

Enabling Information Confidentiality in Publish/Subscribe Overlay Services

Hui Zhang

NEC Laboratories America
Email: huizhang@nec-labs.com

Abhishek Sharma

University of Southern California
Email: abhishek@enl.usc.edu

Haifeng Chen Guofei Jiang

Xiaoqiao Meng Kenji Yoshihira
NEC Laboratories America
Email: {haifeng,gfj,xqmeng,kenji}@nec-labs.com

Abstract—“Alice has a piece of valuable information which she is willing to sell to anyone who is interested in; she is too busy and wants to ask Bob, a professional broker, to sell that information for her; but Alice is in a dilemma where she cannot trust Bob with that information but Bob cannot help her find her customers without knowing that information.”

In this paper, we propose a security mechanism called *information foiling* to address new confidentiality problems arising in pub/sub overlay services [1]. Information foiling extends Rivest’s “Chaffing and Winnowing” [2], and its basic idea is to carefully generate a set of fake messages to hide an authentic message. Information foiling requires no modification inside the broker network so that the routing/filtering capabilities of broker nodes remains intact.

We formally present the information foiling mechanism in the context of publish/subscribe overlay services, and discuss its applicability in other Internet applications. For publish/subscribe applications, we propose a suite of *optimal* schemes for fake message generation in different scenarios. Real-world data are used in our evaluation to demonstrate the effectiveness of the proposed schemes.

Index Terms—security, confidentiality, publish/subscribe, randomized algorithm, data privacy

I. INTRODUCTION

Content-based publish/subscribe (pub/sub) [3][4] enables a dynamic many-to-many communication paradigm for information dissemination: information providers (publishers) and consumers (subscribers) are decoupled through a broker (overlay) network which stores the registered interests (subscriptions) of subscribers and does content-based routing on the messages (events) from publishers to subscribers based on the latter’s interests.

For the benefits such as scalability and performance, a third-party broker overlay network is usually designed for a wide-area pub/sub service with a large and dynamic population of publishers and subscribers. However, data security and privacy concerns over the pub/sub information (e.g., commercial real-time stock price information, current physical location of a customer) may cause publishers/subscribers to place no trust in those third-party broker nodes; a publisher and a subscriber may not know each other in advance and thus have no default trust between each other. In such an environment, a set of new confidentiality problems arise in pub/sub overlay services, as identified in [1]: *Information confidentiality*: can the broker network perform content-based routing without the publishers trusting the broker network with the event

content? *Subscription confidentiality*: can subscribers obtain dynamic data without revealing their subscription functions (content) to the publishers or broker network? *Publication confidentiality*: can publishers control which subscribers may receive particular events?

We propose a new security mechanism called *information foiling* to enable data confidentiality in pub/sub overlay services. The basic idea in information foiling is as originated from Rivest’s “Chaffing and Winnowing” [2]: for a message to be protected, a set of *fake* messages are carefully generated and sent along with the authentic message (all in plain text); the information contained in all the message as a whole is sufficiently confusing to be useless to an attacker sitting inside the broker network; the attacker still has some chance to catch the authentic message (by guessing which is the authentic one), but the catching probability is a controllable parameter in the mechanism. Notice that information foiling requires no modification inside the broker network so that its routing/filtering capabilities are preserved maximally.

Information foiling extends Rivest’s “chaffing and winnowing” by (1) providing a protocol design for enabling confidentiality without encryption in a new application, pub/sub overlay services; (2) formalizing the fake message generation problem, and defining the metrics on measuring the performance of potential solutions; (3) presenting the preliminary results on the design and evaluation of specific fake message generation schemes in pub/sub services.

The core component of information foiling is fake message generation (FMG), a challenging problem in general. We define three metrics to measure the performance of a FMG scheme: *indistinguishability*, *truth deviation*, and communication overhead. We propose a suite of *optimal* FMG schemes based on various assumptions about the knowledge available to the attacker and the information foiler on the routed message. In the context of stock quotes dissemination, real-world stock price data are used in our simulation study to demonstrate the effectiveness of the proposed schemes.

