

Privacy-Preserving Outsourced Profiling

Debmalya Biswas¹, Stephan Haller², Florian Kerschbaum¹

¹ SAP Research CEC Karlsruhe, Vincenz-Priessnitz-Strasse 1, 76131 Karlsruhe, Germany

² SAP Research CEC Zurich, Kreuzplatz 20, 8008 Zurich, Switzerland

firstname.lastname@sap.com

Abstract

Personalized services attract high-value customers. Knowing the preferences and habits of an individual customer, it is possible to offer to that customer well customized and adapted services, matching his needs and desires. This is advantageous for the entity offering the service (e.g., a retailer) as well, as it helps in creating additional sales or improve customer retention. The main unsolved problem today is that the profile of each individual customer would be necessary in order to create such services, posing severe risks regarding privacy and data protection. This paper proposes efficient encryption schemes that allow profiling to be outsourced while preserving privacy. The schemes ensure that the customer is always in control of his profile data, at the same time making shopping data across multiple retailers available to third party service providers to be able to provide targeted services.

I. Introduction

Personalized services attract high value customers. Profiling is the process of generating personalized services adapted to match the shopping needs and preferences of individual customers. Personalized services are advantageous for a customer as he is presented only with offers he might actually be interested in (compared to being spammed with general advertisements), and as such helps in enhancing the overall shopping experience. The entity providing the service also benefits as it helps create additional sales and improves customer retention.

A recent survey [1] confirms the above observation: “Advertisers that are spending premiums to target the biggest spenders and the most frequent shoppers should take note that those shoppers want intelligent ads that speak to their specific needs and shopping intent. The more

personalized those ads are the better chance retailers have of connecting with those customers.”

While the added value of personalized services is evident, the process of profiling, i.e. generating personalized services itself is non-trivial. The main obstacle is that a profile of each individual customer is necessary in order to create such personalized services, posing severe risks regarding privacy and data protection. Current solutions [2], [3] only consider the shopping history of a customer with respect to a specific retailer while generating recommendations. Clearly, the profiling accuracy increases as more data is available. In an ideal scenario, personalized services for a customer should be generated based on his shopping history across multiple retailers, as well as data of other customers with “similar” interests. Unfortunately, the lack of trust prevalent in today’s Business-to-Customer (B2C) environment prevents customers from sharing their shopping history with retailers due to fear of privacy loss, possibility of spamming and even data theft, among others. Even retailers are unwilling to share their respective customer shopping data with other retailers due to competitive reasons.

Our proposal overcomes the above limitations by enabling profiling in a privacy preserving fashion. The proposed solutions allow generating highly personalized services based on shopping data of customers across multiple retailers. The solutions promote data sharing between customers and retailers, as well as among multiple retailers, by providing strict security guarantees that privacy will not be violated.

From a security perspective, the problem is related to “searching over encrypted data”. This is an active research area in the field of cryptography and various *PEKS* (Public-key Encryption with Keyword Search) schemes [4], [5] have been proposed. Applying a *PEKS* scheme, a retailer would store the shopping data of its customers in a shared space in encrypted form. The retailer then provides trapdoors to authorized third party

service providers allowing them to search over the stored encrypted data with respect to specific keywords. However, such an application of *PEKS* leaves the customer completely out of the loop, which would never be acceptable to a privacy conscious customer. We extend *PEKS* in such a way that the customer can not only verify but also select his shopping data that gets exposed to the service providers. We defer a detailed comparison with related works to Section VI.

The rest of the paper is organized as follows. In Section II, we formulate the problem domain including the security requirements of the different parties involved. Section III presents our first encryption scheme *PERK*. In Section IV, we provide a detailed outline of how *PERK* can be applied to perform privacy preserving profiling. In Section V, we present an alternate encryption scheme *PERK_d* that allows privacy preserving profiling with pre-defined keywords, agreed up on in advance between the retailers and service providers. Section VI discusses related work and Section VII concludes the paper.

II. Problem Formulation

The problem domain consists of customers buying products at retailers. The customers are interested in receiving services personalized to their shopping needs and preferences. We refer to the providers responsible for profiling the shopping data across retailers and serving ads to the customers as the service providers. In real-life, a retailer and service provider may be the same entity. The profiling performed by service providers can be with respect to a specific retailer, e.g. a service provider could compile a shopping list for the customer, matching the types of regularly bought items with what is currently on sale at the retailer.

To enable profiling, the retailers need to share the shopping transactions data of the customers at their respective stores with the service providers. This sharing is achieved by each retailer uploading the customer shopping data to a shared storage space. The storage provider is responsible for providing this shared storage space where all the customer data including personal contact details and shopping data are stored. The storage provider is thus in a position to charge the customers, retailers and even the service providers for providing the storage facility. Technically, the customer might also want to sell his data, e.g. to market analysts (a special type of service provider, though the provided service in this case is for the retailers, the customer only gets monetary benefits). We do not expand on the various possible payment scenarios here, rather focusing on the technical details needed to make privacy preserving profiling a reality.

