A Privacy-Preserving System for Targeted Coupon Service

YAO, Jing; ZHENG, Yifeng; GUO, Yu; CAI, Chengjun; ZHOU, Anxin; WANG, Cong; GUI, Xiaolin

Published in:
IEEE Access

Published: 01/01/2019

Document Version:
Final Published version, also known as Publisher’s PDF, Publisher’s Final version or Version of Record

License:
CC BY

Publication record in CityU Scholars:
Go to record

Published version (DOI):
10.1109/ACCESS.2019.2937669

Publication details:

Citing this paper
Please note that where the full-text provided on CityU Scholars is the Post-print version (also known as Accepted Author Manuscript, Peer-reviewed or Author Final version), it may differ from the Final Published version. When citing, ensure that you check and use the publisher's definitive version for pagination and other details.

General rights
Copyright for the publications made accessible via the CityU Scholars portal is retained by the author(s) and/or other copyright owners and it is a condition of accessing these publications that users recognise and abide by the legal requirements associated with these rights. Users may not further distribute the material or use it for any profit-making activity or commercial gain.

Publisher permission
Permission for previously published items are in accordance with publisher's copyright policies sourced from the SHERPA RoMEO database. Links to full text versions (either Published or Post-print) are only available if corresponding publishers allow open access.

Take down policy
Contact lbscholars@cityu.edu.hk if you believe that this document breaches copyright and provide us with details. We will remove access to the work immediately and investigate your claim.
A Privacy-Preserving System for Targeted Coupon Service

JING YAO\(^1,2\), YIFENG ZHENG\(^2,3\), YU GUO\(^2,3\), CHENJUN CAI\(^2,3\), ANXIN ZHOU\(^2,3\), CONG WANG\(^2,3\), AND XIAOLIN GUI\(^1\)

\(^1\)School of Electronic and Information Engineering, Xi'an Jiaotong University, Xi'an 710049, China
\(^2\)Department of Computer Science, City University of Hong Kong, Hong Kong
\(^3\)City 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. It is performed in a privacy-preserving manner, so that eligible users are able to decrypt targeted coupons located via range search, and non-eligible users learn nothing except their non-eligibility. Afterwards, a user who successfully obtains a targeted coupon can redeem it. To protect the business interests of the broker, our system deals with coupon redemption via the blockchain. This makes the whole redemption process transparent as well as automatically enforce the payment settling among the user, vendor, and broker. We propose a working design which protects on-chain data privacy while well supporting verification of coupon validity as required in coupon redemption.

B. SECURITY THREATS AND GOALS

We describe the threats from each party as considered in our system. The broker will honestly follow our protocol, yet attempts to infer the 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** $(sk, d)$: 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 $c$. This is the core design of an ORE scheme, which decides the data security and the data comparison efficiency.

**ORE** $(c_1, c_2)$: 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] ensures 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 these 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.

Motivated by the observations above, our goal is to devise a new secure and efficient range query scheme. For security, our design builds upon the existing ORE scheme proposed by Lewi et al. [17], which is the most secure design in the practical area. Then, we observe that the main reason for the possible attacks is that the order information is revealed after ORE comparison. Therefore, we aim to hide the comparison result in queries and results. This property can be achieved by tokenizing the orders in each ORE block. On the other hand, we propose to hide the location of the first different blocks. As recognized in [17], this leakage primarily comes from the ORE comparison mechanism, i.e., block-by-block comparison in sequence. To reduce this leakage, we apply secure permutation on the encrypted blocks. But simply permuting those blocks could cause mismatches. To address this issue, we propose to embed the hash value of each block’s prefix into the block cipher. Due to the uniqueness of the prefix in each block, the ORE token matching operation can still correctly be performed even blocks are shuffled.