The remainder of the paper is organized as follows. Section II gives the problem definition, and Section III presents the information foiling mechanism. In Section IV, we propose three FMG schemes for publish/subscribe overlay services. The experiment results are shown in Section V. Section VI describes the related work, and Section VII concludes the paper with future work.

II. PROBLEM DEFINITION

A. Pub/Sub Confidentiality

We formulate pub/sub confidentiality as a communication problem. There are three entities: a publisher, a subscriber, and a broker. Both the publisher and the subscriber have private information sources (R_p and R_s). Based on the inputs, the publisher generates a potentially unlimited sequence number of event messages E , and the subscriber has a potentially unlimited number of subscriptions S . At the broker, a subset of the subscriber's subscriptions are stored, which is called the active subscription set. The publisher sends the sequence of events to the broker independently of the subscriber, and the broker sends a sequence of notifications N to the subscriber based on both E and S . Upon an event e , the broker must be able to determine if each subscription s in the active subscription set matches the event based on a function $f(e, s)$, but without learning the information contained in e and s . Contained in the events are essentially data tuples and those in subscriptions are usually selection criteria, such as number comparisons or regular expressions, and the related parameters. The problem definition generalizes naturally to the multiparty case where the number of publishers and subscribers is arbitrarily large.

B. Threat Model

Our threat model is the same as in [5]. A pub/sub broker is assumed to be computationally bounded and exhibits a semi-honest behavior. That is, the broker follows the protocol as prescribed but may record messages to subsequently deduce information not obtainable solely from protocol output (i.e. the outcome of the matching). In the rest of the paper, we use the words "broker" and "attacker" interchangeably when referring to entities in the broker network that try to obtain the authentic information contained in pub/sub messages.

We assume that publishers are honest and publish only valid events. We also assume that subscribers are honest, and a subscriber does not reveal events he/she receives to other subscribers. Otherwise, this would be equivalent to solving the digital copyright problem.

Other security concerns, such as authorization, authentication, integrity, and reliability against DoS attacks, are not the focus of this paper, and existing cryptographical techniques can address those issues as presented in [6].

III. INFORMATION FOILING

A. Information Foiling Mechanism

We propose a mechanism called *information foiling* which achieves data confidentiality by fooling the attacker with fake information. Information foiling introduces a new component, *foiler*, on both publisher and subscriber sides. The functionality of a foiler is to generate a set of k fake messages (called foiling messages in the rest of the paper) $\hat{m}^k = \{\hat{m}_1, \hat{m}_2, \dots, \hat{m}_k\}$ with an input message m , and we call the message set $\{m, \hat{m}^k\}$ a composite set. The number k is a parameter decided by the confidentiality requirement, and

can be personalized for each publisher or subscriber. Both an authentic message and its foiling messages are in plain text when arriving the broker, and they all are valid input for the function $f(e, s)$.

The information foiling mechanism works as following:

- 1) Subscriber: for each active subscription, generates k_s foiling subscriptions, and send them in a random order to the broker which store them all as active subscriptions.
- 2) Publisher: for each event, generates k_p foiling events, and send them in a random order to the broker.
- 3) Broker: upon each arriving event e , decides the subset of the active subscription set and send one notification for each matched subscription.
- 4) (optional) Subscriber: upon a notification associated with one authentic subscription, sends a confirmation request to the publisher.
- 5) (optional) Publisher: upon a confirmation request, sends a reply to the subscriber upon the authenticity of the related event.

Step 4 is optional as we can optimize the confirmation overhead with a shared secret Q between the publisher and subscriber. For example, the publisher can use a hash chain to decide the random position of an authentic message in the composite message set and the chain length is 100 authentic events; if a subscriber sends a confirmation request for the 50th authentic event in a round, the publisher can include the seed of that hash chain in its reply, and the subscriber does not need to ask a confirmation for any notification due to the next 50 authentic events.

Step 5 is optional as besides the same reason for Step 4, there can be access control policy applied by the publisher to decide if a reply is necessary at all.

Last, we note that information foiling requires no modification inside the broker network so that the routing/filtering capabilities of broker nodes remains intact.

