# Online Prediction with Privacy

**Jun Sakuma** jun@cs.tsukuba.ac.jp
University of Tsukuba, 1-1-1 Tennodai, Tsukuba, Ibaraki, 305-8577 Japan
Japan Science and Technology Agency, 4-1-8, Honcho, Kawaguchi, Saitama, 332-0012 Japan

**Hiromi Arai** arai.hiromi.ga@u.tsukuba.ac.jp
University of Tsukuba, 1-1-1 Tennodai, Tsukuba, Ibaraki, 305-8577 Japan

## Abstract

In this paper, we consider online prediction from expert advice in a situation where each expert observes its own loss at each time while the loss cannot be disclosed to others for reasons of privacy or confidentiality preservation. Our secure exponential weighting scheme enables exploitation of such private loss values by making use of cryptographic tools. We proved that the regret bound of the secure exponential weighting is the same or almost the same with the well-known exponential weighting scheme in the full information model. In addition, we prove theoretically that the secure exponential weighting is privacy-preserving in the sense of secure function evaluation.

## 1. Introduction

Online prediction with expert advice is a type of learning algorithm in which a learner predicts the classification of sequentially generated examples. At each time point, experts independently make a prediction for the next example. Then, the learner makes a prediction using the prediction provided by each expert. Typically in this situation, no statistical assumptions are made about the process generating the target sequence. The performance of online prediction algorithms is often evaluated by the regret function, which is the difference between the loss of the learner and the loss of the expert suffering the least loss.

If the learner can observe loss from all the experts, the setting is referred to as the full information model. If the learner can observe loss only from one expert chosen by the learner, the setting is called the partial information model. In this study, we consider a situation where each expert does not wish to disclose these predictions and losses for reasons of privacy. The following two problems pose intuitive examples where privacy preservation is required in online prediction.

**Stock price prediction.** Let there be investors who independently predict the expected price of a certain stock once a day. Due to the proprietary nature of the prediction function used by each investor, they do not wish to disclose their own prediction function and expected price. However, they wish to improve the accuracy of the prediction by jointly aggregating the prediction of the other investors. How can investors make a better prediction without violating the confidentiality of the prediction function?

**Prediction of infection outbreak.** Let there be hospitals which attempt to predict outbreaks of pandemic diseases each day by analyzing personal medical records stored in each hospital. Aggregation of predictions is expected to improve the prediction accuracy; however, such aggregation might cause privacy violation of patients. How can hospitals make a better prediction without sharing sensitive medical information?

These problems can be considered as online prediction problems under limited loss observation, in which prediction and loss cannot be shared among experts (hospitals or investors) at all, in order to preserve confidentiality. Thus, these problems can be formulated neither in the full nor partial information model.

In this study, we focus on the fact that, in some cases, each loss can be observed by at least an expert but not disclosed to others due to its secrecy. If we could exploit such loss information associated with an online prediction in such a way that the loss is not disclosed or estimated by others, online prediction under limited

Appearing in *Proceedings of the 27th International Conference on Machine Learning*, Haifa, Israel, 2010.

*Table 1.* The regret bound and information model

| online procedure | info. model | regret bound | comp. time | privacy loss |
|---|---|---|---|---|
| Exponential Weighting (Vovk90) | full | $R_{\mathsf{EW},T} \leq \sqrt{2T \ln N}$ | small | not cared |
| Exp3 (Auer03) | partial | $R_{\mathsf{Exp3},T} \leq 2\sqrt{e-1}\sqrt{NT \ln N}$ | small | partly disclosed |
| SEW with SFE (proposal) | private | $R_{\mathsf{SFE/DR},T} \leq \sqrt{2T \ln N}$ | large | none |
| SEW with crypto (proposal) | private | $R_{\mathsf{SEW/OR},T} \leq \sqrt{2T \ln N} + O(1)$ | medium | none |

observation could be treated as an instance of the full information model.

Such a treatment might appear to be impossible. In the field of security and cryptography, protocols for private distributed computation have been extensively studied for decades. The key idea in this study is in the use of two well studied cryptographic tools, secure function evaluation and a public-key homomorphic cryptosystem. By using these tools, we can design privacy-preserving online prediction protocols that achieve regret minimization with neither the learner nor the expert being able to observe private loss and prediction.

When a specified function is evaluated with taking distributed private information as input, two types of privacy should be considered, privacy-preserving computation and output privacy. The former aims to evaluate the function without violating the privacy of the distributed inputs; the latter attempts to measure the amount of information leaked thorough the publication of the output. Our focus in this study is on privacy-preserving computation of online prediction. Recently, it is well recognized that differential privacy mechanism provides a theoretical treatment of output privacy (Dwork, 2006). Simple combination of our solution with the Laplace mechanism (Dwork, 2006) might provide a solution both for privacy-preserving computation and output privacy in online prediction. This is remained for the future work.

### 1.1. Related Work

If the learner is allowed to observe the loss of all experts (full information model), the regret of exponential weighting strategies is at most $O(\sqrt{T \ln N})$ where $N$ and $T$ are the number of experts and time steps, respectively (Vovk, 1990), (Littlestone & Warmuth, 1994), (Cesa-Bianchi et al., 1997).

