A Privacy-Preserving System for Targeted Coupon Service

JING YAO1,2, YIFENG ZHENG2,3, YU GUO2,3, CHENGJUN CAI2,3, ANXIN ZHOU2,3, CONG WANG2,3, AND XIAOLIN GUI1

1School of Electronic and Information Engineering, Xi’an Jiaotong University, Xi’an 710049, China
2Department of Computer Science, City University of Hong Kong, Hong Kong
3City University of Hong Kong Shenzhen Research Institute, Shenzhen 518057, China

Corresponding authors: Cong Wang (congwang@cityu.edu.hk) and Xiaolin Gui (xlgui@mail.xjtu.edu.cn)

This work was supported in part by the Research Grant Council of Hong Kong under Grant CityU 11212717 and Grant CityU C1008-16G, in part by the National Natural Science Foundation of China under Grant 61572412 and Grant 61472316, in part by the National Key Research and Development Program of China under Grant 2018YFB1800304, and in part by the Key Research and Development Program of Shaanxi Province under Grant 2019GY-005 and Grant 2017ZDXM-GY-011.

ABSTRACT  It is increasingly popular to use behavioral targeting in online coupon service for targeted coupon delivery, wherein customized coupons can be delivered to only eligible users whose behavioral profiles match targeting profiles specified by the vendor. While allowing the vendor to personalize the service and build loyalty, such practice also raises the security challenges of protecting users’ behavioral profiles and the vendor’ targeting profiles. Our research caters for the trend of targeted coupons yet aims to offer privacy assurance. Unlike prior works which are largely constrained by the mere support for secure behavioral targeting, we ambitiously target a new system design that uniquely provides comprehensive functionalities to satisfy practical needs. Specifically, we explore and present the first full-fledged system design for targeted coupon service with three practical functionalities: (i) Private range search on coupon discounts: it enables a user to securely search for encrypted targeted coupons with certain discount range; (ii) Private behavioral targeting: it ensures that only eligible users can decrypt the ciphertexts of targeted coupons obtained from search; (iii) Private blockchain-empowered coupon redemption: it enables coupon redemption to be securely and transparently settled on the blockchain. We conduct extensive experiments and the results show that our secure system design is practically affordable.

INDEX TERMS Privacy, targeted coupon service, secure range search, behavioral targeting, blockchains.

I. INTRODUCTION

Providing coupons to customers is one of the most popular methods for vendors to boost their sales and increase customer loyalty. In recent years, it has become an increasingly popular practice for vendors to delegate the management and delivery of (digital) coupons to a reputable third-party platform (called broker) such as Groupon [1] and RetailMeNot [2]. This can bring many benefits to the vendor such as ease of storage and management, increased brand exposure, and service scalability. In the meantime, the prevalence of online shopping leads to the increasing popularity of behavioral targeting [3], [4], which aims to personalize the online service based on user behaviors, such as geographic location, browsing histories, and purchase behaviors.

Among others, one intriguing and increasingly popular application of behavioral targeting in the coupon service is targeted coupons [4], where the vendor wants to deliver customized coupons only to eligible users whose behavioral profiles match a targeting profile specified by the vendor. For example, a user who buys frequently from the vendor should be treated much better than a user who just makes visits but has not purchased anything. Such targeted coupons provide a highly effective way for the vendor to personalize the coupon service and build loyalty, offering discounts only for those users who deserve the rewards.

Despite the intriguing benefits, the application of targeted coupons requires the collection of users’ personal behavioral information and raises critical privacy concerns. Meanwhile, on the vendor side, it is also required that non-eligible users should learn nothing about the targeting profile and should
not get the coupons. Otherwise, non-eligible users may try to exploit the information they learn so as to obtain targeted coupons [4]–[6], compromising the interests of the vendor. Therefore, both user privacy and vendor protection should be enforced.

In the literature, few works have been done on privacy-aware targeted coupon delivery [4]–[6]. Prior works are largely constrained by the mere support for privacy-preserving behavioral targeting. They do not enable a full-fledged targeted coupon system suited for satisfying practical needs, which are elaborated as follows. Firstly, in practice users may wish to search for targeted coupons containing particular discounts that they are interested in, before the behavioral targeting process is initiated. For example, a user may want to search for targeted coupons which contain discounts between 20% to 30%. Such range query enables a user to effectively narrow down the set of targeted coupons and derive a small candidate set of targeted coupons pertaining to their interests. The challenge here, however, is that such a search functionality needs to be supported in a privacy-preserving manner, due to that the third-party broker hosting the coupons could be untrusted and may attempt to learn the user’s interests.

In addition, the redemption of targeted coupons also requires careful treatment. For coupon redemption, the conventional and common practice is as follows. The user first sends the redemption request to the vendor, who then verifies the validity of the coupon. If the coupon is valid, the vendor then charges the user and provides the corresponding item or service to the user. In addition, the vendor pays a certain service fee to the broker as a reward per successful coupon redemption [7]. Such practice, however, is fully controlled by the vendor, and may lead to critical problems. In particular, the vendor may cheat in his counts of successful coupon redemption so as to underpay the broker. In the extreme case, the vendor may simply claim that no user has ever redeemed a coupon. This is surely unfair to the broker and will undermine the broker’s interests in providing coupon hosting service for the vendor, putting up a barrier for the development of the whole business.

Therefore, there is a pressing need to explore a new full-fledged system design for targeted coupon service that supports comprehensive functionalities for practical deployment. To address such need, in this paper, we propose a new privacy-preserving system architecture for targeted coupon service. Unlike prior works, our system ambitiously takes the first attempt to uniquely support a suite of secure functionalities for targeted coupon service: (i) privacy-preserving range search on discounts of targeted coupons, (ii) privacy-preserving behavioral targeting for targeted coupon delivery, and (iii) privacy-preserving and transparent coupon redemption via blockchains.

The privacy-preserving range search functionality enables the user to securely search for encrypted targeted coupons hosted by the broker which contain discounts that they are interested in. During the search process, the broker learns nothing about what kind of discount information the user is interested in. In order to efficiently support such secure search, our design resorts to an advanced cryptographic primitive called order-revealing encryption (ORE) [8]. Intuitively speaking, ORE is a technique that efficiently supports comparison on ciphertexts, and thus promises the functionality that we aim to accomplish. However, the efficiency of most practical ORE constructions comes at a price, as they cause some information leakages such as order relations. A delicate treatment of using ORE in our system is thus required. For strong security guarantees, we propose a new ORE construction which can hide the order information in the search process.

With a set of encrypted targeted coupons obtained through the privacy-preserving range search functionality, the privacy-preserving behavioral targeting functionality can then be initiated to enable only the eligible users to extract the targeted coupons. Our behavioral targeting design, tailored for our target broker-aided targeted coupon service setting, ensures that only eligible users can recover the coupon encryption keys for encrypted targeted coupons where their behavioral profiles match the targeting profiles, while non-eligible users learn nothing except their non-eligibility. To cater for the practical needs of the vendor, we show how to flexibly support both exact profile matching and approximate profile matching in a privacy-preserving manner.

