A Note on the Lindell-Waisbard Private Web Search Scheme

Zhengjun Cao[1], Lihua Liu[2,*]

Abstract. In 2010, Lindell and Waisbard proposed a private web search scheme for malicious adversaries. At the end of the scheme, each party obtains one search word and query the search engine with the word. We remark that a malicious party could query the search engine with a false word instead of the word obtained. The malicious party can link the true word to its provider if the party publicly complain for the false searching result. To fix this drawback, each party has to broadcast all shares so as to enable every party to recover all search words and query the search engine with all these words.

We also remark that there is a very simple method to achieve the same purpose of private shuffle. When a user wants to privately query the search engine with a word, he can choose another $n-1$ padding words to form a group of n words and permute these words randomly. Finally, he queries the search engine with these words.

Keywords. Private web search; private shuffle; ElGamal encryption.

1 Introduction

As we see private web search has become a serious problem. The anonymous routing system [5] can be used for this problem but it is somewhat inefficient. So do the private information retrieval [2, 9] and mix-net [1, 7, 4]. In 2009, Castellà-Roca et al. [3] suggested a new approach for the problem. Their proposal is for a group of users to shuffle their search words amongst themselves. After the shuffle, each user has someone's search word (but doesn't know whose), and the parties then query the search engine with the word obtained. Finally, the parties all broadcast the result to all others. Their private shuffle protocol is secure only in the presence of semi-honest adversaries.

In 2010, Lindell and Waisbard [8] pointed out that the scheme suggested by [3] is unrealistic because it is vulnerable to many attacks. They proposed a private shuffle protocol for malicious adversaries and proved its security according to their security definition. They also addressed some practical considerations. At the end of the Lindell-Waisbard scheme, like the previous work [3], each party obtains only one search word and query the search engine with the word.

[01]Department of Mathematics, Shanghai University, Shanghai, China. [2]Department of Mathematics, Shanghai Maritime University, Shanghai, China. * liulh@shmtu.edu.cn

In this note, we would like to remark that a malicious party could query the search engine with a false word instead of the word obtained. Thus the party corresponding to the true word can not obtain the proper searching result. More worse, the malicious party can link the true word to its provider if the victim publicly complain for the false searching result. However, the victim himself can not find who is the malicious party. To fix this drawback, each party has to broadcast all shares so as to enable every party to recover all search words and query the search engine with all these words. We also remark that there is a very simple method to achieve the same purpose of private shuffle.

2 Lindell-Waisbard private web search scheme

Assume that all parties hold a unique session identifier *sid* (e.g., this could be a timestamp). There is a group $\mathbb{G}$ of order q with generator g, to be used for the ElGamal encryption [6]. Let (E, D) denote a CCA2-secure public-key encryption scheme. At the beginning of the scheme, each party P_j has a search word w_j, $j = 1, \cdots, n$. At the end of the scheme, each P_j obtains an arbitrary search word $w'_j \in \{w_1, \cdots, w_n\}$.

We refer to [8] for the Initialization, Shuffle and Verification stages. Its Reveal and Query stages can be described as follows.

Reveal stage:

1. For every (u_i, v_i) in μ_n, P_j computes $s_i^j = u_i^{\alpha_j}$ and sends s_i^j to P_i.
2. Every party P_j computes $w'_j = \frac{v_j}{\prod_{k=1}^{n} s_j^k}$, thereby decrypting the ElGamal ciphertext and recovering the search word w'_j (here j denotes the current index in μ_n and not the index of the party who had input w_j at the beginning of the protocol).

Query stage: After the above shuffle, each party has someone's search word, and the parties then query the search engine with the word obtained. Finally, the parties all broadcast the result to all others.

3 An attack launched by any malicious party in Query stage

The Lindell-Waisbard private web search scheme is builded on the previous work [3]. They claim that the protocol is secure in the presence of malicious adversaries. We now remark that the scheme is vulnerable to an attack launched by any malicious party.

Suppose that P_k is a malicious party and the others are semi-honest. At the end of Reveal stage, P_k obtains a word w'_k which is in the set $\{w_1, \cdots, w_n\}$. In Query stage, P_k can query the search engine with an arbitrary word $\widehat{w_k}$ such that $\widehat{w_k} \neq w'_k$. He broadcasts the searching result corresponding to the word $\widehat{w_k}$. Since the probability that $\widehat{w_k} \in \{w_1, \cdots, w_n\}$ is negligible, the party corresponding to the word w'_k shall not obtain the proper searching result. More worse, if

the victim publicly complains for the false searching result, then P_k can link the true word w'_k to the victim. Note that the victim himself can not find who is the malicious party.