To ensure that false customer data is not exposed by

the retailers (by mistake or intentionally) to the service providers, the customers should have an opportunity to verify the integrity of uploaded data before it is exposed to the service providers. The customers can also use this opportunity to select portions of their shopping data that becomes available to the service providers for profiling. The customers are thus always in control of their data.

To summarize (Fig. 1), the different parties in our ecosystem and their privacy requirements are as follows:

- Customers (*C*): The customers would like to remain as anonymous as possible. So it should not be possible for a retailer to relate the various shopping transactions performed by a customer at its store. Further, the customer should be in a position to:
 - verify the integrity of his shopping data uploaded by the retailers, and
 - Control which of his shopping data is exposed to the service providers.
- Retailers (*R*): While the retailers agree to make the shopping transaction details of their customers available to the service providers, they would like to restrict access to this data only to authorized customers and service providers. The authorization of customers is by purchase history, i.e. a customer should only be allowed access over shopping transactions data pertaining to his purchases. The authorization of service providers is by keywords (detailed below). In particular, it should not be possible for a retailer to access the data of other retailers.
- Storage provider (*S*): The storage provider itself should not be able to access any of the customer data stored on its servers.
- Service providers (*P*): The service providers should only have access to the shopping data exposed by the customers (and not the data uploaded by the retailers). Further, we consider that each service provider caters to specific products, categorized by a set of keywords. Given this, a service provider should only be able to access the exposed customer shopping data containing his authorized keywords.

III. Public Key Encryption with Proxy Re-encryption and Keyword Search (*PERK*)

We define and construct our first encryption scheme Public Key Encryption with Proxy Re-encryption and Keyword Search (*PERK*) in this section. A *PERK* scheme consists of the following polynomial time randomized algorithms:

- $KGEN(1^k)$ outputs a public-private key pair: (A_{pub}, A_{priv}) .

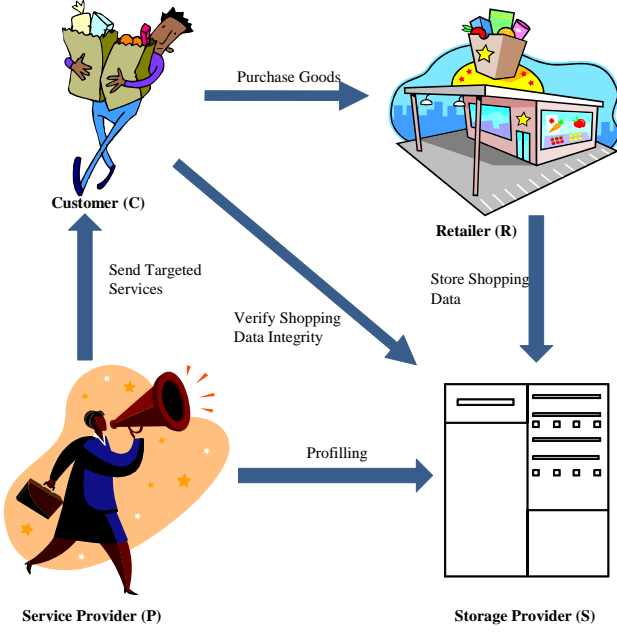


Fig. 1. Problem scenario

- $ENC(A_{pub}, m)$ outputs c_{A1} , the message m encrypted under public key A_{pub} .
- $PRK(A_{priv}, B_{pub})$ outputs a re-encryption key $rk_{A \rightarrow B}$ that allows ciphertexts generated using A 's public key to be decrypted by B 's private key.
- $RENC(rk_{A \rightarrow B}, c_{A1})$ outputs the ciphertext c_{B2} generated by re-encrypting c_{A1} under $rk_{A \rightarrow B}$.
- $DEC(B_{priv}, c_{B2})$ decrypts c_{B2} using B_{priv} , returning the message m .
- $SENC(A_{pub}, W, m)$ outputs a searchable encryption s_W of message m under keyword W and public key A_{pub} .
- $DOOR(A_{priv}, W)$ outputs a trapdoor t_W that allows to search by keyword W .
- $TEST(A_{pub}, s_W, t_{W'})$ outputs the message m if $W = W'$.

We give a construction of the *PERK* scheme based on the Boneh et. al. [4] *PEKS* and Ateniese et. al [6] proxy re-encryption schemes. The construction uses bilinear maps as defined below:

Definition 1 (Bilinear Maps): We say a map $\hat{e} : G_1 \times G_1 \rightarrow G_2$ is a bilinear map if

- 1) G_1 and G_2 are groups of the same prime order p .
- 2) given $g \in G_1$, there is a polynomial time algorithm to compute $\hat{e}(g, g) \in G_2$.
- 3) for all integers $x, y \in \mathbb{Z}_p, g \in G_1$, we have $\hat{e}(g^x, g^y) = \hat{e}(g, g)^{xy}$.