For redemption of targeted coupons, we explore the use of the emerging blockchain technology to support transparent and enforced redemption processing. The blockchain serves as a tamper-proof and transparent ledger, and also supports enforced program execution via smart contracts [9], [10]. We propose to securely deploy the redemption process on the blockchain so that the redemption process is fully transparent, eliminating the problems as posed by the conventional vendor-centric redemption practice. At a high level, in our system the blockchain receives the redemption request for the user, and then verifies in a privacy-preserving manner whether the targeted coupon claimed by the user is a valid one and has not yet been spent. If the verification passes, the blockchain then automatically charges the user for the redemption of that targeted coupon, and transfers money to the vendor as user payment and to the broker as a reward fee.

Our contributions can be summarized as follows.

- We present a full-fledged system architecture for privacy-preserving targeted coupon service, which is the first that supports comprehensive functionalities of coupon discount range search, behavioral targeting, and redemption, all with privacy assurance.
- We propose a customized security design integrating the advanced cryptographic technique ORE to support privacy-preserving range search on targeted coupons’ discount information. We also propose a security design enabling both exact matching and approximate matching of behavioral profiles in a privacy-preserving manner.
We present an endeavor in exploring secure coupon redemption via the emerging blockchain technology and propose a working design supporting coupon redemption to be automatically settled on the transparent blockchain while ensuring on-chain data privacy.

We conduct an extensive evaluation of our system design, measuring computation, communication, and blockchain performance. The results showcase that our secure system design is practically affordable.

The rest of this paper is organized as follows. Section II presents an overview of our system framework. Section III introduces some useful cryptographic primitives. Section IV gives our security design on privacy-preserving range search on coupon discount information. Section V shows how behavioral targeting is applied in our system in a privacy-preserving manner. Section VI presents the blockchain-based coupon redemption design. Section VII reports experimental results. Section VIII discusses the related work. Section IX concludes the whole paper.

II. OVERVIEW
A. SYSTEM ARCHITECTURE
Fig.1 illustrates the high-level architecture of our system for privacy-preserving targeted coupon service. Users locally maintain behavioral profiles on their devices and want to enjoy targeted coupon service while having privacy assurance. Vendor wants to provide targeted coupons to eligible users whose behavioral profiles match a specific targeting profile. For example, the vendor may leverage behavioral data such as the number of products which a user purchased, the average number of products per purchase, and the number of carts that a user has created. For ease of management of targeted coupons and service scalability, the vendor first resort to the broker, possibly a cloud-based reputable platform such as Groupon, for hosting encrypted targeted coupons. To cater for users’ practical needs, the vendor will supply the broker with ciphertexts of targeted coupons that support secure range search. This is to allow a user to search for targeted coupons with specific discount information, before eligibility range search. This is to allow a user to search for targeted coupons so as to underpay the rewards he should give to the vendor. Meanwhile, the vendor should not learn with which user he performs behavioral targeting as this leaks to the vendor what kind of discount information pertaining to a user’s interests, a user’s behavioral profile, the vendor’s targeting profile, and the targeted coupons. The vendor wants to perform behavioral targeting for coupon delivery, yet he is curious in inferring users’ behavioral profiles and interested discount information. In the coupon redemption process, the vendor may attempt to cheat in the number of redeemed coupons so as to underpay the rewards he should give to the broker. A user may attempt to learn information about the targeting profile of the vendor as well as get targeted coupons that he is not eligible to get. For the blockchain, it is trusted for integrity and availability, but not confidentiality, as in prior blockchain applications (e.g., [10], [11]).

Our security goals are as follows. For security against the broker, we aim to ensure that the broker learns no information about the query and the behavioral profile of the user, the targeted coupons and the targeting profile of the vendor. For security against the user, we aim to ensure that non-eligible users should not learn the targeting profile and the targeted coupons of the vendor, during the process of behavioral targeting based coupon delivery. For security against the vendor, we aim to ensure that the vendor learns no information about the behavioral profile and eligibility status of a user, during the process of behavioral targeting based coupon delivery. Meanwhile, the vendor should not learn with which user he performs behavioral targeting as this leaks to the vendor what kind of discount information that a particular user is interested in. For the coupon redemption process, we aim to ensure that the vendor cannot cheat in coupon redemption rates.

III. CRYPTOGRAPHIC PRIMITIVES
A. ORDER-REVEALING ENCRYPTION
An order-revealing encryption (ORE) scheme is defined over a well-ordered domain $\mathcal{D}$, which consists of three algorithms: (1) the setup algorithm $\text{OREsetup}$; (2) the encryption algorithm $\text{OREenc}$; (3) the comparison algorithm $\text{OREcmp}$.

$\text{OREsetup}(\lambda)$: The setup algorithm is a probabilistic algorithm, which takes as input a security parameter $\lambda$ and outputs
a secret key $sk$. The secret key $sk$ will be used by the data owner to encrypt the message.

ORE$_{enc}(sk, d)$: The encryption algorithm is a probabilistic algorithm, which takes as input the secret key $sk$ and a message $d \in D$ and outputs an ORE ciphertext $ct$. This is the core design of an ORE scheme, which decides the data security and the data comparison efficiency.

ORE$_{cmp}(c_1, c_2)$: The comparison algorithm is a deterministic algorithm, which takes as input two ORE ciphertexts $\{c_1, c_2\}$ and returns a bit $b \in \{0, 1\}$ as the comparison result.

### B. PASSWORD-AUTHENTICATED KEY EXCHANGE

Password-authenticated key exchange (PAKE) is a cryptographic protocol which allows two parties (say party $A$ and party $B$) to safely derive a pair of cryptographic session keys from their passwords. Let $pw_A$ denote the password of party $A$ and $pw_B$ the password of party $B$. Each party will supply his own password as the input to the PAKE protocol. In the end of protocol execution, each party will derive a session key. Let $k_A$ denote the session key of party $A$ and $k_B$ the session key of party $A$. The security of PAKE [12] ensure that if and only if $pw_A$ is equal to $pw_B$, $k_A$ is equal to $k_B$, while the password of each party is kept confidential against each other. Meanwhile, an eavesdropper would not be able to brute force guess a password without further interactions with the parties. Such property allows even weak passwords to be used in the protocol without compromising security.

### IV. PRIVACY-PRESERVING RANGE SEARCH ON TARGETED COUPONS

In this section, we present our security design for privacy-preserving range search on targeted coupons. The goal is to enable the user to search in the ciphertext domain for targeted coupons with discount information that the user is interested in. Note that as the discount information of a targeted coupon (e.g., 10%) generally can be represented by its associated integer value, e.g., 10 for 10%, without involving the symbol “%,” we can treat the problem of discount value comparison as integer comparison. So what we need to consider is how to support secure range query on encrypted integer values.

### A. DESIGN RATIONALE

It is necessary for coupon applications to support range queries. For example, the user wants to obtain the coupons of which the discounts lie in the range of user’s request. To enable secure range queries over encrypted data, one promising technique in the current literature is ORE. However, existing ORE schemes can achieve semantic security only before the ciphertexts get compared. After the comparison, certain information will be leaked. Recent advanced ORE attacks [13]–[16] have been presented, which exploit those leakages to compromise data confidentiality. In addition, the most secure ORE design [17], as far as we know, are not efficient enough. It needs to calculate the pairing result in each block comparison. Therefore, directly using current ORE schemes is not sufficient to serve the needs of secure discount search.