A number of studies have been presented concerning online prediction under limited observation of loss. The multi-armed bandit problem or the partial information model (Blum & Mansour, 2007) assume that the learner cannot observe loss values suffered by experts other than the expert chosen by the learner. Auer et. al. have shown that the regret bound of the exponential weighting scheme in this setting is at most $O(\sqrt{NT \ln N})$ (Auer et al., 2003). Comparison of these solutions in terms of the regret bound and privacy loss is summarized in Table 1. These schemes can cope with limited loss observation. However, these are not designed with the intention to protect privacy. Therefore, private information could be partly disclosed or statistically estimated after iterations.

### 1.2. Our Contribution

In this study, we formally introduce a novel information model, termed as the private information model (Section 3). Intuitively, in this model, the learner is not allowed to observe experts' losses and predictions; the experts are not allowed to observe the learner's and other experts' losses and predictions, either.

Our contribution is as follows. Firstly, we introduce a new problem termed as oblivious roulette; roulette playing without seeing the roulette wheel. Then we show that online prediction in the private information model is equivalent to this oblivious roulette problem (Section 3.3). Considering this, we present two types protocols for oblivious roulette (Section 5). The first and second protocols make use of secure function evaluation (SFE) and a homomorphic cryptosystem respectively, to guarantee security (Section 4). These protocols help to convert existing online prediction schemes in the full information model to those in the private information model. Then, as privacy-preserving online prediction, secure exponential weighting (SEW) is presented using these roulette protocols (Section 5.3). We prove that the regret bound of SEW is at most $\sqrt{2T \ln N}$ for the protocol using SFE ($R_{\mathsf{SEW/DR,T}}$) and $\sqrt{2T \ln N} + O(1)$ for the protocol using the homomorphic cryptosystem ($R_{\mathsf{SEW/OR,T}}$). Furthermore, we prove theoretically that SEW is privacy-preserving in the private information model. Comparisons between our solution and existing solutions are summarized in Table 1 again. In order to demonstrate the efficiency of our solution, the results of computational experiments are shown (Section 6).

## 2. Preliminaries

Let there be a *learner* and a set of *experts* $\mathcal{E} = \{1, ..., N\}$. The goal of online prediction is to predict an unknown outcome sequence $y_1, y_2, ...$ of elements of an *outcome space* $\mathcal{Y}$. The outcome sequence is provided by the *environment* arbitrarily. In this paper, we consider the outcome space to be discrete. Then, the outcome space can be regarded as a set of integers $\mathcal{Y} = \{1, 2, ...Y\}$ without loss of generality. In the *full information model*, the learner attempts to predict $y_t$ with observing the prediction of experts (advice) $\boldsymbol{y} = (y_{1,t}, y_{2,t}, ..., y_{N,t})$ at time $t$. The loss resulting from the prediction is evaluated by the *loss function* $\ell : \mathcal{Y} \times \mathcal{Y} \mapsto [0, 1]$.

Let $\boldsymbol{p}_t = (p_{1,t}, ..., p_{N,t})$ be a probability vector called the *strategy*. The learner chooses the $i$-th expert with probability $p_{i,t}$ for its decision and the learner's prediction is set to the chosen expert's prediction. Note that $\boldsymbol{p}_t$ is maintained by the learner to determine its prediction at each time, but is not considered as the learner's output.

Let $j \sim M(i; \boldsymbol{p}_t)$ denote the randomized choice of an expert following strategy $\boldsymbol{p}_t$. Online prediction in the full information model is described as follows:

**Online prediction in the full info. model**

1. the environment arbitrarily chooses the next outcome $y_t$
2. each expert makes a prediction and reveals the prediction $(y_{1,t}, ..., y_{N,t})$ to the learner
3. the learner determines strategy $\boldsymbol{p}_t$ with observed loss values and obtains the prediction $\hat{y}_t = y_{j,t}$ where $j \sim M(i; \boldsymbol{p}_t)$.
4. the environment reveals the outcome $y_t$
5. the learner suffers a loss $\ell(y_t, \hat{y}_t)$, $t \leftarrow t{+}1$. Then return to Step 1.

Let $\ell_{i,t} = \ell(y_{i,t}, y_t)$ be the $i$-th expert's loss at time $t$. Let $\mathsf{H}$ be an online algorithm of the learner which updates his strategy. Then, the learner's loss at time $t$ can be considered as the expected loss of the randomized strategy of $\mathsf{H}$, $\ell_{\mathsf{H},t} = \sum_{i=1}^{n} p_{i,t}\ell_{i,t}$. The total loss of $\mathsf{H}$ is defined by $L_{\mathsf{H},T} = \sum_{t=1}^{T} \ell_{\mathsf{H},t}$.

Let $L_{i,T} = \sum_{t=1}^{T} \ell_{i,t}$ be the total loss of the $i$-th expert during $T$ time steps. Then, the *regret* of the online algorithm $H$ is defined by $R_{\mathsf{H,T}} = L_{\mathsf{H},T} - \min_{i\in\{1,...,N\}} L_{i,T}$.