- 4) the map is non-degenerate (i.e., if g generates G_1 , then $\hat{e}(g, g)$ generates G_2).

□

The algorithms in our *PERK* scheme can now be constructed as follows: We define hash functions $H_1 : \{0, 1\}^* \rightarrow G_1$ and $H_2 : G_2 \rightarrow \{0, 1\}^{log p}$.

- $KGEN(1^k)$ generates bilinear maps G_1 and G_2 of size p based on the security parameter k . The algorithm then picks random $\alpha, \beta \in \mathbb{Z}_p^*$, a generator g of G_1 , and $h = \hat{e}(g, g) \in G_2$. It outputs $A_{pub} = (g, g^\alpha, h^\beta)$ and $A_{priv} = (\alpha, \beta)$.
- $ENC(A_{pub}, m)$ encrypts $m \in G_2$ as $c_{A1} = (g^k, mh^{\beta k})$.
- $PRK(A_{priv} = (\alpha, \beta), B_{pub} = (g, g^\gamma, h^\delta))$ generates $rk_{A \rightarrow B} = g^{\beta \gamma}$.
- $RENC(rk_{A \rightarrow B}, c_{A1})$ re-encrypts $c_{B2} = (h^{\beta \gamma k}, c_{A1} h^{\beta k})$, where $h^{\beta \gamma k} = \hat{e}(g^k, g^{\beta \gamma})$.
- $DEC(B_{priv} = (\gamma, \delta), c_{B2})$ outputs $m = y/x^{1/\gamma}$.
- $SENC(A_{pub}, W, m)$: Compute $s_W = (g^r, m \oplus H_2(t^r))$, where $r \in \mathbb{Z}_p^*$ and $t = \hat{e}(H_1(W), g^\alpha)$.
- $DOOR(A_{priv}, W)$ outputs $t_W = H_1(W)^\alpha$.
- $TEST(A_{pub}, s_W, t_{W'})$: Let $s_W = (x = g^r, y = m \oplus H_2(t^r))$. If $W = W'$,

$$\begin{aligned}
 TEST(A_{pub}, s_W, t_W) &= y \oplus H_2(\hat{e}(t_W, x)) \\
 &= y \oplus H_2(\hat{e}(H_1(W)^\alpha, g^r)) \\
 &= y \oplus H_2(\hat{e}(H_1(W), g)^{\alpha r}) \\
 &= y \oplus H_2(\hat{e}(H_1(W), g^\alpha)^r) \\
 &= y \oplus H_2(t^r) \\
 &= (m \oplus H_2(t^r)) \oplus H_2(t^r) \\
 &= m.
 \end{aligned}$$

Security. We define security for the *PERK* scheme in the sense of standard semantic security as follows:

- 1) Only user B can decrypt ciphertexts c_{B2} (re-encrypted using $rk_{* \rightarrow B}$). Note that the encrypted ciphertexts c_{A1} are exactly like El-Gamal [7], and as such their security only depends on Decisional Diffie-Hellman (DDH) in G_2 .
- 2) $SENC(A_{pub}, W, m)$ does not reveal any information with respect to W unless t_W is available.

The proof of security relies on the difficulty of the Decisional Co-Diffie-Hellman (CoDDH) problem.

Definition 2 (Decisional Co-Diffie-Hellman (CoDDH)): The CoDDH problem is defined as follows: Given $(g, g^\alpha, g^\beta, h^\gamma, Q)$ for a generator g of G_1 , $h = \hat{e}(g, g), Q \in G_2$ and random $\alpha, \beta, \gamma \in \mathbb{Z}_p^*$, decide if $Q = h^{\alpha \beta \gamma}$. □

Note that the security of our scheme relies on the difficulty of the CoDDH problem as compared to the Decisional Bilinear Diffie-Hellman (DBDH) problem intractability required for the *PEKS* scheme in [4]. This is because of the additional parameter h^β in the public key of our encryption scheme required for re-encryption.

Theorem 1: The encryption scheme *PERK* = (*KGEN*, *ENC*, *PRK*, *RENC*, *DEC*, *SENC*, *DOOR*, *TEST*) is semantically secure against a chosen plaintext attack in the random oracle model assuming CoDDH is intractable. \square

In the next section, we describe in detail how the *PERK* encryption scheme can be applied to perform privacy preserving profiling.

IV. Privacy Preserving Profiling with *PERK*

We divide the profiling application scenario into the following phases: Fig. 2 gives a pictorial description of the application scenario, highlighting the interactions between the different parties. The step numbers below correspond to the step numbers in Fig. 2.