**Algorithm 1 ORE$_{enc}$: ORE Ciphertext Encryption**

<table>
<thead>
<tr>
<th>Input</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>Private key $k$; hash function $h$; discount value $v$; attribute $C$; block values ${v_1^p, ..., v_d^p}$; order $cmp \in {&gt;, &lt;}$; secure PRP $\pi$; secure PRF ${F, G_1, G_2}$</td>
<td>ORE ciphertext $ct_R$</td>
</tr>
</tbody>
</table>

1. Generate a nonce $\gamma$.
2. $k_1 \leftarrow F(k, C||1)$, $k_2 \leftarrow F(k, C||2)$.
3. for $i \in \{1, b\}$ do
   4. for $j \in \{1, 2^d\}$ do
      5. if $v_i \neq v_j^p$ then
         6. $t_j \leftarrow G_1(k_1, v_j^p||\text{cmp})$;
         7. $s_{i,j} \leftarrow G_2(k_2, t_j||v_{i-1})$;
         8. $z_{i,j} \leftarrow \text{"true"} \oplus h(s_{i,j}||\gamma)$;
      9. $ct_{R(i,j)} \leftarrow z_{i,1}, ..., z_{i,2^d-1}$; // the $i$-th block entry
   10. $ct_R \leftarrow \{ct_{R(1,i)}, ..., ct_{R(2^d,b)}, \gamma\}$.

B. OUR ORE CONSTRUCTION

Let $|v|$ be the bit length of an integer value $v$, $b$ denotes the number of blocks, and $d$ represents the bit length of each block. That is, $b \times d = |v|$. Given a $|v|$-bit string, $v_i$ is the $i$-th block value, and $v_{i-1}$ is the prefix. $\{v_1^p, ..., v_d^p\}$ are all possible block values for $d$-bit design. For instance, if $d = 2$ bits, it contains total $2^2$ possible values $\{00, 01, 10, 11\}$. We use “$||$” to denote the concatenation. Our ORE scheme $\Pi = \{ORE_{enc}, ORE_{token}, ORE_{cmp}\}$ is defined as follows:

- **ORE$_{enc}(k, v||C)$**: As shown in Algorithm 1, the encryption performed on the vendor side proceeds as follows. Firstly, the vendor generates the private keys $\{k_1, k_2\}$.
Based on our new ORE construction, we now present the complete protocol for privacy-preserving range search on coupon discounts. At a high level, the protocol proceeds as follows. The vendor first encrypts all his coupons with an AES and all the coupons’ discounts with their attributes by the above $ORE_{enc}$ algorithm. Then he outsources all of them to the broker. When the user queries for coupons, he generates a search token by the $ORE_{token}$ algorithm. Then he sends the token to the broker for searching the candidate coupon collection via the $ORE_{cmp}$ algorithm. Specifically, our secure discount search protocol includes the initialization phase and secure discount search phase. The details of these steps are as follows.

- **Initialization**: In this initialization phase, the vendor first uses its private key to generate encrypted coupons. Then, he performs the $ORE_{enc}$ algorithm to encrypt each coupon’s discount with the corresponding attribute. After the completion of coupon encryption, all these entries will be sent to the broker.

  1. The vendor generates two keys $\{k, k^{\prime}\}$ by using the key generation function $KeyGen(\lambda)$, where $\lambda$ is the security parameter.
  2. The vendor encrypts each targeted coupon $m_i$ by computing $cm_i \leftarrow SE(k^{\prime}_{\ell}, m_i || \xi || \eta || \sigma)$, where $SE$ is a symmetric encryption algorithm, $\xi$ is an identifier uniquely identifying the coupon, $\eta$ is the price associated with the coupon, and $\sigma = sig(H(H(m_i || \xi) || \eta))$ is a signature indicating the validity of the coupon, and $H(\cdot)$ is a hash function.
  3. The vendor calls the ORE encryption function $ORE_{enc}$ to generate the ORE ciphertexts of coupons’ discounts with their attributes, i.e., $ct_R \leftarrow ORE_{enc}(k, v||C)$.
  4. The vendor sends $\{cm_i, ct_R\}$ to the broker and sends the secret key $k$ to the user via a secure channel.

- **Secure Coupon Discount Search**: In the search phase, the user first generates the token via $ORE_{token}$ algorithm. Then, he sends the token to the broker. After receiving the token, the broker uses it to locate the matched coupons by running $ORE_{cmp}$ algorithm. Finally, the broker returns the matched coupons to the user, and notices the vendor and the user start the eligibility test protocol. The detailed steps are as follows:

```
\begin{verbatim}
\begin{algorithm}[h]
\caption{OREtoken: ORE Token Generation}
\label{alg:OREtoken}
\begin{algorithmic}
\Input private key $k$; order condition $cmp \in \{>, <\}$; discount value $v$; coupon attribute $C$; secure PRP $\pi$; secure PRF $\{G1, G2\}$.
\Output ORE query token $ct_L$.
\State $k_1 \leftarrow F(k, C||1)$, $k_2 \leftarrow F(k, C||2)$;
\For{$i \in \{1, b\}$}
\State $t_i \leftarrow G1(k_i, v_i||cmp)$;
\State $ct_{L|i} \leftarrow G2(k_2, t_i||v_{j-1})$;
\State $ct_L \leftarrow \{ct_{L||1}, \ldots, ct_{L||b}\}$;
\EndFor
\end{algorithmic}
\end{algorithm}

\begin{algorithm}[h]
\caption{OREcmp: ORE Ciphertext Comparison}
\label{alg:OREcmp}
\begin{algorithmic}
\Input ORE query token $ct_L$, ORE ciphertext $ct_R$.
\Output true or false.
\State $ct_{L||1}, \ldots, ct_{L||b} \leftarrow ct_L$;
\State $ct_{R||1}, \ldots, ct_{R||b}, \gamma \leftarrow ct_R$;
\For{$i \in \{1, b\}$}
\State $z_{i,1}, \ldots, z_{i,2^d-1} \leftarrow ct_{R|i}$;
\For{$j \in \{1, 2^d - 1\}$}
\State $flg \leftarrow z_{ij} \oplus h(ct_{L||j}||\gamma)$;
\EndFor
\EndFor
\If{$flg == \text{true}$}
\State Output true; // return coupon to client
\Else
\State Output false;
\EndIf
\end{algorithmic}
\end{algorithm}
\end{verbatim}
```
1) The user encrypts his query by computing \( c_{t_l} \leftarrow \text{ORE} \text{token}(k, v || |C|| \text{cmp}) \) and sends the \( c_{t_l} \) to the broker.

2) The broker compares \( c_{t_l} \) with each ORE ciphertext \( c_{t_R} \) in turn by using \( \text{ORE}_{\text{cmp}}(c_{t_l}, c_{t_R}) \). The broker adds the \( cm_i \) to the candidate coupon collection, if the check flag of \( c_{t_R} \) is revealed to be “true.” The broker returns the candidate coupon collection to the user, when all the encrypted discounts have been checked.