### 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. $\{v_1^*, \ldots, v_d^*\}$ 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 $\Pi = \{\text{ORE}_{\text{enc}}, \text{ORE}_{\text{token}}, \text{ORE}_{\text{cmp}}\}$ is defined as follows:

- **ORE** $(k, v||C)$: As shown in Algorithm 1, the encryption performed on the vendor side proceeds as follows. Firstly, the vendor generates the private keys $\{k_1, k_2\}$...

### Algorithm 1 ORE** $(c_{\text{enc}})$: ORE Ciphertext Encryption

- **Input**: Private key $k$; hash function $h$; discount value $v$; attribute $C$; block values $\{v_1^*, \ldots, v_d^*\}$; order $\text{cmp} \in \{>, <\}$; secure PRP $\pi$; secure PRF $\{F, G_1, G_2\}$.
- **Output**: ORE ciphertext $ct_g$.

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


Algorithm 2 ORE_{token}: ORE Token Generation

**Input:** private key \( k \); order condition \( cmp \in \{>, <\} \); discount value \( v \); coupon attribute \( C \); secure PRP \( \pi \); secure PRF \( \{ G_1, G_2 \} \).

**Output:** ORE query token \( c_{tL} \).

1. \( k_1 \leftarrow F(k, C||1), k_2 \leftarrow F(k, C||2) \).
2. for \( i \in \{1, b\} \) do
   3. \( t_i \leftarrow G_1(k_1, v_i||cmp) \).
   4. \( c_{tL||i} \leftarrow G_2(k_2, t_i||v_i||1) \).
5. \( c_{tL} \leftarrow [c_{tL||1}, \ldots, c_{tL||b}] \).

Algorithm 3 ORE_{cmp}: ORE Ciphertext Comparison

**Input:** ORE query token \( c_{tL} \), ORE ciphertext \( c_{tR} \).

**Output:** true or false.

1. \( c_{tL||1}, \ldots, c_{tL||b} \leftarrow c_{tL} \).
2. \( c_{tR||1}, \ldots, c_{tR||b}, \gamma \leftarrow c_{tR} \).
3. for \( i \in \{1, b\} \) do
   4. \( z_{i,1}, \ldots, z_{i,2^d-1} \leftarrow c_{tR||i} \).
5. for \( j \in \{1, 2^d-1\} \) do
6. \( flg \leftarrow z_{ij} \oplus h(c_{tL||i}||\gamma) \).
7. if \( flg == "true" \) then
8. Output true; // return coupon to client
9. Output false;

Based on the coupon attribute \( C \). Then, the vendor divides the discount value \( v \) into \( b \) blocks with \( d \) bit length, and each sub-block with the comparison result \( v_i^s||c_{cmp} \) is secretly embedded with the prefix value via pseudorandom functions (PRFs) which are functions that produce pseudorandom output indistinguishable from random sequences. After that, the vendor adds a check flag “true” to the hash value of the above entry by XORing operation. Finally, the vendor permutes all the blocks via a pseudorandom permutation (PRP) \( \pi \), a function that is indistinguishable from a random permutation, and outputs the ORE ciphertext \( c_{tR} \).

1. **ORE_{token}(k, v||C||cmp):** Given a discount value \( v \) with an attribute \( C \) and the order condition \( cmp \in \{>, <\} \), token generation performed on the user side is illustrated in Algorithm 2. In particular, the user firstly divides the discount value \( v \) into \( b \) blocks with \( d \) bits equal length. For each block \( i \in \{1, b\} \), the user encrypts the block value \( v_i \) with the order \( cmp\), and embeds it with the prefix value. Finally, the user permutes each block randomly, and outputs the discount query token \( c_{tL} \).
2. **ORE_{cmp}(c_{tL}, c_{tR}):** Given the ORE query token \( c_{tL} = \{c_{tL||1}, \ldots, c_{tL||b}\} \) and the ORE ciphertext \( c_{tR} = \{c_{tR||1}, \ldots, c_{tR||b}, \gamma\} \), the search procedure on the broker side is shown in Algorithm 3. In particular, the broker first tries to find the matched block pair that satisfies \( c_{tL||i} = c_{tR||i} \) by XORing each sub block of \( c_{tR||i} \) with the hash value \( h(c_{tL||i}||v) \). If it exists, the check flag “true” will be revealed, and the corresponding coupon will be returned to client. Otherwise, the broker outputs “false.”