B. Performance Metrics

Intuitively, information foiling should satisfy two basic requirements. The first requirement can be interpreted as robustness: most (if not all) of the foiling messages should survive any data filtering technique used by the attacker so that the attacker is not able to identify the correct message. The second requirement is referred to as effectiveness: the foiling messages can serve as a deterrent against attacks if the cost of incorrect guess is significant for the attacker.

Assume the attacker has a function $F : \{e, \hat{E}_e\} \rightarrow G$, that takes the composite message set $\{e, \hat{E}_e\}$ as input and outputs a message set $G \subseteq \{e, \hat{E}_e\}$ consisting of messages that the attacker perceives as useful. Then, we can measure the information foiling performance based on the following metrics:

- *indistinguishability*: defined as $\frac{I(e, G)}{|G|}$, where $I(e, G) = 1$ if $e \in G$, 0 otherwise. We assume $|G| \neq 0$, i.e., the attacker will pick at least one message. This metric captures the robustness of the information foiling mechanism. A value of 0 for this metric implies that the

attacker failed to identify the correct message whereas a value of 1 implies that the attacker was able to do so. However, if the *indistinguishability* is $1/r$ ($r > 1$), then the attacker is not able to distinguish the correct message from $(r - 1)$ foiling messages. The best case is when *indistinguishability* equals to $1/(k + 1)$ (where k is the number of fake messages). *Indistinguishability* can also be interpreted as the probability of identification of the true message by the attacker.

- *truth deviation*: defined as $\sum_{g \in G} \frac{D(e,g)}{|G|}$ where $D(e, g)$ is the difference between the values of messages e and g . This metric captures the effectiveness of the information foiling mechanism.

A third metric, *communication overhead*, is also important in measuring information foiling performance. In terms of content matching, a cryptography-based solution essentially replaces the attribute values in an event with wild-card characters generated using the encryption scheme. Our foiling method associates K additional values with the attributes in an event in addition to the original value. The cryptography-based solution will incur a much higher communication overhead as any event (publication) containing attribute A must be delivered to all the subscribers interested in content containing attribute A , regardless of the value for the attribute A specified by the subscriber.

IV. FAKE MESSAGE GENERATION IN CONTENT-BASED PUB/SUB

A. A Simple Probabilistic Model

In content-based pub/sub model, information used in content-based routing is attribute based. Consider an event message m with L attributes. Let the value V_i for attribute A_i in m be a random variable taking values in V according to a probability mass function (pmf) p_{V_i} . Let $V_m = (V_1, V_2, \dots, V_L)$, i.e. V_m is a vector of random variables associated with message m taking values in V^L . If the attribute value random variables are independent of each other, then the pmf for V_m is

$$p_{V_m} = \prod_{i=1}^L p_{V_i}$$

Each of the K foiling messages generated by the information foiling scheme for m can be thought of as a random variable taking values in V^L . In the scenario where the attacker does not know the pmf p_{V_m} , the aim of the information foiling scheme can be to maximize the entropy $H(M)$ of the composite message set $M = \{m, \hat{m}^K\}$ (defined in Section III-A). We discuss other fake message generation schemes for different scenarios next.

B. Fake message generation schemes

In order to achieve good *indistinguishability* results, the fake message generation scheme must adapt to the available information at the attacker. Different scenarios can be defined based on the various assumptions about the knowledge available to

the attacker and to the foiler. In this paper, it is not our intent to provide an analysis of every possible scenario that may be encountered in practice, but we do investigate a few scenarios that provide further insight into the challenging problem of *optimal* fake message generation.

Scenario I: The attribute values in the event messages are generated according to a random model known both to the foiler at the publisher as well as the attacker, but the accurate pmf is not known. In this scenario, fake messages can be generated by corrupting the true message with noise. The attacker may use techniques from signal processing to filter out the noise from the fake messages to help her determine the correct message. We further investigate this scenario in Section V.

Scenario II: The foiler knows the pmf for attribute values V_i but the attacker does not, and the attacker does not even know the random model of how V_i is generated. This scenario was described in detail earlier and to achieve the best *indistinguishability* performance, the aim of fake message generation scheme should be to maximize the entropy of the composite message (true and fake messages taken together). This can be done by generating the fake messages in such a way that the composite message is uniformly distributed over its support set.