We give an example in Fig. 2 on ORE tokens and ciphertexts in our protocol so as to better illustrate how to search for coupons of which the discount values are greater than the token value. As shown in Fig. 2(b), the value 10 of the coupon for coupons of which the discount values are greater than 5% off, the user first generates the ORE token of value “0101” based on the coupons of which the discounts are more than 5% off, the user can access to the broker server. When the system is initialized, the adversary can observe the number of ORE ciphertext \( n \), each ciphertext size \( |ct_R| \), the bit length of each sub-block \( |z| \), the number of blocks \( b \) with \( d \) sub-block in each ciphertext, and random nonces \( \gamma \). The leakages of the initialization phase are defined as \( L_1 \).

\[
L_1(c_{t_R}) = (b, d, (|ct_R|, |z|, \gamma))_n
\]

In the secure discount search phase, the adversary can obtain the access pattern, i.e., the matched ORE ciphertexts and corresponding encrypted blocks. Specifically, when a client sends a token to the broker, the adversary obtains the number of matched encrypted discounts \( m \), the significant different block \( z_{i,j} \) between the search token and the encrypted discount \( c_{t_R} \), the check flag “true,” and the unique nonce \( \gamma \). The leakage \( L_2 \) is defined as:

\[
L_2(v || |C|| \text{cmp}) = (\text{“true”}, \{z_{i,j}, \gamma\}_m)
\]

After the search phase, due to the deterministic property of the query token, the adversary also learns the access pattern. We define the leakage \( L_3 \) to track the access pattern as follow:

\[
L_3(M) = (M_{q \times q})
\]

where \( q \) is the number of tokens sent by the user. \( M_{q \times q} \) maintains the repeated requests. Each element in the \( M_{q \times q} \) is set to be 0 at first. The elements of matrix \( M_{i,j} \) and \( M_{j,i} \) should be set as 1 if token \( t_i \) equals token \( t_j \). Given the leakage definitions, the simulation-based security definition of our scheme is given as follows:

Definition 1: Let \( \Omega = (\text{KeyGen}, \text{Build}, \text{Query}) \) be our scheme for secure discount search. Let \( A \) be a probabilistic polynomial time (PPT) adversary and \( S \) be a PPT simulator. Given leakage \( L_1, L_2 \) and \( L_3 \), we define two probabilistic experiments \( \text{Real}_{\text{A}}(k) \) and \( \text{Ideal}_{\text{A},S}(k) \):

\[
\text{Real}_{\text{A}}(k): \text{The vendor executes KeyGen}(1^k) \text{ for private keys. A selects a set of coupons and asks the vendor to encrypt their discounts via Build (algorithm 1). Then A conducts a polynomial number of q queries and asks the vendor for}
\]

\[
\text{Ideal}_{\text{A},S}(k): \text{The broker initializes the system and simulates all responses of the real system.}
\]

We now formally prove the security strength of our design that can achieve strong protection on coupons’ discounts and user privacy. Besides, we further discuss how our design can defend against recently advanced leakage-abuse attacks on encrypted range-based operations. Since our protocol is based on symmetric-key based searchable encryption, we follow the existing security framework of search over encrypted data [18] to conduct security analysis.

Our new ORE design is built on top of the ORE scheme proposed in [17], which naturally inherits the property of semantic security. That is, an outside attacker who steals and dumps the ORE ciphertext cannot derive any useful information from it. Based on the security strength, we precisely describe the leakages in running our secure discount search protocol and formalize the simulation-based security definition. Specifically, we define the views of an adversary who can access to the broker server. When the system is initialized, the adversary can observe the number of ORE ciphertext \( n \), each ciphertext size \( |ct_R| \), the bit length of each sub-block \( |z| \), the number of blocks \( b \) with \( d \) sub-block in each ciphertext, and random nonces \( \gamma \). The leakages of the initialization phase are defined as \( L_1 \).

\[
L_1(c_{t_R}) = (b, d, (|ct_R|, |z|, \gamma))_n
\]

In the secure discount search phase, the adversary can obtain the access pattern, i.e., the matched ORE ciphertexts and corresponding encrypted blocks. Specifically, when a client sends a token to the broker, the adversary obtains the number of matched encrypted discounts \( m \), the significant different block \( z_{i,j} \) between the search token and the encrypted discount \( c_{t_R} \), the check flag “true,” and the unique nonce \( \gamma \). The leakage \( L_2 \) is defined as:

\[
L_2(v || |C|| \text{cmp}) = (\text{“true”}, \{z_{i,j}, \gamma\}_m)
\]

After the search phase, due to the deterministic property of the query token, the adversary also learns the access pattern. We define the leakage \( L_3 \) to track the access pattern as follow:

\[
L_3(M) = (M_{q \times q})
\]

where \( q \) is the number of tokens sent by the user. \( M_{q \times q} \) maintains the repeated requests. Each element in the \( M_{q \times q} \) is set to be 0 at first. The elements of matrix \( M_{i,j} \) and \( M_{j,i} \) should be set as 1 if token \( t_i \) equals token \( t_j \). Given the leakage definitions, the simulation-based security definition of our scheme is given as follows:

Definition 1: Let \( \Omega = (\text{KeyGen}, \text{Build}, \text{Query}) \) be our scheme for secure discount search. Let \( A \) be a probabilistic polynomial time (PPT) adversary and \( S \) be a PPT simulator. Given leakage \( L_1, L_2 \) and \( L_3 \), we define two probabilistic experiments \( \text{Real}_{\text{A}}(k) \) and \( \text{Ideal}_{\text{A},S}(k) \):
secure tokens and query results via Query (algorithm 3). Finally, \( A \) returns a bit as the output.

\textbf{Ideal}_{A,S}(k): \( A \) selects a set of coupons \( m \). Then \( S \) simulates ORE ciphertexts for \( A \) based on \( L_1 \). \( A \) performs a polynomial number of \( q \) requests. According to the leakages \( L_2 \) and \( L_3 \), \( S \) returns the simulated tokens and the search results. Finally, \( A \) returns a bit as the output.

\( \Omega \) holds the non-adaptively secure with leakages \((L_1, L_2, L_3)\) if for all PPT adversaries \( A \), there exists a simulator \( S \) such that: \( \Pr[\text{Ideal}_{A}(k) = 1] - \Pr[\text{Ideal}_{A,S}(k) = 1] \leq \text{negl}(k) \), where \( \text{negl}(k) \) is a negligible function in \( k \).

\textbf{Theorem 1}: \( \Omega \) is non-adaptively secure with \( (L_1, L_2, L_3) \) if \( F \), \( G_1 \), \( G_2 \) are secure PRFs.

\textbf{Proof}: Given \( L_1 \), the simulator \( S \) generates the simulated discount \( c_{\text{\#}} \), which is indistinguishable from the real one \( c_{\text{\#}} \). Specifically, the sizes of \( c_{\text{\#}} \) is \( |c_{\text{\#}}| \), which is the same as the real one. For each simulated ORE ciphertext, it contains \( b \times d \) sub block entries with equal bit length \( |z| \). But \( S \) generates random strings for each sub block.

According to \( L_2 \), \( S \) simulates the first token and its corresponding search result. The \( S \) randomly selects \( m \) encrypted discount entries. Note that the number of entries \( m \) is the same as that runs in the \textbf{Real} game, and assigns the resulting check flags “true” to the each of the simulated entry which has been selected. Then the mask can be simulated as “true” \( \oplus h(z_{i,j} || y) \), where \( z_{i,j} \) is a random string as the simulated token. It can be derived from \( O_{G_2}(O_{G_1}(v_i | \text{cmp}) | v_{i-1}) \), where \( O_{G_1}, O_{G_2} \) are random oracles. Meanwhile, \( S \) updates \( M_{j,1} = 1 \) in a matrix \( M_{q \times q} \) according to \( L_3 \).

