4] Q. Hu, W. Lee, and D. Lee. Indexing techniques for wireless data broadcast under data clustering and scheduling. In *Proc. of the 8th International Conference on Information and Knowledge Management*, pages 351–358, Kansas City, USA, 1999.

[5] Q. Hu, W. Lee, and D. Lee. A hybrid index technique for power efficient data broadcast. *Distrib. Parallel Databases*, 9(2):151–177, 2001.

[6] T. Imielinski, S. Viswanathan, and B. R. Badrinath. Energy efficient indexing on air. In *Proc. of the International Conference on Management of Data*, pages 25–36, Minneapolis, MI, USA, 1994.

[7] T. Imielinski, S. Viswanathan, and B. R. Badrinath. Power efficient filtering of data on air. In *Proc. of the 4th International Conference on Extending Database Technology*, pages 245–258, Cambridge, UK, 1994.

[8] T. Imielinski, S. Viswanathan, and B. R. Badrinath. Data on air: Organization and access. *IEEE Trans. Knowledge and Data Engineering*, 9(3):353–372, 1997.

[9] D. L. Lee and C. Leng. A partitioned signature file structure for multiattribute and text retrieval. In *Proc. of the 6th International Conference on Data Engineering*, pages 389–397, Los Angeles, USA, 1990.

[10] K. Lee, H. V. Leong, and A. Si. A semantic broadcast scheme for a mobile environment based on dynamic chunking. In *Proc. of the 20th International Conference on Distributed Computing Systems*, pages 522–530, Taipei, Taiwan, 2000.

[11] W. Lee and D. Lee. Using signature and caching techniques for information filtering in wireless and mobile environments. *Journal of Distributed and Parallel Databases*, 4(3):57–67, 1996.

[12] S. Mittra. Iolus: a framework for scalable secure multicasting. In *Proc. of the International Conference of the Special Interest Group on Data Communication*, pages 277–288, Cannes, France, 1997.

[13] N.Shivakumar and S.Venkatasubramanian. Energy efficient indexing for information dissemination in wireless systems. *ACM-Baltzer Journal of NOMAD*, pages 433–446, 1995.

[14] K. Tan and J. X. Yu. Generating broadcast programs that support range queries. *IEEE. Trans. on Knowledge and Data Eng.*, 10(4):668–672, 1998.

[15] T. W. Yan and H. García-Molina. Index structures for selective dissemination of information under the boolean model. *ACM Trans. Database Syst.*, 19(2):332–364, 1994.