Our ORE construction guarantees that the broker learns neither order relations across different discounts, nor two different requests are conducted in the same order. Besides, it can protect the position of the first different block by random permutation. Therefore, recent attacks that rely on these leakages can hardly be launched anymore. Due to the deterministic property of PRF, the discount comparison can still correctly be performed via token matching.

C. PROTOCOL OF PRIVACY-PRESERVING RANGE SEARCH FOR TARGETED COUPONS

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 algorithm 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 \( \{k, k_1\} \) by using the key generation function \( KeyGen(\lambda^2) \), where \( \lambda \) is the security parameter.
  2. The vendor encrypts each targeted coupon \( m_i \) by computing \( cm_i \leftarrow SE(k_1^\prime, m_i || \xi || \eta || \sigma) \), 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., \( c_{tR} \leftarrow ORE_{enc}(k, v_i||C_i) \).
  4. The vendor sends \( \{cm_i, c_{tR}\} \) 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:
1) The user encrypts his query by computing 
\[ ct_l \leftarrow \text{OREtoken}(k, v|||C|||\text{cmp}) \] 
and sends the \( ct_l \) to the broker.

2) The broker compares \( ct_l \) with each ORE ciphertext \( ct_R \) in turn by using \( \text{OREcmp}(ct_l, ct_R) \). The broker adds the 
\( cm_i \) to the candidate coupon collection, if the check flag 
of \( ct_R \) 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 cipher-
texts 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 
discount information “10% off” is expressed as “1010” in 
binary form. It is divided into two blocks \( \{10, 10\} \). Each 
block is embedded with the order information and the prefix 
values via two secure PRF \( G_1 \) and \( G_2 \). The ORE comparison 
algorithm shows two cases. In the first case, to search for the 
coupons of which the discounts are more than 5% off, the user 
first generates the ORE token of value “0101” based on the 
order condition “>,” as shown in Fig. 2(a). Then the broker 
seeks the matched block via XORing the token with the encrypted 
discount by entry. The result shows that the 
highlighted entries will reveal the check flag “true,” which 
represents that the coupon’s discount is more than 5% off. 
In the second case, to search for the coupons of which the 
discount values are more than 15% off, the user first generates 
the ORE token of value “1111” based on the order condition 
“>,” as shown in Fig. 2(a). Then the server attempts to find 
the matched block via XORing the token with the encrypted 
discount by entry. The result shows there is no entry matching 
the token which means that the coupon’s discount is no more than 
15% off.

**D. SECURITY ANALYSIS**

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 
onymmetric-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 inform-
ation 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 defini-
tion. 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 ciphertexts 
\( n \), 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 \( L_1 \).

\[
L_1(\text{ct}_R) = (b, d, |ct_R|, |z|, \gamma)_{\text{ideal}}
\]

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 \( z_{i,j} \) \((i \in \{1, b\}, j \in \{1, d\})\) between the search token 
and the encrypted discount \( ct_R \), the check flag “true,” and 
the unique nonce \( \gamma \). The leakage \( L_2 \) is defined as:

\[
L_2(v|||C|||\text{cmp}) = (\text{"true"}, |z_{i,j}|, \gamma)_{\text{ideal}}
\]

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 \( M_{i,i} \) 
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 \( \text{Real}_A(k) \) and \( \text{Ideal}_{A,S}(k) \):

\( \text{Real}_A(k) \): The vendor executes \( \text{KeyGen}(1^k) \) for private 
keys. \( A \) selects a set of coupons and asks the vendor to encrypt 
their discounts via \( \text{Build} \) (algorithm 1). Then \( A \) conducts 
a polynomial number of \( q \) queries and asks the vendor for
secure tokens and query results via Query (algorithm 3). Finally, A returns a bit as the output.