In the \( j - th \) query (for \( j \in \{2, q\} \), if \( L_3 \) indicates that the query has appeared before, \( S \) forms the same tokens as before and selects the same discount entries. Meanwhile, it updates the element \( M_{j,x} \) and \( M_{j,1} \) to be “1.” Otherwise, \( S \) simulates a new token and selects other discount entries to reset the check flag to “true” by following the procedure of the first request via \( L_2 \).

Due to the pseudo-randomness of PRF, the secure hash function and the semantic security of symmetric encryption, \( A \) cannot distinguish the outputs of the real experiment and the simulated one. \( \square \)

\textbf{Discussion}. We further discuss how our design can defend against recent attacks on ORE. According to the leakage classification in [14], we first classify the attacks into two main categories: (1) Ideal leakage based attacks; and (2) Most significant different bit leakage profile (MSDB) leakage based attacks. Then we roughly describe how these attacks work and show that our design can defend against these attacks.

We first discuss the ideal leakage based attacks. The Ideal leakage profile contains minimal information. It only contains the ciphertext distribution before searching and the order information of plaintexts after searching. The existing attacks exploiting Ideal leakage can be classified into two categories according to the auxiliary knowledge: (1) inference attacks; (2) reconstruction attacks. We discuss the possibility of preventing these attacks against our ORE scheme in the following.

All the existing inference attacks [13]–[16] reveal the ciphertexts by mapping each ciphertext to data in the publicly available dataset. The mapping function is mainly based on the order relations of the ciphertexts. Given that our ORE scheme hides the order information by tokenizing the order information and comparing the ciphertexts by text matching, our proposed scheme will not leak the order information in both token generation and the ciphertext comparison. Therefore, it naturally defends against these attacks.

The reconstruction attacks [19], [20] further reduce the leakage requirement of the inference attacks. They only exploit the access pattern (i.e., the search result) or the communication volume (i.e., the number of returned data) to launch the attack. By observing the distribution of the query result, the attacker can infer the relative ordering of ciphertexts. We are aware that adding dummy ciphertext can mitigate this attack effectively. Meanwhile, we are aware that the success rate of these attacks highly relies on the distribution of queries. It makes the attacker difficult to reveal the plaintext, since the domain of our token is too large to collect enough queries to reconstruct the indices (i.e., we tokenize the value, the attribute and the order condition together).

Next, we discuss MSDB leakage based attacks. The MSDB leakage profile contains the most significant different bit between two ciphertexts after searching. All the existing attacks [14], [15] need to exploit MSDB leakage to assign the bits of ciphertexts which has already been revealed. Since we encrypt the plaintext by block and use the secure permutation to hide the location of the first different block, the attacker cannot directly point out the leaky block belongs to which position in the plaintext. Therefore, these two attacks are failed to reveal the bit in MSDB leakage profile.

\section*{V. PRIVACY-PRESERVING BEHAVIORAL TARGETING FOR COUPON DELIVERY}

After a user gets back a set of encrypted targeted coupons through the above privacy-preserving range search process, our system then supports behavioral targeting to enable a user to decrypt targeted coupons that he is eligible for, while preserving the privacy of both the user’s behavioral profile and the vendor’s targeting profile. In the following, we give our security design to support such privacy-preserving behavioral targeting. For simplicity of presentation, we will first focus on the support for privacy-preserving exacting profile matching in our design, and then discuss how to support approximate profile matching.

\subsection*{A. DESIGN RATIONALE}

As the user has the ciphertexts of targeted coupons from the privacy-preserving range search procedure, we need to consider how to securely transfer the coupon encryption keys to eligible users. Our basic idea for supporting privacy-preserving behavioral targeting is to let the user and
the vendor engage in a PAKE-based protocol, in which the user inputs the behavioral profile as his “password” and the vendor inputs the targeting profile as his “password.” From the PAKE-based protocol execution for a targeted coupon, the vendor and the user will produce their own session key respectively. Then, the vendor uses his session key to encrypt the coupon encryption key for that targeted coupon, and sends the key ciphertext to the user. The vendor uses his own session key to decrypt the received ciphertext of the coupon encryption key. Here, the property of PAKE, as introduced above, ensures that the user can produce the same session key as the vendor and so can successfully recover the coupon encryption key, if and only if his behavioral profile exactly matches the vendor’s targeting profile.

To apply the above basic idea for privacy-preserving behavioral targeting, some practical considerations should be further addressed. Firstly, the whole protocol must have only one round, even the user holds multiple encrypted targeted coupons. This allows the exchange of messages in PAKE to be easily facilitated by the broker. Here, the aid of the broker for simply forwarding messages is to prevent the vendor from knowing about with which user he is running the private behavioral targeting protocol. Hiding such information respects the privacy of the user. As the vendor knows the discount information of each targeted coupon, knowing which user is running the PAKE-based protocol enables the vendor to learn the interests of the user.

Secondly, it would be desirable to have a protocol that supports the execution of multiple PAKE instances in parallel. Meanwhile, while the vendor may have a targeting profile per targeted coupon, the user has only one behavioral profile. So, it should be possible for the user to use the same behavioral profile in concurrent PAKE instances. To cater for such practical needs, we delicately employ the SPAKE2 cryptographic primitive proposed in [12] as a base for our privacy-preserving behavioral targeting design. The SPAKE2 supports PAKE in the concurrent setting and supports the use of same input and one-round execution [21].

**B. PROTOCOL OF PRIVACY-PRESERVING BEHAVIORAL TARGETING**

We now present the protocol for privacy-reserving behavioral targeting. Suppose that the user receives a set of encrypted targeted coupons \( \{cm_1, \ldots, cm_l \} \) from the broker at the privacy-preserving range search process. Besides, following existing works [4]–[6], the user’s behavioral profile is represented as a vector \( \mathbf{a} \) where each element in the vector is an integer referring to the value of certain attribute such as the number of products which a user purchased, the average number of products per purchase, and the number of carts that a user has created. Likewise, the targeting profile for a targeted coupon is also represented as a vector \( \mathbf{b} \) with the same dimension, which characterizes the user behavior that the vendor wants to target.

At a high level, the protocol proceeds as follows. Through the aid of the broker, the user runs a SPAKE2-based protocol with the vendor, and produces a set of session keys \( \{s_i\} \). The vendor also produces a set of session keys \( \{s_{vi}\} \), and uses each \( s_{vi} \) to encrypt the coupon encryption key \( k^i \) corresponding to the targeted coupon ciphertext \( cm_i \). The user receives the ciphertext of the coupon encryption key for each targeted coupon \( cm_i \), and uses his corresponding session key to decrypt the ciphertext. If and only if the user’s behavioral profile exactly matches the targeting profile of the targeted coupon \( cm_i \), the user can successfully recover the targeted coupon \( cm_i \). In detail, the whole protocol proceeds as follows.