Scenario III: Both the foiler and the attacker know the pmf for attribute values V_i . In this scenario, we think a conservative FMG approach is appropriate which takes maximizing *indistinguishability* as the highest priority and then maximizing true deviation. According, we propose the following bucketing FMG algorithm:

- A FIFO (First In, First Out) queue is maintained in the foiler to cache m ($V_i, \hat{V}_i^1, \dots, \hat{V}_i^k$) vectors generated in the past. Initially it is empty;
- If the current value of V_i can be found in some vector at the FIFO queue, the same values of ($V_i, \hat{V}_i^1, \dots, \hat{V}_i^k$) from that vector is used for this run, and the FIFO queue is updated accordingly;
- Otherwise: the foiler divides the range into $(k + 1)$ sub-ranges (buckets) with equal cumulative probability. The foiler picks one value proportionally to its PDF from each bucket except the bucket where the real value falls into, and the resulting ($V_i, \hat{V}_i^1, \dots, \hat{V}_i^k$) vector is used as the composite data set to be delivered and also put into the FIFO queue for caching.

When multiple attribute values are to be protected in a message and they are independent from each other, the above bucketing algorithm is applied to each attribute individually. When multiple attribute values are to be protected in a message and they are dependent from each other, a master attribute is specified by the foiler and the bucketing algorithm is applied to this attribute; the fake values of the rest attributes are then decided based on the correlation with the master attribute.

Intuitively, the fake values generated by the bucketing algorithm look probabilistically the same to any attacker, which leads to the maximal *indistinguishability* of $\frac{1}{k+1}$; the range

bucketing is for keeping the distance of the authentic value to any of the fake values.

V. EVALUATION

We evaluate *information foiling* in pub/sub stock quotes dissemination service where the current stock price is the content (event) to be protected. Stock prices were collected from [7] with one minute frequency. We use the two metrics - *indistinguishability* and *truth deviation*- introduced in Section III-B to assess the efficacy of our scheme.

A. Experimental Setup

Fake Message Generation: For each instant price for some stock, we generate K fake messages containing false prices for that stock. [8] presents the following discrete-time model describing how the stock prices change:

$$\Delta S = \mu S \Delta t + \sigma S \epsilon \sqrt{\Delta t} \quad (1)$$

where ΔS is the change in the stock price, S in a small interval of time, Δt , and the random variable ϵ has the standard normal distribution. μ is the expected rate of return per unit time from the stock and σ captures the stock price volatility. Both μ and σ are assumed constant and like in [8], we set $\mu = 0.12$ and $\sigma = 0.15$. From Eqn. (1), it is evident that for a fixed stock price S , ΔS is normally distributed with mean $\mu S \Delta t$ and standard deviation $\sigma S \sqrt{\Delta t}$.

For the stock price at time t , S_t , we generate the K fake messages as follows:

$$\tilde{S}_t^i = S_t + \eta_i, \quad i = 1 \dots K \quad (2)$$

where \tilde{S}_t^i is the i th fake message for S_t and η_i is white Gaussian noise. Keeping the variance of η_i small creates fake messages whose values are close to the correct stock price and is desired as per the *indistinguishability* criterion. However, the *average deviation* is directly proportional to the variance of η_i and to achieve higher *average deviation*, we need the variance of η_i to be larger than ΔS_{t-1} . We explore this trade-off in more detail later.

Attacker’s Strategy: In our experiments, the attacker eavesdropping on the data through the pub/sub broker network uses the following two strategies to guess the correct message out of $K + 1$ messages for each event (closing stock price).

- *Uniform Sampling:* The attacker picks each of the $K+1$ messages as the correct message with the same probability. Hence, on an average, the attacker guesses the correct message with probability $1/(K + 1)$ and the *indistinguishability* constraint is satisfied.
- *Extended Kalman Filter* [9]: Assuming that the attacker knows Eqn. (1), she can use an extended Kalman filter to filter out the noise and generate estimates, \hat{S}_t^i of the true stock price from the fake message, \tilde{S}_t^i she observes. She then picks the observed message j as the correct one where

$$j = \arg \min |\tilde{S}_t^i - \hat{S}_t^i| \quad i = 1, \dots, K + 1$$

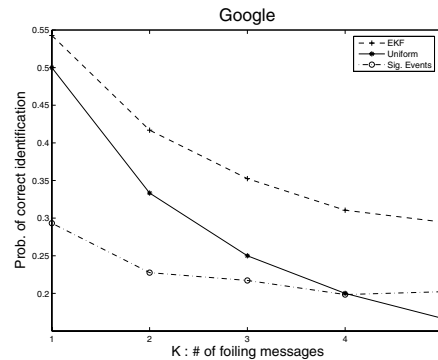


Fig. 1. Indistinguishability results: Google data set.