The learner updates $\boldsymbol{p}_t$ so as to minimize the regret during $T$ prescribed time steps[1]. The exponential weighting scheme employs the following procedure to update the strategy. At $t = 1$, $w_{i,1}$ is initialized as $1/N$ for all $i$. Then, for $t \geq 2$,

$$w_{i,t} = \exp\Big(-\eta \sum_{s=1}^{t-1} \ell(y_{i,s}, y_s)\Big) \tag{1}$$

$$W_t = \sum_{i=1}^{N} w_{i,t}, \quad p_{i,t} = \frac{w_{i,t}}{W_t} \tag{2}$$

where $\eta > 0$ is a user parameter.

Then, for any outcome sequence, when the loss function is convex in the learner's prediction, the regret of the exponential weighting ($\mathsf{EW}$) is bounded (Vovk, 1990; Littlestone & Warmuth, 1994), thus:

$$L_{\mathsf{EW},T} - \min_{i\in\{1,...,N\}} L_{i,T} \leq \sqrt{2T \ln N}.$$

## 3. Privacy in Online Prediction

As discussed in the introductory section, the main objective of this study is to introduce the notion of privacy into online prediction, by means of the *private information model*. In this section, we define the private information model in comparison with the full and the partial information model.

### 3.1. Sequence Observation Models

Before presenting the definition, sequence observation models of sequences are defined. Given a set of sequences, the model defines which parts of sequences are observable by a party.

**Definition 1.** *(public and private) Let there be a party and a sequence $\boldsymbol{x} = (x_1, x_2, ...)$. If the party can observe $(x_1, x_2, ..., x_{t-1})$ but not the other elements at time $t$, then $\boldsymbol{x}$ is public. If the party cannot observe any elements of $\boldsymbol{x}$ at any time, then $\boldsymbol{x}$ is private.*

**Definition 2.** *($\boldsymbol{d}$-private) Let there be a party and $N$ sequences $\boldsymbol{x}_1, ..., \boldsymbol{x}_N$. Let $\boldsymbol{x}_i = (x_{i,1}, x_{i,2}, ...)$. Let the party hold index vector $\boldsymbol{d} = (d_1, ..., d_{t-1}) \in \{1, 2, ..., N\}^{t-1}$ at time $t$. If the party can observe nothing but $(x_{d_1,1}, ..., x_{d_{t-1},t-1})$ from the sequences at time $t$, then $\boldsymbol{x}_1, ..., \boldsymbol{x}_N$ is $\boldsymbol{d}$-private.*

In the full information model, the loss sequences of experts are public to the learner. In the partial information model, when $\boldsymbol{d}$ is a decision sequence of the learner, the sequences of loss values of experts are $\boldsymbol{d}$-private. Table 2 summarizes the two information models. Information models of the experts are not described in this table because an expert is not considered to be a computational party in these models.

[1] For simplicity, we consider the case that the number of time steps $T$ is known by the learner in advance. One can apply "guessing techniques" which guarantee a similar regret bound for unknown $T$ (Auer et al., 2003).

*Table 2.* Information model and sequence observation.

| | The $i$-th expert's seq. | Learner's seq. |
|---|---|---|
| full | public | — |
| partial | $\boldsymbol{d}$-private | — |
| private | private from the other parties | private from the other parties |

### 3.2. Private Information Model

Based on these sequence observation models, our private information model can be stated as follows:

**Definition 3.** *(Private information model) Let there be a learner and $N$ experts. If each party's loss and prediction sequences are private from the other parties, then all parties satisfy the private information model.*

Table 2 summarizes this model again. The privacy of the learner's sequence is considered here. If the learner's information is not private, the problem becomes slightly easier, but still non-trivial because experts' sequences are still private among parties. Considering cases where the learner is one of the experts, as examples described in the introduction, we assume both the learner's and experts' sequences are private.

Finally, the problem of online prediction in the private information model can be stated:

**Statement 1.** *(Online prediction in the private information model) Let $H$ be an online prediction algorithm. Let there be a learner and $N$ experts satisfying the private information model at time $t-1$. After execution of $H$ at time $t$, the learner learns a prediction from $H$ and information inferred from the prediction, but nothing else; all parties are still in the private information model.*

### 3.3. Our Approach

In this subsection, we describe the outline of our solution. In principle, our solution is designed as a secure function evaluation of the exponential weighting scheme. Recall that the sequence of the $j$-th expert is private from the $i$-th expert ($i \neq j$). Nevertheless, the $i$-th expert can independently evaluate Eq. 1 because $w_{i,t}$ is updated only with the loss of the $i$-th expert. On the other hand, the $i$-th expert cannot update Eq. 2 because the $i$-th expert needs to have $w_{i,t}$ for all $i$ for the update of $p_{i,t}$. In this setting, the learner needs to learn $\hat{y} = y_j$ s.t. $j \sim M(i; \boldsymbol{p}_t)$ without knowing anything except $\hat{y}_t$.