- **Setup**: The setup phase establishes a set of system-wide public parameters. In particular, the following public parameters are generated: a finite cyclic group \( G \) of order \( p \), where \( p \) is a large prime and its generator is an element \( g \); a public element \( m_u \) associated with the vendor; two hash functions \( h_1 \) and \( h_2 \) considered as two random oracles; an identifier \( ID_v \) for the vendor. Note that we fix the output length of hash function \( h_2 \) to be the same as the length of \( k_i \).

- **Private Behavioral Targeting**: The broker randomly generates an element \( m_v \in G \) and an identifier \( ID_u \) for the user and sends it to both the user and the vendor. Then, it proceeds as follows.

1) The vendor computes \( Y^u_i = g^{y_i}m_v^{h_1(b_i)} \) for the targeting profile \( \mathbf{b}_i \) of each tested targeted coupon \( cm_i \), where \( y_i \in \mathbb{Z}_p \).

2) The user computes \( X^s = g^s m_u^{h_1(a)} \), where \( x \in \mathbb{Z}_p \), and sends \( X^s \) to the vendor through the broker.

3) The vendor computes \( K_{vi} = (X^s m_u h_1(b_i))^{y_i} \) and the session key \( s_{vi} = h_2(ID_u, ID_v, X^s, Y^u_i, K_{vi}) \), for each \( i \in \{1, \ldots, l\} \).

4) The vendor encrypts each coupon encryption key \( k_i^i \) by computing \( c_{ki} = k_i^i \oplus s_{vi} \), and sends \( \{c_{ki}^i, Y^u_i\}_{i=1}^l \) to the user through the broker.

5) The user computes \( K_i = (Y^u_i m_v h_1(a))^x \) and the session keys \( s_i = h_2(ID_u, ID_v, X^s, Y^u_i, K_i) \), for each \( i \in \{1, \ldots, l\} \). Then the user decrypts each ciphertext \( c_{ki}^i \) via \( k_i^i = c_{ki}^i \oplus s_i \) and obtains the correct coupon plaintext \( m_i (s_i = s_{vi}) \) or a dummy plaintext (if \( s_i \neq s_{vi} \)).

**Remarks**: The above protocol is advantageous in that it allows the user to concurrently perform eligibility tests for each (encrypted) targeted coupon. Although our above protocol focuses on behavioral targeting via exact matching of profiles, we note that our design is flexible enough to support approximate matching. That is, we can resort to locality-sensitive hashing (LSH), a technique for similarity search, to compute the LSH values of profiles [22], [23]. Then, approximate profile matching can be transformed into exact matching of LSH values of profiles.

**C. SECURITY ANALYSIS**

The security of our privacy-preserving behavioral targeting protocol directly follows from that of SPAKE2 [12]. Specifically, the security of SPAKE2 ensures that the input (i.e., behavioral profile) of the user and the input (i.e., targeting profile) of the vendor are protected against
each other. So, the vendor learns no information about the user’s behavioral profile through the behavioral targeting process. Besides, as the interaction with the user is through the broker, the vendor is oblivious to which user he interacts with, so the vendor knows nothing about which user is interested in the specific discount information of the tested targeted coupons. Moreover, as the decryption of targeted coupons is performed on the user side, the vendor also learns nothing about the user’s eligibility status. As for the security against the user, it is ensured that non-eligible users learn nothing about the targeting profile except his non-eligibility status. From the SPAKE2-based protocol, non-eligible users obtain session keys not matching the vendor’s session keys and thus are not able to obtain the coupon encryption keys. Note that the broker simply helps route messages between the vendor and the user. The security of SPAKE2 ensures that these exchanged messages are indistinguishable from random values and the broker learns nothing.

VI. SECURE BLOCKCHAIN-EMPOWERED REDEMPTION OF TARGETED COUPONS

In this section, we explore how to leverage the emerging blockchain technology to support transparent yet privacy-preserving redemption of targeted coupons.

A. DESIGN RATIONALE

Recall that in our redemption process, there are three participating parties, including the user, the vendor, and the broker. The user is the one who initiates the redemption procedure to redeem one of his targeted coupons for commodity or service on the vendor side. A conventional redemption procedure would be as follows: After deciding which coupon to redeem, the user sends the selected coupon to the vendor for requesting redemption. Once receiving a redemption request, the vendor checks the coupon’s validity and accepts the request if the validation procedure passes. In the meantime, if the coupon is successfully redeemed, the user pays the price associated with the redeemed coupon, and the vendor pays the broker a certain service fee as a reward for successful coupon redemption.

However, the above conventional approach is vendor-centric and is not fair to the broker. In particular, the vendor may cheat in the coupon redemption rates so as to underpay the broker. In the extreme case, the vendor may cheat that no user has ever redeemed a coupon, so as to repudiate the service fee. Because the service fee will be automatically paid via the blockchain to the broker once a redemption request is validated, the vendor now is not able to cheat in coupon redemption rates and underpay or repudiate the service fee to the broker.

B. PROTOCOL OF BLOCKCHAIN-EMPOWERED SECURE COUPON REDEMPTION

We now present our blockchain-empowered secure coupon redemption. The user initiates the redemption process by sending a redemption request for a particular targeted coupon to the blockchain. The blockchain then verifies the coupon, including whether it has been redeemed before and whether the coupon is valid. If the coupon is verified, the blockchain automatically transfers money to the vendor and the broker accordingly. We assume that the blockchain initiates an empty redemption list I at the setup stage, and the public key of the vendor, i.e., pk

vendor

is correctly recorded. Also, we assume that the service fee for the broker per successful coupon redemption is fixed as \( \eta^* \), and that both the user and vendor have sufficient money deposited in their blockchain accounts. Here, we use Account

user

, Account

vendor

, and Account

broker

, to respectively denote the blockchain account of the user, the vendor, and the broker. We now describe the details of our protocol.

- **Redemption Request Generation:** The user computes the redemption request \( re \) for a targeted coupon \( m \), and sends it to the blockchain. Specifically,
  1. The user first computes the hash result \( H (m \parallel \zeta) \).
  2. Then, the user constructs the coupon redemption request \( re = \{H (m \parallel \zeta), \eta, \sigma\} \). Here, recall that \( \zeta \) is an identifier uniquely identifying the coupon, \( \eta \) is the price associated with the coupon, and \( \sigma = \text{sig} (H (H (m \parallel \zeta) \parallel \eta)) \) is a signature indicating the validity of the coupon. The user sends \( re \) to the blockchain.

- **Redemption Verification and Automatic Payment:** Upon receiving the redemption request \( re \), the processing on the blockchain proceeds as follows.
  1. The blockchain looks up the redemption list \( I \) to check whether \( \{ H (m \parallel \zeta), \sigma\} \) is in \( I \).
  2. If there is a hit, the blockchain rejects the redemption request.
  3. Otherwise, the blockchain computes \( \tau \leftarrow H(H (m \parallel \zeta) \parallel \eta) \) and verifies the attached signature \( \sigma \) via \( \text{Verify}_{pk\text{vendor}} (\tau, \sigma) \) to ensure that the coupon is indeed issued by the vendor and is valid.
4) If $\sigma$ is valid, the blockchain adds \{H(m \parallel \zeta), \sigma\} to the redemption list $I$. In the meantime, the blockchain transfers money from the user to the vendor via $Account_v \leftarrow Account_u$, and transfers money from the vendor to the broker via $Account_b \leftarrow Account_v$.