Initial Setup:

- 1) Retailer R and customer C run the algorithm $KGEN(1^k)$ to generate their respective public-private key pairs: (R_{pub}, R_{priv}) and (C_{pub}, C_{priv}) .
- 2) Customer C registers with storage provider S : C sends his public key C_{pub} and contact details on which he would like to receive notifications to S . Various modes of communication need to be accommodated, e.g. emails, pushing information to mobile phones, among others. Specifically, we need the preferred communication modes for verification and advertised service messages (e.g. a customer may like to receive verification messages via email, and advertised services on mobile phone). Let the preferred modes of communication for verification and service messages are C_V and C_{Ad} , respectively. Assuming C_V and C_{Ad} are unique, a new customer record is created for C . The reference u_C of this newly created record is then returned to C .
- 3) Customer C registers with retailer R : C sends the pair $\langle C_{pub}, u_C \rangle$ to R .
- 4) R generates the re-encryption key $rk_{R \rightarrow C} = PRK(R_{priv}, C_{pub})$ for C and stores it in the record referenced by u_C .

For each shopping transaction performed by C at R :

- Store C 's shopping transaction data at the designated storage space S_D provided by S :
 - 5) C presents a unique id r_{C_i} to R . r_{C_i} can be the reference of the storage record where the

details of this shopping transaction will be stored at S_D . Assuming S pre-allocates storage space for each registered customer, C can procure such transaction ids in bulk in advance from S . This is very likely in the scenario where the customer has to pay S for providing storage services.

- 6) R encrypts the shopping transaction data d_{C_i} of C under its public key R_{pub} generating the ciphertext $c_{R1} = ENC(R_{pub}, d_{C_i})$. R then stores the encrypted data c_{R1} in the record referenced by r_{C_i} at S_D .
- Verify the integrity of stored shopping transaction data at S_D :
 - 7) S updates the record referenced by r_{C_i} , re-encrypting c_{R1} under $rk_{R \rightarrow C}$, generating the ciphertext $c_{C2} = RENC(rk_{R \rightarrow C}, c_{R1})$.
 - 8) S sends a notification to C via C_V , notifying him that new shopping transaction data is now available for verification at the data record referenced by r_{C_i} .
 - 9) C accesses the ciphertext c_{C2} from S_D by reference r_{C_i} .
 - 10) C then decrypts c_{C2} using his secret key C_{priv} to obtain $d_{C_i} = DEC(C_{priv}, c_{C2})$. C is now in a position to verify the integrity of his stored shopping transaction data d_{C_i} at S_D .
- Generate searchable encryptions: C generates searchable encryptions of his shopping data d_{C_i} with respect to the keywords he would like to make available for profiling by the service providers. For each chosen keyword f ,
 - 11) C generates a corresponding searchable encryption value $s_f = SENC(C_{pub}, f, d_{C_i})$. C then updates the shopping transaction record referenced by r_{C_i} at S_D , inserting the ciphertext s_f .
 - 12) C gets the list of authorized service providers P_1, \dots, P_n with respect to keyword f from S .
 - 13) C generates the trapdoor $t_f = DOOR(C_{priv}, f)$, and sends it to P_1, \dots, P_n .

Profiling by the service providers:

- 14) For each authorized keyword f , service provider P compares his trapdoor values t_f with the searchable encryption values s_f exposed by each customer C . On successful match, P obtains the shopping transaction data $d_{C_i} = TEST(C_{pub}, s_f, t_f)$.
- 15) P uses the acquired shopping data of customers to perform profiling.
- 16) P provides targeted services to customers, notifying each customer C via C_{Ad} . \square

(Step 13) Note that the trapdoors need to be generated only once per (authorized) service provider, and not for