For the results presented in this section, the variance of Gaussian noise added to the correct stock price to generate the fake messages was of the same order as the variance of ΔS (i.e. equal to the volatility of the change in stock price at a given price S). Figure 1 shows the probability of identifying the correct stock price by the adversary. The curves labeled “Uniform” and “EKF” refer to the uniform sampling and the Kalman filter strategies used by the attacker, respectively. We explain the curve labeled “Sig. Events” later. We also generated the results for many other companies including IBM, Amazon, and Microsoft, and skipped them in this paper due to the similarity.

For $K = 1$ (only one foiling message), the Kalman filter approach leads to slightly higher correct identifications than uniform sampling. However, as K is increased the adversary can guess correctly more often by doing uniform sampling. Hence, information foiling can achieve *message indistinguishability* even when the attacker resorts to sophisticated signal processing techniques to filter out the noise.

As seen in Figure 1, a small number for fake messages ($K = 2$) are enough to severely reduce the adversary’s ability to identify the correct message. It is important to fool the adversary with only a small number of fake messages per correct message because of the communication overhead. Also, increasing K beyond a point does not reduce the probability of correct identification significantly. Hence, depending on the data to be protected, a small number of fake messages would be enough to drastically reduce the probability of correct identification.

Significant Events: We refer to a comparatively large change in the current stock price as a significant event. The curve labeled “Sig. Events” shows the probability of correct guess by the adversary for significant events.

Because of the time (iterations) needed by the Kalman filter estimates to adapt to the sudden change in the stock price, the Kalman filter approach does much worse compared to uniform sampling when the stock price changes by a large amount. As shown in Figure 1, the performance of the Kalman filter approach degrades by more than 20% compared to its overall performance in the case of significant events (for $K \geq 3$).

Truth deviation vs indistinguishability: We now explore the trade-off between having a higher *truth deviation* (to hurt

Factor	Avg. Truth Deviation
1	0.0131
10	0.0414
25	0.0509
50	0.093
100	0.1314

TABLE I

GOOGLE DATASET: TRUTH DEVIATION (AVG. PRICE CHANGE PER MINUTE = 0.167)

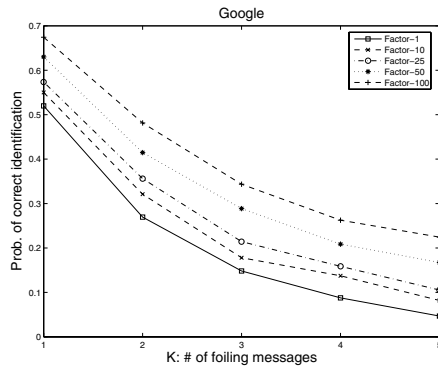


Fig. 2. Google data set: Indistinguishability

the attacker more on an incorrect guess) and the ability of the attacker to identify the correct message. In our evaluation, the function $D(e, g)$ in truth deviation is materialized as $D(e, g) = |e - g|$. Table (I) lists the *avg. truth deviation* (over 500 rounds) achieved for data on Google's stock prices by adding noise with different variances. A value of 10 for "Factor" means the variance of the noise was 10 times the variance of ΔS . As expected, higher variance for the added noise achieves a higher *truth deviation*.

Figure (2) shows the performance of the Kalman filter based approach for the different scenarios listed in Table (I). For higher noise variance, the Kalman filter based approach does quite well and the adversary sees more than double improvement in its ability to identify the correct message and performs better than uniform sampling. But note that even in high variance noise scenario, the efficacy of the Kalman filter based approach quickly decrease with increasing K .