This problem can be compared to an imaginary roulette game, called *oblivious roulette.* Let there be $N$ dealers and a player. Then the game of oblivious roulette is stated as follows:

**Statement 2.** *(Oblivious roulette) Let there be $N$ dealers and a player. The $i$-th dealer arbitrary chooses $m_i \in [0,1]$ and $y_i \in \mathcal{Y}$ as input for $i = 1, ..., N$. Let $p_i = \frac{m_i}{\sum_{i=1}^{N} m_i}$ and $\boldsymbol{p} = (p_1, ..., p_N)$. After the protocol execution, the player learns $\hat{y} = y_j$ where $j \sim M(i; \boldsymbol{p})$ and information inferred from $\hat{y}$, but nothing else. The dealers learn nothing, either.*

Note that the player is not allowed to know from which dealer $\hat{y}$ came in this statement. As mentioned above, we can see that the exponential weighting in the private information model is equivalent to the game of oblivious roulette by replacing the dealers and player by the experts and learner, respectively. In the next section, we introduce two cryptographic tools for solving this game of oblivious roulette securely. Then, two types of roulette protocols are introduced.

## 4. Cryptographic Tools

### 4.1. Secure Function Evaluation

Secure function evaluation (SFE) is a general and well studied cryptographic primitive which allows two or more parties to evaluate a specified function of their inputs without revealing (anything else about) their inputs to each other (Goldreich, 2004; Yao, 1986).

In principle, any private distributed computation, including online prediction, can be securely evaluated by means of SFE. However, although polynomoially bounded, naive implementation of the exponential weighting using SFE can be too inefficient. Therefore, in order to achieve online prediction efficiently in the private information model, we make use of existing SFE solutions for small portions of our computation as a part of a more efficient overall solution.

### 4.2. Homomorphic Public-key Cryptosystem

Given a corresponding pair of private and public keys $(\mathsf{sk}, \mathsf{pk})$ and a message $m$, then $c = \mathrm{Enc}_{\mathsf{pk}}(m; \ell)$ denotes a (random) encryption of $m$, and $m = \mathrm{Dec}_{\mathsf{sk}}(c)$ denotes decryption. The encrypted value $c$ uniformly distributes over $\mathbb{Z}_Q (= \{0, ..., Q-1\})$ if $\ell$ is taken from $\mathbb{Z}_Q$ randomly. An *additive homomorphic cryptosystem* allows addition computations on encrypted values without knowledge of the secret key. Specifically, there is some operation $\cdot$ (not requiring knowledge of $\mathsf{sk}$) such that for any plaintexts $m_1$ and $m_2$,

$$\mathrm{Enc}_{\mathsf{pk}}(m_1 + m_2 \mod N; \ell) = \mathrm{Enc}_{\mathsf{pk}}(m_1; \ell_1) \cdot \mathrm{Enc}_{\mathsf{pk}}(m_2; \ell_2)$$

where $\ell$ is uniformly random provided that at least one of $\ell_1$ and $\ell_2$ is. Based on this property, it also follows that given a constant $k$ and $\mathrm{Enc}_{\mathsf{pk}}(m_1; \ell)$, we can compute multiplications by $k$ via repeated application of $\cdot$, denoted as $\mathrm{Enc}_{\mathsf{pk}}(km \mod N; k\ell) = \mathrm{Enc}_{\mathsf{pk}}(m; \ell)^k$. In what follows, we omit the random number $\ell$ from our encryptions for simplicity.

Our solution makes use of semantically secure[2] additively homomorphic encryption, such as Paillier's cryptosystem (Paillier, 1999).

## 5. Online Prediction in the Private Information Model

As discussed, our problem is viewed as the game of oblivious roulette. Here, we consider the distributed roulette protocol shown in Fig. 1. In the rest of this paper, $a \in_r A$ denotes that element $a \in A$ is chosen from set $A$ uniformly at random. The output of distributed roulette follows Lemma 1.

```
Distributed roulette
• The i-th dealer's input: $m_i \in [0,1]$ and $y_i \in \mathcal{Y}$
• The player's output: $\hat{y}$
1. Initialization: Round $k = 1$.
2. For $i = 1, ..., N$, the i-th dealer independently chooses:
       $a_{i,k} \leftarrow \begin{cases} y_i, & \text{with prob. } m_i, \\ Y+1, & \text{otherwise.} \end{cases}$
   and sends $a_{i,k}$ to the player.
3. The player chooses $j \in_r \{1, ..., N\}$.
   (a) If $a_{j,k} \in \mathcal{Y}$, the player outputs $a_{j,k}$ as $\hat{y}$.
   (b) Else $k \leftarrow k + 1$ and go to Step 2.
```

*Figure 1.* Distributed roulette

**Lemma 1.** *Let $\hat{y}$ be the output of the distributed roulette. Then, $\hat{y} = y_j$ where $j \sim M(i; \boldsymbol{p})$.*