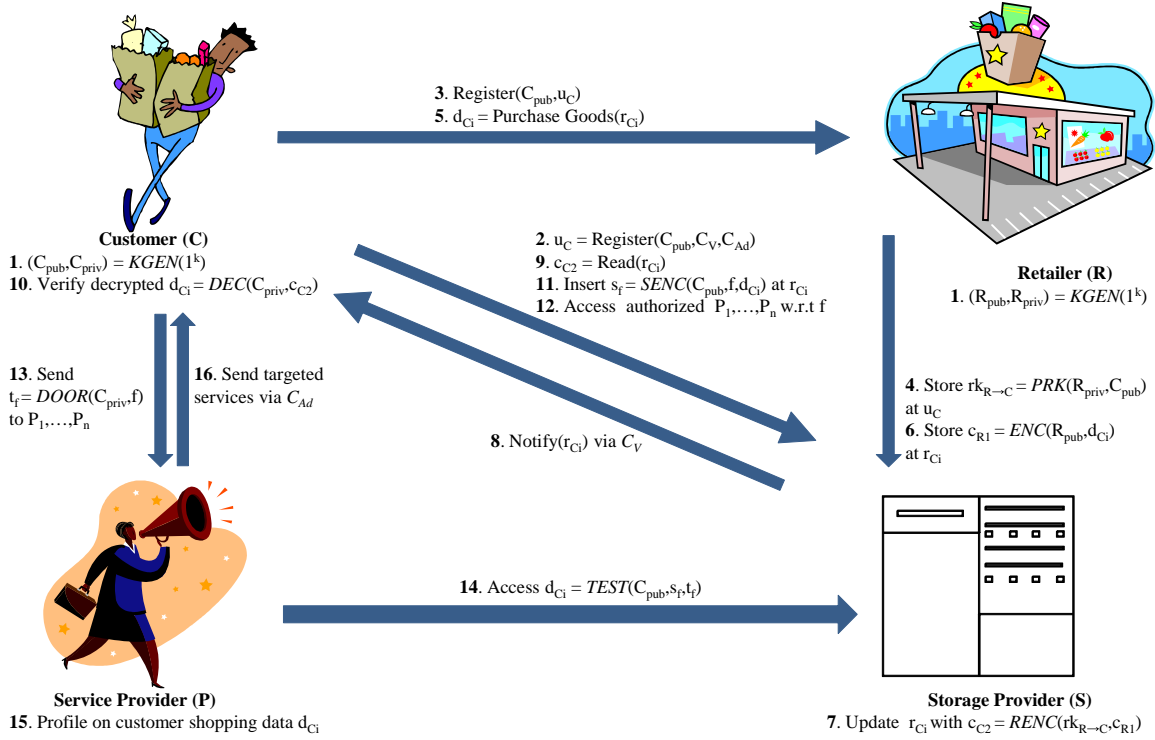


Fig. 2. Applying *PERK* to perform privacy preserving profiling

each transaction. If the set of keywords with respect to which C would like expose his shopping data to P remains constant over time, then the trapdoors can also be generated as part of the initial setup.

V. Privacy Preserving Profiling with Pre-defined Keywords

In this section, we present an alternate scheme where the retailers and service providers agree on a pre-defined set of keywords for profiling. This scheme is particularly relevant when the retailer and service provider are the same entity. The customer can still regulate for which keywords his shopping data becomes available to a service provider for profiling, however he can only choose among the set of pre-defined keywords (i.e., he cannot add any keywords that are not in the pre-defined set of keywords). The storage provider no longer needs to provide access control for the service providers (with respect to their authorized keywords). We consider a more ad-hoc setting where a service provider can directly approach a customer to get access to his shopping data with respect to some of the pre-defined keywords, possibly in return for some monetary benefits for the customer. The monetary benefit can of course also be in the form of targeted services of

interest to the customer.

As before, we first give the construction for such an encryption scheme and then present its usage.

A. Public Key Encryption with Proxy Re-encryption and Pre-defined Keywords Search (*PERK_d*)

A *PERK_d* scheme consists of the following polynomial time randomized algorithms:

- $KGEN_d(1^k)$ outputs a key: A_{key} .
- $ENC_d(A_{key}, m)$ outputs c_{A1} , the message m encrypted under key A_{key} .
- $PRK_d(A_{key}, B_{key})$ outputs a re-encryption key $rk_{A \rightarrow B}$ that allows ciphertexts generated using A 's key to be decrypted by B 's key.
- $RENC_d(rk_{A \rightarrow B}, c_{A1})$ outputs the ciphertext c_{B2} generated by re-encrypting c_{A1} under $rk_{A \rightarrow B}$.
- $DEC_d(B_{key}, c_{B2})$ decrypts c_{B2} using B_{key} , returning the message m .
- $DOOR_d(B_{key}, m)$ outputs a trapdoor t_m that allows to search with respect to message m .
- $TEST_d(m_{B2}, t_{m'})$ outputs 'YES' if $m = m'$.

The algorithms in the *PERK_d* scheme can be constructed as follows: The construction as in case of *PERK*

is also based on bilinear maps (See Definition 1).

- $KGEND_d(1^k)$ generates bilinear maps G_1 and G_2 of size p based on the security parameter k . The algorithm then picks random $\alpha, \beta \in Z_p^*$, a generator g of G_1 . It outputs $A_{key} = (\alpha, \beta)$.
- $ENC_d(A_{key}, m)$ encrypts $m \in G_2$ as $c_{A1} = (x = g^r, y = m^\beta g^{r\alpha})$, where $r \in Z_p^*$.
- $PRK_d(A_{key} = (\alpha, \beta), B_{key} = (\gamma, \delta))$ generates $rk_{A \rightarrow B} = (\eta = \gamma - \alpha, \lambda = \delta/\beta)$.
- $RENC_d(rk_{A \rightarrow B}, c_{A1})$ re-encrypts $c_{B2} = (x', y')$, where $x' = x^\lambda = g^{r\lambda}$ and $y' = y^\lambda x'^\eta = m^\delta g^{r\lambda\gamma}$.
- $DEC_d(B_{key} = (\gamma, \delta), c_{B2})$ outputs $m = (y'/(x'g^\gamma))^{1/\delta}$.
- $DOOR_d(B_{key}, m)$ outputs $t_m = (t_1, t_2, t_3)$, where $t_1 = g^s, t_2 = g^{s\gamma}, t_3 = \hat{e}(g^s, m^\delta)$ for random $s \in Z_p^*$.
- $TEST_d(m_{B2}, t_{m'})$: If $m = m'$,