Ideal \( A_{LS}(k) \): 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: \( \Pr[\text{Real}_A(k) = 1] - \Pr[\text{Ideal}_A_{LS}(k) = 1] \leq \text{negl}(k) \), where negl(\( k \)) is a negligible function in \( k \).

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

Proof: Given \( L_1 \), the simulator \( S \) generates the simulated discount \( \triangleright_p \), which is indistinguishable from the real one \( \triangleright_R \). Specifically, the sizes of \( \triangleright_R \) is \( |\triangleright_R| \), 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” \( \oplus h(z'_{ij} || y) \), where \( z'_{ij} \) is a random string as the simulated token. It can be derived from \( O_{G_2}(O_{G_1}(v_i cmp) || v_{i-1}) \), where \( O_{G_1}, O_{G_2} \) are random oracles. Meanwhile, \( S \) updates \( M_{1,1} = 1 \) 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 \( M_{1,j} \) 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.

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 ciphertext. We are aware that adding dummy ciphertext can mitigate this attack effectively. Meanwhile, we 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.

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.

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 user 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 \{cm₁, \cdots, cmₙ\} 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 \(s_i\) to encrypt the coupon encryption key \(k_{1i}\) 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 \(m_v\) 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 \(k_{1i}\).

- **PRIVATE BEHAVIORAL TARGETING:** The broker randomly generates an element \(m_u \in G\) 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 \(Y^a_i = g^{y_i} m_v^{h_1(b_i)}\) for the targeting profile \(b_i\) of each tested targeted coupon \(cm_i\), where \(y_i \in R Z_p\).

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

  3) The vendor computes \(K_{vi} = (X^* / m_u^{h_2(b_i)} y_i)^{s_{vi}}\) for each \(i \in \{1, \cdots, l\}\). 

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

  5) The user computes \(K_i = (Y^* / m_u^{h_1(a)})^x\) and the session keys \(s_i = h_2(ID_u, ID_v, X^*, Y^*_{1i}, K_i)\), for each \(i \in \{1, \cdots, l\}\). Then the user decrypts each ciphertext \(c_{k_{1i}}\) via \(k_{1i} = c_{k_{1i}} \oplus s_i\), and obtains the correct coupon plaintext \(m_i\) (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 payment to the broker. We note that the above problems are due to the lack of transparency in the redemption process, so our insight is to make the redemption of targeted coupons transparent. We propose to leverage the emerging blockchain technology to support transparent coupon redemption, so as to enable the broker to know whenever a redemption request occurs and that he is entitled to earn the service fee.

At a high level, instead of validating each targeted coupon on the vendor side, we conduct the validation process transparently on the blockchain. For on-chain data privacy protection, the user will provide a hash of the targeted coupon to the blockchain for validation rather than the targeted coupon in cleartext. Once the coupon is validated, the redeemed coupon will be logged and the service fee will be automatically paid to the broker. The blockchain will also check whether a coupon has been redeemed before so as to prevent coupon reuse. If it is detected that a coupon has been used redeemed, the redemption request of the user will be rejected. 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 $\eta \ast$, and that both the user and vendor have sufficient money deposited in their blockchain accounts. Here, we use Account$_u$, Account$_v$, and Account$_b$, 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 $H(m || \zeta)$.

2) Then, the **user** constructs the coupon redemption request $re = \{H(m || \zeta), \eta, \sigma\}$. Here, recall that $\zeta$ is an identifier uniquely identifying the coupon, $\eta$ is the price associated with the coupon, and $\sigma = \text{sig}(H(H(m || \zeta) || \eta))$ 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 $\{H(m || \zeta), \sigma\}$ is in $I$.

2) If there is a hit, the blockchain rejects the redemption request.