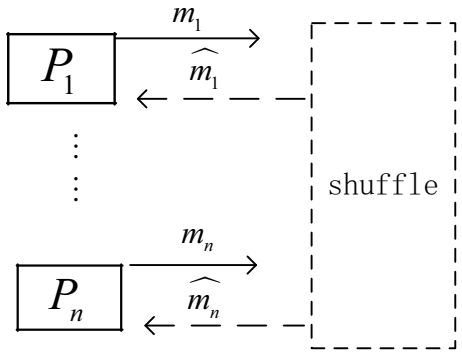
4 A modification of Lindell-Waisbard scheme

Note that the Lindell-Waisbard private web search scheme requires many broadcast channels. For example, each party P_j has to broadcast (h_j, pk_j) in Initialization stage, $(sid, P_j, \mathbf{true})$ or $(P_j, \mathbf{false})$ in Verification stage, and the searching result in Query stage. In view of that each party can access to these broadcast channels, in Reveal stage for every (u_i, v_i) in μ_n each party P_j can broadcast s_i^j to all others, instead of sending it to P_i in the mode of point-to-point. Hence, every party can recover all search words $w'_1, \cdots, w'_n$. Finally, every party can query the search engine with all these search words. See the following Table 1 for the differences between the original Lindell-Waisbard scheme and its modification.

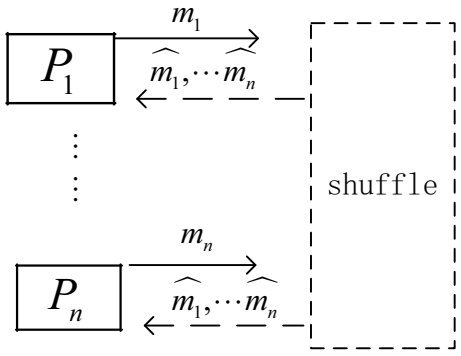
Table 1: Difference between the Lindell-Waisbard scheme and the modification

	The Lindell-Waisbard scheme	The modification
Reveal	For every (u_i, v_i) in μ_n, P_j computes $s_i^j = u_i^{\alpha_j}$ and sends s_i^j to P_i. Every party P_j computes w'_j.	For every (u_i, v_i) in μ_n, P_j computes $s_i^j = u_i^{\alpha_j}$ and broadcasts s_i^j and *the proof of* α_j to all others. Every party P_j checks the proofs and computes $w'_1, \cdots, w'_n$.
Query	Each party P_j queries the search engine with w'_j, and broadcasts the searching result.	Each party P_j queries the search engine with $w'_1, \cdots, w'_n$.

Graph 1: The Lindell-Waisbard shuffle



Graph 2: The modified Lindell-Waisbard shuffle



The modification requires that P_j broadcasts the zero-knowledge proof of α_j with respect to u_i. The requirement can not be removed. Otherwise, there exists a similar attack launched by

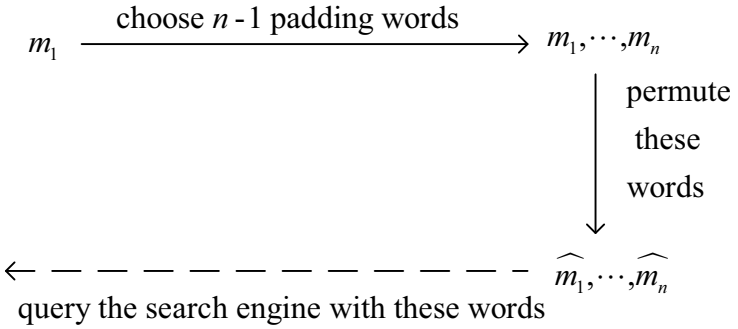
any malicious party. Suppose that P_j is the malicious party and the others are semi-honest. In Reveal stage, P_j broadcasts $\widehat{s_i^j}$ such that $\widehat{s_i^j} \neq u_i^{\alpha_j}$ for some index i. Hence, the others shall obtain $w'_1, \cdots, \widehat{w'_i}, \cdots, w'_n$. If the party corresponding to w'_i complains for the false word $\widehat{w'_i}$, P_j can link the true word w'_i to the victim. However, the victim himself can not find who is the malicious party.

We refer to the Graph 1 and Graph 2 for the essential differences between the original Lindell-Waisbard scheme and its modification.