$$\begin{aligned} \hat{e}(g^s, y') &= t_3 \hat{e}(g^{s\gamma}, x') \\ \Rightarrow \hat{e}(g^s, m^\delta g^{r\lambda\gamma}) &= \hat{e}(g^s, m^\delta) \hat{e}(g^{s\gamma}, g^{r\lambda}) \\ \Rightarrow \hat{e}(g^s, m^\delta g^{r\lambda\gamma}) &= \hat{e}(g^s, m^\delta g^{r\lambda\gamma}). \end{aligned}$$

Security.

Theorem 2: The encryption scheme $PERK_d = (KGEND_d, ENC_d, PRK_d, RENC_d, DEC_d, DOOR_d, TEST_d)$ is semantically secure against a chosen plaintext attack. \square

We do not give the proof here due to space restrictions. We suffice it to say that the proof depends on standard security assumptions, and does not need any new hardness assumptions.

B. Privacy Preserving Profiling with $PERK_d$

The $PERK_d$ scheme can be applied to perform privacy preserving profiling as follows: Let \mathcal{F} denote the set of pre-defined keywords. Fig. 2 gives a pictorial description of the application scenario, with the step numbers below corresponding to the step numbers in the figure.

Initial Setup:

- 1) Retailer R and customer C run the algorithm $KGEND_d(1^k)$ to generate their respective keys: $R_{key} = (\alpha, \beta)$ and $C_{key} = (\gamma, \delta)$.
- 2) C registers with S : C sends a random $r \in Z_p^*$ and his contact details C_V and C_{Ad} (for verification and service messages, respectively) to S . Assuming the contact details are unique, a new customer record is created for C . The reference u_C of this newly created record is then returned to C .
- 3) C registers with R : C sends the triplet $\langle \gamma + r, \delta r, u_C \rangle$ to R .

- 4) R stores the pair $\langle \gamma - \alpha + r, \delta r/\beta \rangle$ in the record referenced by u_C .
- 5) S extracts the re-encryption key $rk_{R \rightarrow C} = (\gamma - \alpha + r - r, \delta r/\beta r)$ for C with respect to R , and updates the record referenced by u_C .

For each shopping transaction performed by C at R :

- Store C 's shopping transaction data at the designated storage space S_D provided by S :
 - 6) C presents a unique storage reference r_{Ci} to R .
 - 7) For each pre-defined keyword $f \in \mathcal{F}$ present in the shopping transaction data d_{Ci} of C , R stores the ciphertext $f_{R1} = ENC_d(R_{key}, f)$ in the record referenced by r_{Ci} at S_D .
- Verify the integrity of stored shopping transaction data at S_D :
 - 8) S updates the record referenced by r_{Ci} , re-encrypting each f_{R1} under $rk_{R \rightarrow C}$, generating the ciphertext $f_{C2} = RENC_d(rk_{R \rightarrow C}, f_{R1})$.
 - 9) S sends a notification to C via C_V , notifying him of his newly added shopping data at r_{Ci} .
 - 10) C accesses the ciphertext f_{C2} from S_D by reference r_{Ci} .
 - 11) C verifies each keyword f by decrypting $f = DEC_d(C_{key}, f_{C2})$.
- Generate trapdoors:
 - 12) For each keyword f requested by a service provider P , C generates the trapdoor $t_f = DOOR_d(C_{key}, f)$ and sends it to P .

Profiling by the service providers:

- 13) For each pre-defined keyword f , service provider P checks the presence of f in C 's shopping records by comparing his trapdoor value t_f with the encrypted keyword values f_{C2} stored at S_D using $TEST_d(f_{C2}, t_f)$.
- 14) P uses this acquired keyword-based shopping data for profiling.
- 15) P sends targeted services to customers C via C_{Ad} . \square

Comparison with profiling based on $PERK$ (Section IV).

The first noticeable difference is clearly the lack of an $SENC$ operation in the $PERK_d$ scheme. Recall that on applying the $PERK$ scheme, a customer C is in a position to choose the keywords with respect to which he would like to make his data available to the service providers. C generates searchable encryptions and provides corresponding trapdoors to the service providers for his chosen set of keywords (Steps 11-13 in Section IV). With $PERK_d$, it is actually the retailer who generates the searchable encryptions of customer shopping data, with respect to the set of pre-defined keywords \mathcal{F} (Step 7). The customer can regulate this data by only providing trapdoors