*Proof.* In round 1, the player finds $a_{j,k} = y_j \in \mathcal{Y}$ with probability $\frac{m_j}{N}$. In round $k > 1$, the player finds $a_{j,k} = Y + 1 \notin \mathcal{Y}$ for any choice of $j$ with probability $1 - \frac{\sum_{i=1}^N m_i}{N}$. Thus, the probability that $a_{j,k} = y_j \in \mathcal{Y}$ at the end of the protocol is

$$\sum_{k=0}^{\infty} \frac{m_j}{N} \left(1 - \frac{\sum_{i=1}^N m_i}{N}\right)^k = \frac{m_j}{\sum_{i=1}^N m_i} = p_j. \qquad (3)$$

Thus, $j \sim M(i; \boldsymbol{p})$ holds. Consequently, $a_{j,k} = y_j$ where $j \sim M(i; \boldsymbol{p})$. □

Thus, this distributed roulette protocol allows the player to play the roulette without sharing $m_i$. However, this is not secure in the sense of Statement 2 because the learner might estimate $m_i$ from the sequences of $a_{j,k}$. Additionally, the learner is aware of $j$, from which dealer $\hat{y}$ came at each iteration.

### 5.1. Distributed Roulette using SFE

In this section, we show that the distributed roulette protocol can be made secure efficiently in the sense of Statement 2 by means of SFE. Evaluation of $a_{i,k}$ in step 2 can be performed independently by each dealer, so this step does not reveal any private information to the others. Therefore, we apply SFE only to the message exchange in step 2 and the computation in step 3.

SFE is applied as follows. In the beginning, the $i$-th dealer inputs $a_{i,k}$ and $j_i \in_r \{1, ..., N\}$; the player inputs $j' \in_r \{1, ..., N\}$ to SFE. SFE privately evaluates $j = j' + \sum_{i=1}^N j_i \mod N$. By choosing $j$ in this way, the player and dealers can randomly choose a single dealer without learning which dealer is chosen. Then, SFE privately checks whether $a_{j,k} \in \mathcal{Y}$ holds. If this holds, $a_{j,k}$ is returned as $\hat{y}$. Otherwise, "Play again" is returned. Since SFE does not change the output of any computation, the behavior of the distributed roulette protocol follows Lemma 1. Furthermore, since all messages are exchanged over SFE, the security of this protocol is reduced to that of SFE (Yao, 1986), Thus, the security proof is omitted here.

The protocol is expected to be terminated after $1/\sum_{i=1}^N m_i$ rounds in average. When $\sum_{i=1}^N m_i$ is too small, many rounds might be required for termination. However, note that $p_j$ is unchanged if a common constant is multiplied to $m_i$ for all $i$. Then, $\sum_{i=1}^N m_i$ can be controlled so that $\sum_{i=1}^N m_i$ is not too small. This adjustment is also performed by SFE in the experiments shown in the latter section. Let the protocol be terminated after $K$ rounds. Then, SFE performs modulo addition and equality tests $K$ times for a single roulette play. Since these operations are all elemental, the computation is much more efficient compared to the naive SFE implementation of the roulette.

### 5.2. Oblivious Roulette using Crypto

Next, we present yet another protocol for oblivious roulette by means of a homomorphic public-key cryptosystem. The protocol is shown in Fig. 2.

We prove that the oblivious roulette protocol approximately follows distributed roulette.

[2] If the cryptosystem is semantically secure, any information regarding message $m$ is not learned from encryption $\mathrm{Enc}_{\mathsf{pk}}(m)$ for all $m \in \mathbb{Z}_Q$.

**Lemma 2.** *Let $\hat{y}$ be the output of oblivious roulette. Let $\mu = \frac{\sum_{i=1}^{N} m_i}{N}$ and*

$$M'(i;\boldsymbol{p},N,Q) = (1-\gamma)M(i;\boldsymbol{p}) + \gamma U(i;N). \qquad (4)$$

*where $U(i;N)$ is the uniform distribution over $\{1,...,N\}$, $Q$ is a security parameter of the cryptosystem, and $\gamma = \frac{(1-\mu)\frac{N}{Q}}{1-(1-\mu)(1-\frac{N}{Q})}$. Then, $\hat{y} = y_j$ where $j \sim M'(i;\boldsymbol{p},N,Q)$*

*Proof.* From the homomorphic property of the cryptosystem, eq. 6 in step 3 is reorganized as

$$c_{i,k} \quad = \quad \mathrm{Enc}_{\mathsf{pk}}((a_{i,k} - j_k)r_{i,k} + y_i \mod N).$$

Then, in step 5, the player learns $u_i$ from decryption

$$u_i = \mathrm{Dec}_{\mathsf{sk}}(c_k) = r_{i,k}(a_{i,k} - j_k) + y_j \mod N. \qquad (5)$$

Note that the player cannot learn which dealer prepared $u_i$ from the order of received messages because the first dealer randomly shuffles the order of messages in step 4. $u_i$ can take two types of values dependent on the $i$-th expert's choice of $a_{i,k}$ in step 3.

Case 1 ($a_{i,k} - j_k = 0$): This case happens when the $j$-th dealer's choice was $a_{j,k} = j'_k$ and this corresponds with the player's choice $j_k$. When this case happens, the player always finds $u = y_j \in \mathcal{Y}$ in step 5a since the first term of eq. 5 is 0 regardless of random value $r_{i,k}$.

