SCIENCE CHINA
Information Sciences

• **Supplementary File** •

An Enhanced Searchable Encryption Scheme for Secure Data Outsourcing

Rui Zhang[1,2], Jiabei Wang[1,2*], Zishuai Song[1,2] & Xi Wang[1,2]

[1]*State Key Laboratory of Information Security (SKLOIS), Institute of Information Engineering (IIE), Chinese Academy of Sciences (CAS), Beijing* 100093*, China;*
[2]*School of Cyber Security, University of Chinese Academy of Sciences, Beijing* 100049*, China*

Appendix A The Deterministic Blind Signature

We apply the pseudorandom function to this transformation so that the correctness of our concrete SA-SCF-PECKS scheme can be guaranteed. The transformed blind signature scheme is also correct, (one-more) unforgeable and perfectly blind in the standard model. Besides, the scheme is efficient and round-optimal. We first present the deterministic blind signature scheme, then prove that this scheme is a secure blind signature in the standard model briefly.

Appendix A.1 Deterministic Blind Signature Scheme (*deBS*)

- $KeyGen_{deBS}(1^{\lambda})$:
  - It generates parameters $P_{deBS} = (p, g, \hat{g}, G, \hat{G}, T, e, F^1, F^2)$, where $G, \hat{G}$ are two groups with prime order p. g, $\hat{g}$ are the generators of group G and $\hat{G}$, respectively, and F^1, F^2 are two different pseudorandom functions (PRFs) for the user and the signer respectively. The bilinear map $e : G \times \hat{G} \rightarrow T$ is used.
  - It selects $\bar{h}, x, y \xleftarrow{\$} \mathbb{Z}_P^*$, $k_S \xleftarrow{\$} \{0,1\}^{\lambda}$, computes $(H, \hat{H}, \hat{X}, \hat{Y}) = (g^{\bar{h}}, \hat{g}^{\bar{h}}, \hat{g}^x, \hat{g}^y)$, and lets $(pk_{deBS} = (H, \hat{H}, \hat{X}, \hat{Y})$, $sk_{deBS} = (k_S, \bar{h}, x, y))$ as signer's public/private key pair.
  - It selects value $k_U \xleftarrow{\$} \{0,1\}^{\lambda}$, and sets k_U as user's secret key.
- $User\text{-}Request(P_{deBS}, k_U, pk_{deBS}, m)$:
  - It returns $\perp$ if $H = 1_G$ or $e(H, \hat{g}) \neq e(g, \hat{H})$;
  - It computes $r \leftarrow F^1_{k_U}(m)$, and $Com = g^m H^r$;
  - Finally, it returns $(\xi = Com, st = (m, r))$.
- $Signer\text{-}Issue(P_{deBS}, sk_{deBS}, \xi)$:
  - It first computes $a' \leftarrow F^2_{k_S}(Com)$, and sets $\bar{\sigma} = (A', B', C') = (g^{a'}, (g^x Com)^{\frac{a'}{y}}, H^{\frac{a'}{y}})$;
  - It finally returns $\bar{\sigma}$ to the user.
- $User\text{-}Process(P_{deBS}, k_U, pk_{deBS}, \bar{\sigma}, st)$:
  - It returns $\perp$ if $A' = 1_G$ or $e(C', \hat{Y}) \neq e(A', \hat{H})$;
  - It sets $B' = B'C'^{-r}$;
  - It returns $\perp$ if $e(B', \hat{Y}) \neq e(A', \hat{X}\hat{g}^m)$;
  - It computes $a \leftarrow F^1_{k_U}(r)$, and returns $\sigma_m = (A, B) = (A'^a, B'^a)$.
- $Verify_{deBS}(P_{deBS}, pk_{deBS}, m, \sigma_m)$:
  - It returns 0, if $e(B, \hat{Y}) \neq e(A, \hat{X}\hat{g}^m)$;
  - It returns 1, otherwise.

* Corresponding author (email: wangjiabei@iie.ac.cn)

Appendix A.2 Analysis of *deBS*

Theorem 1. *deBS* is a secure blind signature scheme in the standard model.