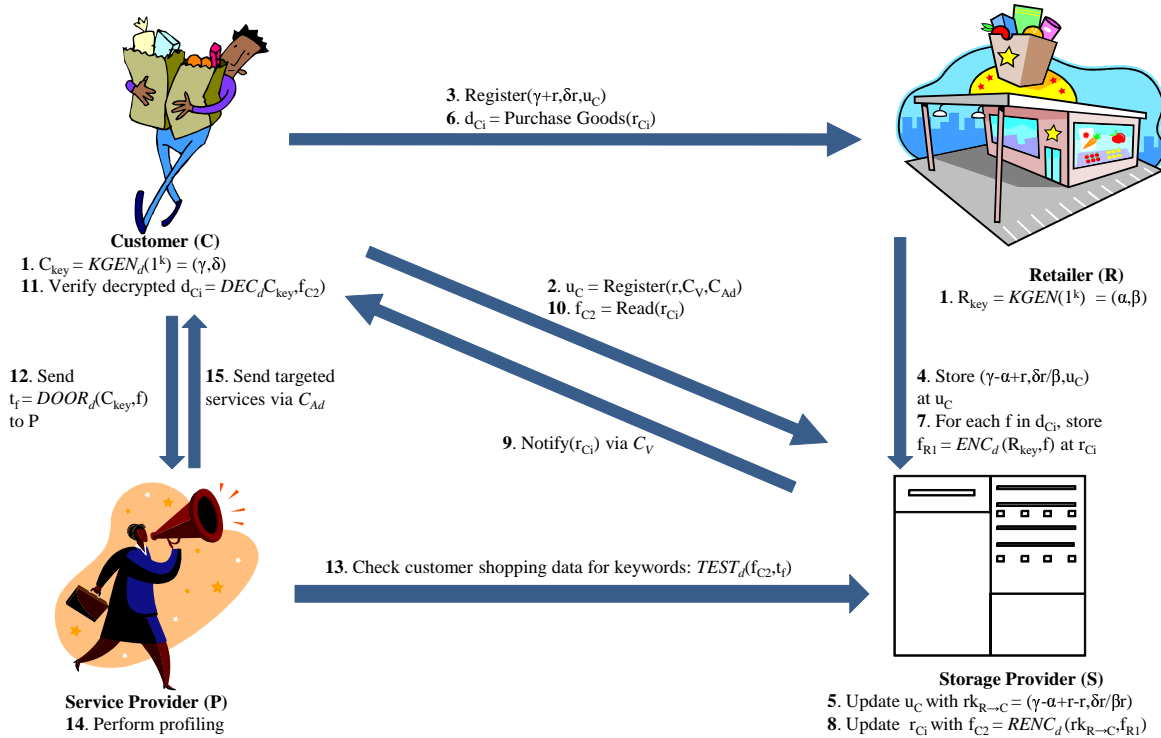


Fig. 3. Applying $PERK_d$ to perform privacy preserving profiling

of his chosen set \mathcal{F}_C of keywords to the service providers (Step 12). Note that $\mathcal{F}_C \subseteq \mathcal{F}$, and C cannot choose a keyword $f \notin \mathcal{F}$. The other main difference is with respect to the type of data that finally becomes available for profiling to the service providers. While $PERK$ (Step 14 in Section IV) outputs the whole shopping record on a successful match, $PERK_d$ (Step 13) only lets the service providers know if a shopping record of C contains a specific keyword.

VI. Related Work

We mainly build on two primitives from cryptography in this work: “searching over encrypted data” and “proxy re-encryption (delegation)”.

The commonly used encryption scheme to search over encrypted data is referred to as $PEKS$. Most of the recent $PEKS$ implementations [4], [5] are actually based on Identity-Based Encryption (IBE) schemes. Shamir [8] first introduced a public-key encryption scheme where any publicly known string (e.g., email address) could be used as a public key. In 2001, Boneh and Franklin [9] provided the first practical implementation of an IBE scheme based on bilinear maps. Both [4], [5] show how an IBE scheme can be adapted to $PEKS$, with [4] showing that $PEKS$

implies IBE (with the reverse probably false). As argued earlier, $PEKS$ by itself is not sufficient for our application scenario. $PEKS$ can for instance be used by a retailer to outsource profiling to a third party service provider, however there is no scope for the customer to intervene in this process. We extend $PEKS$ with re-encryption capability to enable the customer to regulate his exposed shopping data.

Proxy re-encryption [10], [6] is a related research area that allows a ciphertext for A to be re-encrypted into a ciphertext for B (can be decrypted using B ’s secret key). The envisioned application of such an encryption scheme is delegation [11], e.g. an employee can delegate his confidential encrypted emails to his secretary, without any need to forward his secret key. As with $PEKS$, such delegation schemes by themselves are not sufficient for our application scenario. We only use delegation between the retailer and customer. Technically, it is possible for the customer as well to use proxy re-encryption to make his data available to the service providers. However, this is clearly not very efficient requiring a re-encrypted version of each customer shopping record for each authorized service provider to be stored, leading to a lot of redundancy.