Case 2 ($a_{i,k} - j_k \neq 0$): The case happens when (1) the $j$-th dealer's choice was $a_{j,k} = Y + 1$, or (2) the $j$-th dealer's choice $j'_k$ does not corresponds with the player's choice $j_k$. In this case, $u_i$ distributes uniformly at random over $\mathbb{Z}_Q$ since $a_{i,k} - j_k$ is non-zero and $r_{i,k}$ is random. Note that even when this case happens, the player finds $u = y_j$ with probability $1/Q$.

In round 1, the player finds $a_{j,k} = y_j \in \mathcal{Y}$ with probability $\frac{m_j}{N} + (1-\mu)\frac{1}{Q}$. In round $k > 1$, the player finds $a_{j,k} = Y + 1 \notin \mathcal{Y}$ for any choice of $j_k$ with probability $(1-\mu)(1-\frac{N}{Q})$. Thus, the probability that the player finds $a_{j,k} = y_j \in \mathcal{Y}$ at the end of the protocol is

$$\begin{aligned} p'_i &= \sum_{k=0}^{\infty} \Big(\frac{m_i}{N} + \Big(1-\mu\Big)\frac{1}{Q}\Big)\Big\{(1-\mu)\Big(1-\frac{N}{Q}\Big)\Big\}^k \\ &= \frac{\frac{m_i}{N} + (1-\mu)\frac{1}{Q}}{1-(1-\mu)(1-\frac{N}{Q})}. \end{aligned}$$

Setting $\gamma$ as in Lemma 2, $p'_i = (1-\gamma)\frac{m_i}{\sum_{i=1}^{N} m_i} + \gamma\frac{1}{N}$ holds. Thus, the output follows the probability distribution of eq. 4 . □

**Oblivious Roulette**

- Public input: number of dealers $N$, security parameter $Q$
- Player's input: key pair($\mathsf{pk}, \mathsf{sk}$),
- $i$-th dealers' input: public key $\mathsf{pk}$, $m_i \in [0,1]$, $y_i \in \mathcal{Y}$
- Player's output: $\hat{y}$
- Dealers' output: none

1. Initialization: Round $k = 1$.
2. The player chooses $j_k \in_r \{1,2,...,N\}$ and sends $\mathrm{Enc}_{\mathsf{pk}}(-j_k)$ to all dealers
3. For $i = 1,...,N$, the $i$-th dealer independently chooses $r_{i,k} \in_r \mathbb{Z}_Q$ and
$$a_{i,k} \leftarrow \begin{cases} j'_k, & \text{with prob. } m_i, \\ Y+1, & \text{otherwise} \end{cases}$$
where $j'_k \in_r \{1,2,...,N\}$. Then computes
$$c_{i,k} \leftarrow \Big(\mathrm{Enc}_{\mathsf{pk}}(a_{i,k}) \cdot \mathrm{Enc}_{\mathsf{pk}}(-j_k)\Big)^{r_{i,k}} \cdot \mathrm{Enc}_{\mathsf{pk}}(y_i) \quad (6)$$
and sends $c_{i,k}$ to the first dealer.
4. The first dealer randomly shuffles $(c_{k,1},...,c_{k,N})$ and sends them to the player
5. The player decrypts $u_i \leftarrow \mathrm{Dec}_{\mathsf{sk}}(c_{i,k})$ for all $i$ and computes $Y' = \{u_1,...,u_N\} \cap Y$.
   (a) If $Y' \neq \emptyset$, return $u \in_r Y'$ as $\hat{y}$.
   (b) Else, $k \leftarrow k+1$ and go to step 2.

*Figure 2.* Oblivious Roulette

**Remark 1.** *Let $M = \sum_{i=1}^{N} m_i$. Then, $\gamma = \frac{N^2 - MN}{N^2 + MQ - MN}$. The key size $Q$ is usually set to quite large, such as $Q = 2^{1024}$ for security reasons. Considering that $M \leq N$ holds and $N^2 \ll Q$ typically holds, we can regard $\gamma \simeq 0$.*

Next, we show the security of the oblivious roulette protocol. We assume the player and dealers behave *semi-honestly*; this assumes parties follow a specified protocol properly, but might also use their records of intermediate computations in order to attempt to learn other parties' private information.

**Lemma 3.** *Assuming the player and dealers behave semi-honestly, the oblivious roulette protocol is secure in the sense of Statement 2.*

The proof is omitted here. Intuitively, dealers do not hold the private key, and all messages observed by the dealers are encrypted; the dealers learn nothing. On the other hand, the player holds the public key. However, all the messages observed by the player are randomized and the order thereof is shuffled by the dealers except that the message forms the final output. Thus,

*Table 3.* Computation time per step (second) and info. model. The results are the average of 100 iterations.

| | model | $N=2$ | $N=4$ | $N=8$ | $N=16$ | $N=32$ |
|---|---|---|---|---|---|---|
| EW | full | $0.562 \times 10^{-8}$ | $1.09 \times 10^{-8}$ | $2.03 \times 10^{-8}$ | $3.78 \times 10^{-8}$ | $7.25 \times 10^{-8}$ |
| Exp3 | partial | $0.534 \times 10^{-8}$ | $1.02 \times 10^{-8}$ | $1.56 \times 10^{-8}$ | $2.75 \times 10^{-8}$ | $5.13 \times 10^{-8}$ |
| SEW/OR | private | 13.8 | 27.9 | 56.4 | 113 | 233 |
| SEW/DR | private | 65 | 164 | 306 | 608 | 1207 |

the player learns nothing except the final output.