5 A simple method for private web search

The essence of a private shuffle protocol is to mix a user's search word with another $n-1$ words such that an adversary can not know which is the user's true search word. In fact, there is a very simple method to achieve the same purpose. Concretely, when a user wants to privately query the search engine with a word, he first chooses $n-1$ padding words to form a group of n words and then permutes these words. Finally, he queries the search engine with these words.

Graph 3: A simple private web search method



It is easy to see that the simple scheme is secure because the adversary can know the true word with the probability of $1/n$. In comparison with the modified Lindell-Waisbard scheme, the simple method requires relatively little cost.

6 Further discussions

We have received some comments on the manuscript. Somebody argues that

> The attack proposed in this paper could be viewed as a type of denial of service where a malicious party always complains in the protocol, causing the whole session to abort.
>
> The last simple fix does not work because one will know all the words are from the same user, and as long as one of the words is sensitive, it is linked to that user.
>
> The correctness guarantee is not required for the Lindell-Waisbard scheme, and as

such malicious parties are allowed to perform denial of service type attacks (the attack mentioned above is one such attack).

We now want to point out that:

In the Lindell-Waisbard Scheme, it is very likely to happen that a malicious adversary changes the search word from others when submitting it to a search engine. This is because: 1) his malicious behavior cannot be detected by others so that he does not undertake any obligations; 2) the false searching result broadcasted could tempt the victim to complain.

The last simple fix is helpful to explain the essence of Lindell-Waisbard Scheme. From each user's point of view, the Lindell-Waisbard shuffle is just mixing his searching word with other $n-1$ words submitted by other $n-1$ users. We do not consider whether an adversary can find a "sensitive" word among these n words. Actually, it is difficult to define the term "sensitive" in the scenario.

The replacement attack cannot be falsely regarded as a type of "denial of service", because it takes place just at the end of the whole session. In such case, users can obtain correct searching results, except for the victim.

Their security model has actually considered replacement attacks but they did not pay attentions to the proposed malicious attack in the note. It points out in the introduction that [8]: "we still have to deal with 'replacement attacks' where the first party carrying out the mix replaces all of the encrypted search words with terms of its own, except for the one ciphertext belonging to the user under attack."

In short, this note is helpful to explain the gist of Lindell-Waisbard private shuffle and correct some misunderstandings about "denial of service" and "malicious attack in cryptography".

7 Conclusion

We show that there is a drawback in the Lindell-Waisbard private web search scheme. We also remark that there is a very simple method to achieve the same purpose of the Lindell-Waisbard scheme.

References

[1] D. Chaum. Untraceable Electronic Mail, Return Addresses, and Digital Pseudonyms. Communications of the ACM, 24(2), pp. 84-88, 1981.

[2] B. Chor, O. Goldreich, E. Kushilevitz and M. Sudan. Private Information Retrieval. Journal of the ACM, 45(6), pp. 965-981, 1998.

[3] J. Castellà-Roca, A. Viejo and J. Herrera-Joancomarti. Preserving User's Privacy in Web Search Engines. In: Computer Communications, 32(13-14), pp. 1541-1551, Elsevier, 2009.

[4] Y. Desmedt and K. Kurosawa. How to Break a Practical MIX and Design a New One. In: EUROCRYPT 2000, LNCS, vol. 1807, pp. 557-572, Springer-Verlag, 2000.

[5] R. Dingledine, N. Mathewson and P. Syverson. Tor: The Second-Generation Onion Router. In: Proceedings of the 13th USENIX Security Symposium, pp. 303-320, 2004.

[6] T. ElGamal. A Public Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms. In: CRYPTO'84, LNCS, vol. 196, pp. 10-18, Springer-Verlag, 1984.

[7] M. Jakobsson. A Practical MIX. In: EUROCRYPT'98, LNCS, vol. 1403, pp. 448-461, Springer-Verlag, 1998.

[8] Y. Lindell, E. Waisbard. Private Web Search with Malicious Adversaries. In: Proceedings of 10th International Symposium, PETS 2010, LNCS, vol. 6205. pp. 220-235. Springer-Verlag, 2010. The revised version is available at `http://u.cs.biu.ac.il/~lindell/abstracts/Private%20Search_abs.html`

[9] R. Ostrovsky and W. Skeith. A Survey of Single-Database PIR: Techniques and Applications. In: the 10th PKC, LNCS, vol. 4450, pp. 393-411, Springer-Verlag, 2007.