VI. RELATED WORK

Rivest's "chaffing and winnowing" mechanism [2] motivated our work on information foiling. In his original work, information confidentiality is enabled using only authentication primitives. A message to be delivered is splitted into many equal sized parts and a valid MAC is appended to each part as a separate packet. For each authentic packet, called "winnowing", some packets, called "chaffing", are generated with random values for the message part and appended with the same MAC. The two types of packets are mixed and then sent to the receiver. Anyone with the authentication key for the MAC values can distinguish the authentic packets from fake ones. However, no details on fake message generation were presented in the original paper.

Existing approaches address our problems with cryptographic techniques. For example, Wang et al. [1] suggests the

application of *computing with encrypted data* technique [10] for the information confidentiality problem. Raiciu and Rosenblum [5] design a secure pub/sub protocol by extending the secure multiparty computation model PSM [11]. EventGuard [6] relies on a pre-trusted server for the distribution of symmetric keys among publishers and subscribers; all messages communicated between a publisher and its subscribers are encrypted by some symmetric key before entering the broker network and the broker network is utilized only for overlay channel-based multicasting. Those approaches share one or more drawbacks such as pre-distribution of some secrets (cryptographical keys) among the publishers and subscribers, which is not scalable and conflicts with the dynamic many-to-many communication model, or weaken filtering/routing capabilities of the broker nodes by the poor expressivity of subscription functions with the input of encrypted data.

VII. CONCLUSIONS & FUTURE WORK

In this paper, we introduced a new security mechanism called *information foiling* that addresses the confidentiality issues involved in content-based routing via a pub/sub broker network. Our scheme is complementary to the traditional cryptography-based security schemes and offers probabilistic guarantees on information confidentiality.

We also highlight several interesting open problems for future work. The need for a stronger guiding theory to better understand an analytic study on the fundamental trade-off between the fake message number, indistinguishability, and truth deviation is important. Investigating the interaction between a foiler and an attacker in game theory is interesting. The designs of optimal FMG schemes for other interesting and important application scenarios are needed.

REFERENCES

- [1] C. Wang, A. Carzaniga, D. Evans, and A. L. Wolf, "Security issues and requirements for internet-scale publish-subscribe systems," in *Proc. of the HICSS-35, Big Island, Hawaii.*, 2002.
- [2] R. L. Rivest, "Chaffing and winnowing: Confidentiality without encryption," in *RSA Laboratories CryptoBytes* 4(1), 1998.
- [3] P. Eugster, P. Felber, R. Guerraoui, and A. Kermarrec, "The many faces of publish/subscribe," in *ACM Computing Surveys*, 35(2):114–131, 2003.
- [4] F. Fabret, H. A. Jacobsen, F. Llirbat, J. Pereira, K. A. Ross, and D. Shasha, "Filtering algorithms and implementation for very fast publish/subscribe systems," in *SIGMOD '01*. New York, NY, USA: ACM, 2001, pp. 115–126.
- [5] C. Raiciu and D. S. Rosenblum, "A secure protocol for content-based publish/subscribe systems," http://www.cs.ucl.ac.uk/staff/C.Raiciu/files/secure_pubsub.pdf.
- [6] M. Srivatsa and L. Liu, "Securing publish-subscribe overlay services with eventguard," in *the Proceedings 12th ACM Conference on Computer and Communication Security*, 2005.
- [7] <http://finance.yahoo.com>.
- [8] R. R. Gallati, "Securities, random walk on wall street," http://ocw.mit.edu/NR/rdonlyres/Sloan-School-of-Management/15-433InvestmentsSpring2003/040F67B1-EF93-4C0D-B902-4C5241D6E609/0/154332random_walk.pdf.
- [9] D. Simon, *Optimal State Estimation*. New York: John Wiley and Sons, 2006, ch. 13.
- [10] M. Abadi, J. Feigenbaum, and J. Kilian, "On hiding information from an oracle," in *Proc. of the 19th Annual ACM Conference on Theory of Computing*, 1987.
- [11] Y. Ishai and E. Kushilevitz, "Private simultaneous messages protocols with applications," in *Israel Symposium on Theory of Computing Systems*, 1997, pp. 174–184.