### 5.3. Secure Exponential Weighting in the Private Information Model

Finally, our secure exponential weighting in the private information model is shown. The protocol basically follows the exponential weighting in the full information model shown in Section 2. The differences between these are (1) the $i$-th expert updates eq. 1 in step 3 instead of the learner, and (2) the distributed or oblivious roulette protocol is used to obtain prediction $\hat{y}_t = y_{j,t}$ where $j \sim M(i; \boldsymbol{p}_t)$.

**Theorem 1.** *The secure exponential weighting is secure in the sense of Statement 1.*

The message exchange in the secure exponential weighting occurs only in the roulette protocol. Since our roulette protocols are guaranteed to be secure, Theorem 1 obviously holds. If distributed roulette with SFE is used, the computation thereof exactly follows the exponential weighting in the full information model; the regret bound is $\sqrt{2T \ln N}$. If the oblivious roulette protocol is used, the regret bound is slightly changed because the strategy of the learner deviates from $M(i; \boldsymbol{p})$ as shown in Lemma 2.

**Theorem 2.** *If the oblivious roulette protocol is used in the secure exponential weighting, the regret bound is* $R_{\mathsf{SEW/OR},T} \le \frac{1-\gamma}{\eta} \ln N + \frac{\eta}{2} L_{\mathsf{SEW/OR}} + \frac{\gamma}{N} \sum_{i=1}^{N} L_i$. *Assuming* $\gamma \le 1/T$ *and setting* $\eta = \sqrt{2 \ln N / T}$, $R_{\mathsf{SEW/OR},T} \le \sqrt{2T \ln N} + c$ *where* $c = O(1)$.

The sketch of proof is shown in Appendix. Again, $\gamma$ can be quite small with large $Q$. So the regret bound can be almost the same with that of exponential weighting even when oblivious roulette is used.

**Application to Examples.** In examples in Section 1, all experts (investors or hospitals) wish to share prediction results; no specific learner exists. In such a case, the threshold cryptosystem can be used. In this cryptosystem, all parties share a common public key while each party holds a different private key. Decryption cannot be performed by fewer than $t$ parties and can be performed by any group of at least $t$. By means of the threshold cryptosystem, all experts can jointly perform secure exponential weighting and can share prediction results in the private information model.

## 6. Experiments and Discussion

We performed experiments to examine the computational efficiency of our protocol. Programs were implemented in Java. As the cryptosystem, (Paillier, 1999) with 1024-bit keys was used. For secure function evaluation, Fairplay (Malkhi et al., 2004) was used. The environment generate $y_t = \{0, 1\}$ randomly at each time. Each expert is assigned a prediction accuracy which uniformly distributes from 0.5 (random) to 0.75 (the best accuracy). Each expert makes a random prediction at each time following the assigned prediction accuracy. We compared exponential weighting learner in the full information model (EW), Exp3 learner in the partial information model (Exp3), and two secure exponential weighting learners, one uses distributed roulette with SFE (SEW/DR) and the other uses oblivious roulette (SEW/OR).

Fig. 3 shows the regret of each learner. Since SEW/DR behaves exactly the same with EW, both are not separated. The regret of SEW/OR is almost the same with EW, too. It is remarkable that SEW learners in the private information model suffer less regret than Exp3 learner in the partial information model. In the partial information model, unobserved information is never exploited for prediction. In our protocol, the use of cryptographic tools allows the learner to exploit hidden experts' predictions for the improvement of the learner's prediction without observing them.

Instead, the computation time of SEW learners is much larger than that of EW and Exp3 learner due to cryptographic operations included. Table 3 shows the learner's computation time (cpu time) per step of online prediction. The computation time of SEW/OR per one round is at most a few minutes in this setting. If the interval of decision making is not too short, SEW is sufficiently practical. The computation of SEW/OR is about 2-6 times faster compared to that of SEW/DR. This is because SFE costs larger time even when the protocol is designed such that only primitive operations are processed by SFE.

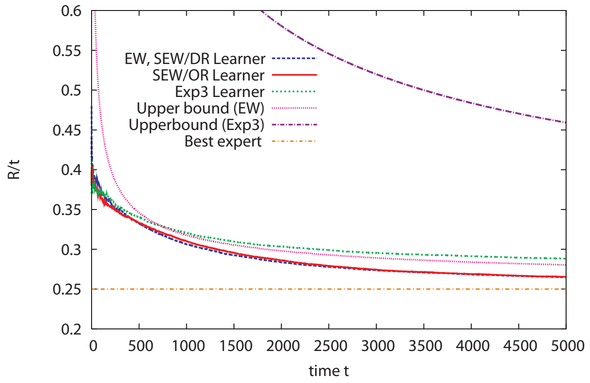

*Figure 3.* Expectation of regret per time (avg. of 100 iterations, $N = 10$, $T = 5000$).