3) Otherwise, the blockchain computes $\pi \leftarrow H(H(m || \zeta) || \eta)$ and verifies the attached signature $\sigma$ via $\text{Verify}_{pk_{vendor}}(\pi, \sigma)$ 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 $L$. In the meantime, the blockchain transfers money from the user to the vendor via Account$_{u}\leftarrow Account_{v}$, 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/reputiduating 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 a year of transactional records for 6000, 9000, 12000 and 15000 users, and we choose two sub-files from prediciton files as the experimental dataset. The first file is 120826

| TABLE 1. Storage overhead of ORE ciphertexts under varying parameter settings. |
|---|---|---|---|
| No. of coupons | 2-bit per sub-block (MB) | 4-bit per sub-block (MB) | 8-bit per sub-block (MB) |
| 3000 | 0.60 | 1.42 | 11.92 |
| 6000 | 1.19 | 2.84 | 23.44 |
| 9000 | 1.79 | 4.26 | 35.16 |
| 12000 | 2.38 | 5.68 | 46.88 |
| 15000 | 2.98 | 7.10 | 58.59 |

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

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 $8.3 \times$ 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.

4) SECURE BLOCKCHAIN-EMPOWERED COUPON REDEMPTION

Recall that the blockchain is leveraged in our system for transparent coupon redemption and business fraud prevention. The blockchain-empowered coupon redemption process consists of two main functions, i.e., 1) redemption request submission which records user’s request on the blockchain, and 2) redemption request verification which verifies the validity of the coupon request and automatically rewards the vendor and broker if the request is valid. To evaluate the real cost of our coupon redemption process, we implement and deploy the above two functions (in the form of smart contracts) on the Ethereum test network, i.e., Rinkeby, and evaluate how many gases have been consumed. In particular, there are two types of costs on the Rinkeby, including transaction cost and execution cost, which is related to the size of transaction data and the computational operations in the Ethereum virtual machine (EVM) respectively. As of the time of writing, the gas price of Ethereum equals to $2 \times 10^{-9}$ ether and each ether worths $269.06$ USD [25]. We conduct the experiment with 100 randomly selected coupons in the ‘‘coupon_list_train.csv.’’ The experimental results are depicted in Table 2.

Table 2. On-chain cost of coupon redemption.

<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>

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.

VIII. RELATED WORK

IX. CONCLUSION
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 achievable performance of our system design.

REFERENCES


[13]']


[29] J. Yao received the B.E. degree from the Xi’an University of Technology, Xi’an, China, in 2009, and the M.E. degree from Shannxi Normal University, Xi’an, in 2012. She is currently pursuing the Ph.D. degrees with the School of Electronic and Information Engineering, Xi’an Jiaotong University and with the Department of Computer Science, City University of Hong Kong, Hong Kong. Her research interests include cloud computing security, outsourcing storage security, and data privacy.

YIFENG ZHENG received the B.E. degree in information engineering from the South China University of Technology, Guangzhou, China, in 2013, and the Ph.D. degree in computer science from the City University of Hong Kong, Hong Kong, in 2019. From 2014 to 2015, he was a Research Assistant with the Department of Computer Science, City University of Hong Kong, where he is currently a Research Fellow. His current research interests include security in cloud media applications, security in the IoT applications, and privacy-aware computing. He received the Outstanding Teaching Award for Teaching Assistants, the Outstanding Academic Performance Award, and the Chow Yei Ching School of Graduate Studies Scholarship from the City University of Hong Kong, in 2018.

YU GUO received the B.E. degree in software engineering from Northeastern University, in 2013, and the M.S. degree in electronic commerce and the Ph.D. degree in computer science from the City University of Hong Kong, in 2014 and 2019, respectively, where he is currently a Postdoctoral Fellow. His research interests include cloud computing security, search-able encryption, and privacy-preserving data processing.

CHENGJUN CAI received the B.S. degree in computer science and technology from Jinan University, in 2016. He is currently pursuing the Ph.D. degree with the City University of Hong Kong, where he was a Research Assistant. His research interests include distributed system security and privacy-enhancing technologies.
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.