C. SECURITY ANALYSIS

The transparency and correctness of computation on the blockchain [24] ensure that the whole coupon redemption process is transparent and correctly enforced. This guarantees that a successfully redeemed coupon must be valid (through enforced signature verification) and cannot be reused; the vendor cannot cheat in the coupon redemption rates for underpaying/repudiating services fees to the broker. In addition, the targeted coupons are never exposed to the transparent blockchain. The blockchain only receives the hash that uniquely identifies a coupon and public information such as price information and signatures. So the content of a targeted coupon is well protected.

VII. EXPERIMENTS

A. EXPERIMENT SETUP

We implement our secure range query design in C++ and secure behavioral targeting in Python (v 2.7.6). We deploy the two programs on a machine which is equipped with Intel Xeon E5-2680 8-core CPU (2.7GHz) and 32 GB memory running Ubuntu version 16.04. To evaluate the performance of our secure coupon delivery design, we use the Ethereum test network, i.e., Rinkeby, to test the on-chain cost of our secure coupon redemption. The smart contract is implemented by the Ethereum programming language Solidity (v0.5.2). We use OpenSSL (v1.0.2g) for the implementation of cryptographic building blocks, including symmetric-key encryption via AES-256, pseudo-random function via HMAC-256. The signature function is implemented by the Elliptic-Curve Digital Signature algorithm (ECDSA). Here, we select a real-world dataset to evaluate the performance of our design. The test data comes from an open-source dataset for the coupon purchase prediction. It contains a year of transactional records for 22873 users, and we choose two sub-files from prediction files as the experimental dataset. The first file is “coupon_list_train.csv,” which includes the discount information for all coupons. The second file is “coupon_purchase_train.csv,” which contains users’ browsing logs and purchase logs. To test the efficiency of our encrypted discount search, we encrypt the column of “price_rate” in “coupon_list_train.csv,” and evaluate the search latency. For the behavior targeting test, we select two column records “purchase_flg” and “i_date” from “coupon_visit_train.csv” file. Here, we assume that the vendor decides to give the coupons to a customer based on the past purchase behavior and browsing behavior. In the coupon redemption, the customer randomly selects a coupon from the candidate coupons which have been correctly decrypted in the behavioral targeting process. For the “double payment” checking, the initial coupon redeemed list is formed by checking the “purchase_flg” in “coupon_visit_train.csv” file.

B. EVALUATION RESULTS

We now present the experiment results for different phases respectively, including system setup, encrypted range search on coupon discounts, privacy-preserving behavioral targeting, and secure blockchain-empowered coupon redemption.

1) SYSTEM SETUP

We first examine the performance on the system setup in which the vendor encrypts targeted coupons and stores them at the broker. In particular, we examine the storage overhead due to ORE ciphertexts for supporting encrypted range search and the computation overhead due to the generation of ORE ciphertexts.

a: STORAGE OVERHEAD

Recall that in the system setup phase, the vendor needs to take ORE_enc algorithm to encrypt the discounts of the coupons and outsources the encrypted coupons with the encrypted discounts to the broker. Since the coupon encryption algorithm is not the focus of our design (it can be easily implemented by AES), we only test the storage overhead of the encrypted discounts. Given that ORE_enc encrypts the discounts by blocks, the bit length of the block will directly affect the size of an ORE ciphertext. Therefore, we take the 2-bit, 4-bit and 8-bit parameter settings as the bit length of sub-block for the ORE encryption and examine the ciphertext size. To accurately assess the storage overhead due to the ORE encryption algorithm, we respectively select 3000, 6000, 9000, 12000, and 15000 coupons’ discount values from the “coupon_list_train.csv” dataset to test their storage overhead. Note that, the storage overhead of each discount value in the “coupon_list_train.csv” dataset takes 32 bits.

For the b-bit ORE design, each block contains $2^b - 1$ sub-blocks, where each is truncated as 32 bits. With a 128- bit nonce, the ciphertext size for a 32-bit discount is $128 + 32/b \times (2^b - 1) \times 32$ bits. From Table 1, we see that the experimental results are in line with expectations. In addition, the experimental results also show that the storage overhead of the

<table>
<thead>
<tr>
<th>No. of coupons</th>
<th>2-bit per sub-block (MB)</th>
<th>4-bit per sub-block (MB)</th>
<th>8-bit per sub-block (MB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>3000</td>
<td>0.60</td>
<td>1.42</td>
<td>11.72</td>
</tr>
<tr>
<td>6000</td>
<td>1.19</td>
<td>2.84</td>
<td>23.44</td>
</tr>
<tr>
<td>9000</td>
<td>1.79</td>
<td>4.26</td>
<td>35.16</td>
</tr>
<tr>
<td>12000</td>
<td>2.38</td>
<td>5.68</td>
<td>46.88</td>
</tr>
<tr>
<td>15000</td>
<td>2.98</td>
<td>7.10</td>
<td>58.59</td>
</tr>
</tbody>
</table>

\[\text{https://www.kaggle.com/c/coupon-purchase-prediction.}\]
encrypted discounts increases as the block length increases, but it is practically acceptable.

b: COMPUTATION OVERHEAD

Similar to the evaluation of storage overhead, we also take the 2-bit, 4-bit and 8-bit parameter settings to examine the computation overhead in encrypting the discounts of all the coupons. Here, we respectively select 800, 1600, 3200, 6400 and 12800 discounts from coupon_list_train.csv to test the computation overhead in our ORE enc algorithm.

Fig. 3(a) presents the time costs in the discount encryption over the above-mentioned five discount value datasets, respectively. In principle, the computation overhead is mainly dependent on the partitioning method and the number of discount values (i.e., number of coupons). As shown in Fig. 3(a), the experimental results are consistent with this principle. Specifically, for 800 discount values, the time cost increases from 0.035s to 0.107s when the parameter setting increases from 2-bit to 8-bit. In addition, the time cost for the 8-bit ORE design increases from about 0.41s to approximately 6.71s when the number of discount values increases from 0.8K to 12.8K. Although the time cost of the discount value encryption will continue to increase as the number of the discount values increases, our proposed design is still capable to handle the task of privacy-preserving range search. It is because that the system initialization phase is a one-time cost. The vendor only needs to encrypt the discount values once in the system lifetime. Besides, considering the security guarantee given by our ORE design, the computation overhead of our ORE encryption is still acceptable.

2) ENCRYPTED RANGE SEARCH ON COUPON DISCOUNTS

We now examine the performance of search latency when a user searches for coupons with discounts in a particular range pertaining to the user’s interests. Given that our coupon discount search algorithm is implemented by leveraging our ORE cmp algorithm to compare the token with each ciphertext of the discounts stored in the broker, we test the efficiency of our ORE cmp algorithm under different sizes of discount value datasets which are stored in the broker. Furthermore, we leverage the same datasets as those used to test the computation overhead of ORE enc to evaluate the impact of the discount dataset size on search efficiency. In addition, given that the parameter setting of ORE enc will influence the efficiency of ORE cmp, we respectively test the search efficiency under different parameter settings, including 2-bit, 4-bit and 8-bit.