## 7. Conclusion

We presented online prediction algorithms in the private information model, assuming each expert can observe the loss while the loss cannot be disclosed to others due to privacy concern. Our secure exponential weighting for online prediction enables exploitation of private loss values possessed by the experts by making use of cryptographic tools. We proved that the regret bound of our secure exponential weighting is the same or almost the same with the regret of exponential weighting in the full information model. The security of the parties' loss values and predictions are theoretically guaranteed. Our future work is to apply the secure exponential weighting to the repeated game where the payoff matrix of games and players' strategies are desired to be kept private from each other.

## Acknowledgments

We would like to thank to the anonymous reviewers for providing thoughtful comments, also thanks to Jia Yuan Yu for deepen discussions. The work is supported by Japan Science Technology Agency, Sakigake (PRESTO) program .

## References

Auer, P., Cesa-Bianchi, N., Freund, Y., and Schapire, R.E. The nonstochastic multiarmed bandit problem. *SIAM Journal on Computing*, 32(1):48–77, 2003.

Blum, A. and Mansour, Y. Learning, regret minimization, and equilibria. *Algorithmic Game Theory*, pp. 79–102, 2007.

Cesa-Bianchi, N. and Lugosi, G. *Prediction, learning, and games*. Cambridge Univ Pr, 2006.

Cesa-Bianchi, N., Freund, Y., Haussler, D., Helmbold, D.P., Schapire, R.E., and Warmuth, M.K. How to use expert advice. *Journal of the ACM (JACM)*, 44 (3):427–485, 1997.

Dwork, C. Differential privacy. *33rd International Colloquium on Automata, languages and programming*, pp. 1–12, 2006.

Goldreich, O. *Foundations of Cryptography: Volume 2, Basic Applications*. Cambridge University Press, 2004.

Littlestone, N. and Warmuth, M.K. The weighted majority algorithm. *Information and computation*, 108: 212–212, 1994.

Malkhi, D., Nisan, N., Pinkas, B., and Sella, Y. Fairplay: a secure two-party computation system. In *Proc. of the 13th USENIX Security Symposium*, pp. 287–302, 2004.

Paillier, P. Public-key cryptosystems based on composite degree residuosity classes. In *Eurocrypt'99*, pp. 223–238. Springer, 1999.

Vovk, V.G. Aggregating strategies. In *Proceedings of the third annual workshop on Computational learning theory*, 1990.

Yao, A. C.-C. How to generate and exchange secrets. In *Proceedings of the 27th IEEE Symposium on Foundations of Computer Science (FOCS)*, pp. 162–167, 1986.

### Proof of Theorem 2 (Sketch)

The proof methodology used here is similar to (Cesa-Bianchi & Lugosi, 2006), so the detail is omitted here.

The lower bound of $\ln \frac{W_T}{W_0}$ is:

$$\ln \frac{W_T}{W_0} \geq -\frac{\eta}{1-\gamma}\min_i L_{i,T} - \ln N. \qquad (7)$$

Using $\frac{w_{i,t}}{W_t} = \frac{p_{i,t}-\gamma/N}{1-\gamma}$ and $\mathrm{e}^{-x} \leq 1 - x + x^2/2$, we have

$$\frac{W_{t+1}}{W_t} \leq 1 - \frac{\eta-\eta^2/2}{1-\gamma}\sum_{i=1}^{N} p_{i,t}\ell(y_{i,t},y_t) + \frac{\eta\gamma/N}{1-\gamma}\sum_{i=1}^{N}\ell(y_{i,t},y_t).$$

Using $\ln(1+x) \leq x$ and summing over $t$ we get

$$\ln \frac{W_{T+1}}{W_1} \leq \frac{\eta(1-\eta/2)}{1-\gamma}\sum_{t=1}^{T}\sum_{i=1}^{N} p_{i,t}\ell(y_{i,t},y_t) + \frac{\eta\gamma/N}{1-\gamma}\sum_{t=1}^{T}\sum_{i=1}^{N}\ell(y_{i,t},y_t).$$

Combining eq. 7 with the above equation, we have

$$R^T_{\mathsf{SEW/OR}} \leq \frac{1-\gamma}{\eta}\ln N + \frac{\eta}{2}L_{\mathsf{SEW/OR}} + \frac{\gamma}{N}\sum_{i=1}^{N} L_i. \qquad (8)$$

From this and $\eta = \sqrt{2\ln N/T}$, the following holds:

$$R^T_{\mathsf{SEW/OR}} \leq (1-\gamma)\sqrt{\frac{T\ln N}{2}}\ln N + \sqrt{\frac{T\ln N}{2}} + \gamma T$$

Since $\gamma < 1/T$, $R_{\mathsf{SEW/OR},T} = \sqrt{2T\ln N} + O(1)$.