Proof. Firstly, the *deBS* scheme is correct. From the algorithms above, we have $Com = g^m H^r$, $B' = (g^x Com)^{\frac{a'}{y}} = g^{\frac{a'x}{y}} Com^{\frac{a'}{y}} = g^{\frac{a'x}{y}} (g^m H^r)^{\frac{a'}{y}}$ and $C' = H^{\frac{a'}{y}}$. Then, in *User-Request* we have $B' = B'C'^{-r} = g^{\frac{a'x}{y}} (g^m H^r)^{\frac{a'}{y}} H^{\frac{-a'r}{y}} = g^{\frac{a'x}{y}} g^{\frac{a'm}{y}}$. So, if (A', B') is valid, then it satisfies $e(B', \hat{Y}) = e(A', \hat{X}\hat{g}^m)$. Besides, it is easy to show that *deBS* is deterministic (satisfies our definition). Combining Lemma 1 and 2, the proof is complete.

Lemma 1 (One-More Unforgeability)**.** *deBS* is (one-more) unforgeable if the Blind Signature One More (BSOM) assumption (see in [1]) is intractable and the functions F^1, F^2 are pseudorandom.

Proof. The construction I for a single message in [1] has been proven unforgeable based on the BSOM assumption. The only difference between *deBS* and that scheme is that the randomly chosen values r, a', and a in [1] while in *deBS* are generated by the pseudorandom functions F^1, F^2. For the property that the output of pseudorandom function is indistinguishable from the real randomness. We can easily prove *deBS* is also unforgeable.

Lemma 2 (Blindness)**.** *deBS* is perfectly blind if the functions F^1, F^2 are pseudorandom.

Proof. For the pseudorandomness of functions F^1, F^2, we can easily follow the proof in [1] to prove *deBS* is also perfectly blind. We omit the details here for the sake of space.

Appendix B A Brief Review to Previous schemes

We take a brief review to Fang *et.al*'s SCF-PEKS scheme [2] and Zhang *et al.*'s PECSK Scheme [3] here, as shown in Table B1 and Table B2.

Table B1 Fang *et.al*'s SCF-PEKS scheme [2]

System Setup:	generate related parameters as $Param = (g, \mathbb{G}, \mathbb{G}_1, p, e, h)$; $e : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_1$, p is the prime oder of $\mathbb{G}$, $\mathbb{G}_1$, g is a generator of $\mathbb{G}$; $h : \{0,1\}^* \rightarrow \mathbb{Z}_p^*$ is a collision resistant hash function, and keyword $w \in \mathbb{Z}_p^*$.	
Key Generation:	Server:	choose $r_{s,1} \xleftarrow{\$} \mathbb{Z}_p^*$, $r_{s,2} \xleftarrow{\$} \mathbb{G}^*$; compute $s = g^{r_{s,1}}$, let $pk_S = (s, r_{s,2})$, $sk_S = r_{s,1}$.
	Querier:	choose $r_{q,1} \xleftarrow{\$} \mathbb{Z}_p^*$, $r_{q,2} \xleftarrow{\$} \mathbb{G}^*$; compute $q = g^{r_{q,1}}$, let $pk_{QU} = (q, r_{q,2})$, $sk_{QU} = r_{q,1}$.
Data Owner $(Param, pk_S, pk_{QU}, w)$		Server $(Param, sk_S)$
Encryption: choose $r_{o,1}$, $r_{o,2} \xleftarrow{\$} \mathbb{Z}_p^*$; compute $C_{w,1} = g^{r_{o,1}}$; $temp = h(e(s, r_{s,2})^{r_{o,1}})$; $C_{w,2} = (q \cdot g^{-w})^{\frac{r_{o,2}}{temp}}$; $C_{w,3} = e(g,g)^{r_{o,1}}$; $C_{w,4} = e(g, r_{q,2})^{r_{o,2}}$; let $C_w = (C_{w,1}, C_{w,2}, C_{w,3}, C_{w,4})$.	$\xrightarrow{\text{ciphertext } C_w}$	*Test*: compute $temp = h(e(C_{w,1}, r_{s,2})^{r_{s,1}})$; check if $e((C_{w,2})^{temp}, d_{T_w})(C_{w,3})^{r_{T_w}} = C_{w,4}$; if yes, output "1"; otherwise, output "0".
Querier $(Param, sk_{QU}, w)$		
Trapdoor generation: choose $r_{T_w} \xleftarrow{\$} \mathbb{Z}_p^*$; compute $d_{T_w} = (r_{q,2} \cdot g^{-r_{T_w}})^{\frac{1}{r_{q,1}-w}}$; let $T_w = (r_{T_w}, d_{T_w})$.	$\xrightarrow{\text{trapdoor } T_w}$	

Table B2 Zhang *et al.*'s PECSK Scheme [3]