Fig. 3(b) shows the search latencies under different settings. We find an interesting result that the 4-bit ORE scheme achieves a better efficiency than other ORE schemes. Specifically, when the number of discount values (i.e., number of coupons) is 1600, the query latency with the 4-bit ORE design is around 1.1s, which is roughly one-third of the latency with the 8-bit design. Meanwhile, it takes 8.72s to sequentially compare 12800 discount values with the token under 4-bit ORE design, while the 2-bit design requires 12.72s. Kindly note that the overhead mainly comes from the cost of blocks matching and cryptographic operations during the ORE comparison. Since the 4-bit design requires less comparison than the 2-bit design, it would be desirable to have better performance. However, this is not to say that the larger each block, the more efficient the range search. From Fig. 3(b), we observe that the ORE ciphertext size increases by almost 8.3x from the 4-bit design to the 8-bit design. It is because that reading a large size ciphertext will introduce a significant overhead. Thus, the 4-bit construction has a better performance than the 8-bit construction. Overall, the evaluation results shown in Fig. 3(b) confirms that our design can support secure coupon discount search efficiently.

3) PRIVACY-PRESERVING BEHAVIORAL TARGETING

We now examine the performance of privacy-preserving behavioral targeting. Recall that our privacy-preserving behavioral targeting design leverages SPAKE2 to enable eligible users to obtain the coupon decryption keys. Given that SPAKE2 can be used in the concurrent setting, we focus on measuring the computation cost on the user side for eligibility test over a varying number of targeted coupons. In specific, we randomly select 50, 100, 200, 400, and 800 coupons as the selected coupons to test the user’s computation costs in
the private behavioral targeting process so as to derive the coupon decryption keys.

Fig. 3(c) shows the user’ computation overhead in privacy-preserving behavioral targeting. In principle, the cost increases while the number of candidate coupons increases. This is because although the user only needs to compute the encrypted behavioral profile \(X^*\) once in the whole eligibility test process, he still needs to sequentially calculate the session keys \(s_i\) to decrypt each key \(k'_i\) used to encrypt the candidate coupons and decrypt each coupon with the decrypted key. From our experimental results, we can see that the experimental results fit the theoretical time complexity. For example, when the number of coupons is 800, the user only needs to spend 27.54s to test his eligibility. The results confirm that our design performs satisfactorily at the performance.

<table>
<thead>
<tr>
<th>Contract cost</th>
<th>Smart contract deployment</th>
<th>Redemption request submission</th>
<th>Redemption request verification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Transaction cost (gas)</td>
<td>976256</td>
<td>170709</td>
<td>56731</td>
</tr>
<tr>
<td>Execution cost (gas)</td>
<td>698172</td>
<td>142077</td>
<td>92189</td>
</tr>
<tr>
<td>Price (USD)</td>
<td>0.45</td>
<td>0.08</td>
<td>0.04</td>
</tr>
</tbody>
</table>

TABLE 2. On-chain cost of coupon redemption.

VIII. RELATED WORK
Our work is related to the line of works on ORE for secure range queries. In the literature, the early result ([26]–[28] to just list a few), also known as order-preserving encryption (OPE), only supports numeric comparison between the ciphertexts, and the orders are directly learned from ciphertexts. To improve security, the notion of ORE is proposed. The ORE ciphertexts themselves do not show order relations, so as to defend against inference attacks. The first ORE scheme is proposed by Boneh et al. [29]. However, it is impractical to use. Later, a practical ORE scheme [8] has been proposed to improve the efficiency of the ORE algorithm [29]. However, it leaks the first different bit of two messages. To mitigate this leakage, Lewi and Wu [17] propose an ORE scheme that achieves semantic security while leaks the first different bit block of two messages during comparisons. The core idea of their scheme is to split a message into bit blocks with equal length, and conduct encrypted comparison from the first blocks of two messages. Subsequently, Cash et al. [30] provide another different ORE scheme based on bilinear pairing to further reduce the leakage above, hiding the location of the first different block. In this work, we start from the latest practical ORE scheme [17] for secure range queries, and provide ORE comparison protocol with guaranteed security to support privacy-preserving range search on coupon discounts.

Our work is also akin to the research on privacy-aware targeted coupon delivery, where few works have been presented [4]–[6]. We note that these works all operate under the user-vendor setting and focus only on the support for privacy-preserving behavioral targeting. In contrast, we present the first full-fledged system design that simultaneously supports privacy-preserving range search on coupon discounts, privacy-preserving behavioral targeting, and privacy-preserving blockchain-empowered transparent coupon redemption. Note that for our design of privacy-preserving behavioral targeting, we leverage PAKE but work under a concurrent and broker-aided setting to fit with our target scenario, which is also different from prior works. We remark that the comprehensive practical functionalities make our design a strong candidate for deploying secure targeted coupon service in practice.

IX. CONCLUSION
In this paper, we explored and presented the first full-fledged system design for targeted coupons service. We started with the support for privacy-preserving range search on coupon discounts, through proposing a customized security design.
integrating the advanced cryptographic technique ORE. We then showed how to support privacy-preserving behavioral targeting, by building on the cryptographic technique SPAKE2 to efficiently support concurrent blind eligibility test on multiple targeted coupons. We further considered the problems residing in the conventional coupon redemption process, and explored a new secure blockchain-empowered coupon redemption design. We conducted extensive experiments, and the results validated the practically affordable performance of our system design.

REFERENCES

ANXIN ZHOU received the B.E. degree in computer science and technology from the University of Science and Technology of China, in 2018. He is currently a Research Assistant with the Department of Computer Science, City University of Hong Kong. His research interests include secure computation and data privacy.

CONG WANG received the B.E. degree in electronic information engineering from Wuhan University, China, and the M.E. degree in communication and information systems and the Ph.D. degree in electrical and computer engineering from the Illinois Institute of Technology, USA. He is currently an Associate Professor with the Department of Computer Science, City University of Hong Kong. His research has been supported by multiple government research fund agencies, including the National Natural Science Foundation of China, the Hong Kong Research Grants Council, and the Hong Kong Innovation and Technology Commission. His current research interests include data and computation outsourcing security in the context of cloud computing, blockchain and decentralized applications, network security in emerging Internet architecture, multimedia security, and privacy-enhancing technologies in the context of big data and the IoT. He is a member of the ACM. He is one of the Founding Members of the Young Academy of Sciences of Hong Kong. He received the Outstanding Research Award, in 2019, the Outstanding Supervisor Award, in 2017, and the President’s Awards from the City University of Hong Kong, in 2016. He was a corecipient of the Best Student Paper Award from the IEEE ICDCS 2017, and the Best Paper Award from the CHINACOM 2009, MSN 2015, and the IEEE ICPADS 2018. He has served as the TPC Co-Chair for a number of IEEE conferences/workshops. He serves/has served as an Associate Editor for the IEEE Transactions on Dependable and Secure Computing (TDSC), the IEEE Internet of Things Journal (IoT-J), and the IEEE Networking Letters.

XIAOLIN GUI received the B.E., M.E., and Ph.D. degrees from Xi’an Jiaotong University, Xi’an, in 1988, 1993, and 2001, respectively, where he is currently a Professor and the Deputy Dean of the School of Electronic and Information Engineering. His research interests include secure computation, data privacy, and the Internet of Things.