From a profiling/personalization perspective, recent research [2], [3] has focused on improving the accuracy of

recommendations [12]. Various techniques such as behavioral/content based profiling have been used to improve the quality of recommended services. These works are complementary to our work as we only aim to increase the volume of source data available for profiling, irrespective of the technique used. Our encryption schemes enable privacy preserving profiling on shopping data of multiple customers across multiple retailers.

[13], [14] also study privacy preserving profiling, however their application scenario is online advertising, and as such their challenges and the techniques proposed are not directly relevant here. The online advertising ecosystem consists of the following main parties: customer, advertiser, publisher, dealer, ad-network (service provider in our case). To advertise their products across multiple domains, advertisers get in touch with an ad-network. Publishers are the domain owners hosting ads served by the ad-network. The ad-network is responsible for both serving ads to customers as well as monitoring their clicks. The ad-network is thus in a position to profile customer access data across his partner advertisers and publishers. However, the source data collection mechanism here is clearly very different from our application scenario. Basically, all customer access data is collected directly by the (centralized) ad-network as compared to our scenario where the retailers first collect customer shopping data (at their respective stores) and then share it with the service providers. The dealer is responsible for anonymizing customer access to prevent the brokers from identifying the customers. Again, the anonymization mechanisms used are only relevant for protecting the online privacy of customers, e.g. hiding network addresses, protecting cookies, etc.

VII. Conclusion

In this paper, we presented two encryption schemes to solve the important business problem of providing highly personalized services to customers. The services are generated based on customer shopping history at multiple retailers in a privacy preserving fashion. Our schemes offer the following advantages over state-of-the-art:

- Profiling across multiple retailers, not limited to a single retailer.
- Full privacy control over own profile by the customer. Others (retailers, service providers, etc.) only get access to portions of the shopping data that the customer explicitly gives them access to, and only in a pseudonymised fashion.
- From a security perspective, this is the first proposal to combine proxy re-encryption and searchable encryption schemes.

Acknowledgement

This work is partly funded by the European Commission through the ICT program under Framework 7 grant 213531 to the SecureSCM project.

References

- [1] "Survey finds personalized ads attract high-value customers," <http://www.choicestream.com/news/pressrelease.asp?id=84>, 2009.
- [2] C. M. Cumby, A. E. Fano, R. Ghani, and K. Marko, "Building intelligent shopping assistants using individual customer models," *In proceedings of the International Conference on Intelligent User Interfaces (IUI)*, pp. 323–325, 2005.
- [3] G. Adomavicius and A. Tuzhilin, "Using data mining methods to build customer profiles," *IEEE Computer*, 34(2), pp. 74–82, 2001.
- [4] D. Boneh, G. D. Crescenzo, R. Ostrovsky, and G. Persiano, "Public key encryption with keyword search," *In proceedings of the International Conference on the Theory and Applications of Cryptographic Techniques (EUROCRYPT)*, pp. 506–522, 2004.
- [5] J. Camenisch, M. Kohlweiss, A. Rial, and C. Sheedy, "Blind and anonymous identity-based encryption and authorised private searches on public key encrypted data," *In proceedings of the International Conference on the Practice and Theory in Public Key Cryptography (PKC)*, pp. 196–214, 2009.
- [6] G. Ateniese, K. Fu, M. Green, and S. Hohenberger, "Improved proxy re-encryption schemes with applications to secure distributed storage," *In proceedings of the Network and Distributed System Security Symposium (NDSS)*, pp. 29–44, 2005.
- [7] T. E. Gamal, "A public key cryptosystem and a signature scheme based on discrete logarithms," *In proceedings of Advances in Cryptology (CRYPTO)*, pp. 10–18, 1984.
- [8] A. Shamir, "Identity-based cryptosystems and signature schemes," *In proceedings of Advances in Cryptology (CRYPTO)*, pp. 47–53, 1984.
- [9] D. Boneh and M. K. Franklin, "Identity-based encryption from the weil pairing," *In proceedings of Advances in Cryptology (CRYPTO)*, pp. 213–229, 2001.
- [10] A. Ivan and Y. Dodis, "Proxy cryptography revisited," *In proceedings of the Network and Distributed System Security Symposium (NDSS)*, 2003.
- [11] S. Wohlgenuth and G. Muller, "Privacy with delegation of rights by identity management," *In proceedings of the Emerging Trends in Information and Communication Security (ETRICS)*, pp. 175–190, 2006.
- [12] "Special issue on recommender systems," *Communications of the ACM*, 40(3), 1997.
- [13] V. Toubiana, A. Narayanan, D. Boneh, H. Nissenbaum, and S. Barocas, "Adnostic: Privacy preserving targeted advertising," *In proceedings of the Network and Distributed System Security Symposium (NDSS)*, 2010.
- [14] S. Guha, A. Reznichenko, K. Tang, H. Haddadi, and P. Francis, "Serving ads from localhost for performance, privacy, and profit," *In proceedings of Hot Topics in Networking (HotNets)*, 2009.