System Setup:		initialize parameters as $Param = (p, G_1, G_2, G_3, e, H_1, H_2)$; $e : G_1 \times G_2 \rightarrow G_3$, p is the prime order of G_1, G_2, G_3; $H_1 : \{0,1\}^* \rightarrow \mathbb{Z}_p^*$, $H_2 : G_3 \rightarrow \mathbb{Z}_p^*$.
Key Generation:		let m be the fixed number of keywords in the encryption algorithm; choose $p_0, p_1, ..., p_m \xleftarrow{\$} G_1$, $g_{G_1,1}, g_{G_1,2} \xleftarrow{\$} G_1$, $g_{G_2} \xleftarrow{\$} G_2$; $r_q \xleftarrow{\$} \mathbb{Z}_p^*$, and compute $q = (g_{G_2})^{r_q}$; let $pk_{QU} = (g_{G_1,1}, g_{G_1,2}, g_{G_2}, q, p_0, p_1, ..., p_m)$, $sk_{QU} = r_q$.
Data Owner $(Param, pk_{QU}, W = (w_1, ..., w_m))$		Server $(Param)$
Encryption: choose $r_{o,1}$, $r_{o,2} \xleftarrow{\$} \mathbb{Z}_p^*$; construct polynomial as: $F(x) = r_{o,1} \cdot (x - H_1(w_1)) \cdots (x - H_1(w_m)) + r_{o,2}$; $= \theta_0 + \theta_1 x + ... + \theta_m x^m$; choose $r_{o,3} \xleftarrow{\$} \mathbb{Z}_p^*$, compute $C_0 = (g_{G_2})^{r_{o,3} \cdot r_{o,2}}$; $C_1 = H_2(e(g_{G_1,2}, g_{G_2})^{(\theta_0+\theta_1+...+\theta_m) \cdot r_{o,3}})$; for $i = 0$ to m, $C_{q,i} = q^{\theta_i \cdot r_{o,3}}$; for $i = 0$ to m, $C_{p,i} = p_i^{\theta_i \cdot r_{o,3}}$; let $C_W = (C_0, C_1; C_{q,0}, C_{q,1}, ..., C_{q,m}; C_{p,0}, C_{p,1}, ..., C_{p,m})$.	$\xrightarrow{\text{ciphertext } C_W}$	*Test*: compute $S_1 = \prod_{i=0}^m e(T_i, C_{q,i})$; $S_2 = e((g_{G_1,1})^{rt}, C_0)$; $S_3 = \prod_{i=0}^m e(C_{p,i}, q^{rt}) = \prod_{i=0}^m e(p_i^{\theta_i \cdot r_q \cdot r_{o,3}}, q^{rt})$; check if $H_2(S_1/(S_2 \cdot S_3)) = C_1$ if yes, output "1"; otherwise, output "0".
Querier $(Param, sk_{QU}, Q = (w_1', ..., w_m'))$		
Trapdoor generation: choose $r_t \xleftarrow{\$} \mathbb{Z}_p^*$; compute $T_0 = (g_{G_1,2})^{1/r_q} \cdot ((g_{G_1,1})^{(H_1(w_1')^0+...+H_1(w_s')^0)/r_q \cdot s} \cdot p_0)^{rt}$; $T_1 = (g_{G_1,2})^{1/r_q} \cdot ((g_{G_1,1})^{(H_1(w_1')^1+...+H_1(w_s')^1)/r_q \cdot s} \cdot p_1)^{rt}$; ... $T_m = (g_{G_1,2})^{1/r_q} \cdot ((g_{G_1,1})^{(H_1(w_1')^m+...+H_1(w_s')^m)/r_q \cdot s} \cdot p_m)^{rt}$; let $T_Q = (T_0, T_1, ..., T_m, (g_{G_1,1})^{rt}, q^{rt})$.	$\xrightarrow{\text{trapdoor } T_Q}$	

References

1 Ghadafi E. Efficient round-optimal blind signatures in the standard model. International Conference on Financial Cryptography and Data Security. Springer, Cham, 2017: 455-473
2 Fang L, Susilo W, Ge C, et al. A secure channel free public key encryption with keyword search scheme without random oracle. International Conference on Cryptology and Network Security. Springer, Berlin, Heidelberg, 2009: 248-258
3 Zhang B, Zhang F. An efficient public key encryption with conjunctive-subset keywords search. Journal of Network and Computer Applications, 2011, 34(1): 262-267