Protocols for Authenticated Oblivious Transfer

Mehrad Jaberi, Hamid Mala
Department of Computer Engineering
University of Isfahan
Isfahan, Iran
mehrad.jaberi@eng.ui.ac.ir, h.mala@eng.ui.ac.ir

Abstract Oblivious transfer (OT) is a basic building block in many cryptographic protocols. In this paper, we exploit some well-known authenticated Diffie-Hellman-based key exchange protocols to build three authenticated 1-out-of-2 oblivious transfers. We show that our proposed protocols are secure in the semi-honest model. We also compare our schemes with three similar 1-out-of-2 OT protocols and show that authentication in our schemes costs only up to either two more exponentiations or one message signing, compared to those with no authentication.

1 Introduction

Oblivious transfer (OT) is a basic cryptographic protocol and it is used as a core building block in secure multiparty computations (SMC). The simplest form of OT, which is called 1-out-of-2 OT, is a protocol in which the sender has two secrets m_0, m_1 and the receiver has a select bit sb. The sender has no output at the end of the protocol, while the receiver learns m_{sb}. Our proposed schemes are 1-out-of-2 OT protocols.

Considering the importance of OT and its key role in cryptographic applications, it is vital to introduce secure and efficient OT protocols. On the other hand, since OT is being used usually as a black-box, it is essential for the involved parties to be authenticated. In the current paper, we will introduce simple, secure and time efficient OT protocols. Despite previous key exchange based schemes, our OT protocols are authenticated as well. We exploit the most well-known Diffie-Hellman based authenticated key agreement schemes (KAS) [1,2,3] to construct new authenticated OTs.

Related work. Since 1981[4], where Rabin introduced the notion of OT (another similar concept had been proposed in 1970 under the name of “conjugate coding” [5]), there have been many papers proposing new OT protocols or trying to improve earlier ones [6,7,8,9]. The two notable protocols that are similar to ours, are [10] and [9], which are not as efficient as our schemes. Like our proposed protocols, [10] and [9] have been also constructed by exploiting Diffie-Hellman KAS. On the other hand, [11,12,13] tried to construct OT protocols as secure as possible. The recent effort has been made in [6], where Diffie-Hellman KAS [14] was used to construct an efficient OT. Note that the OT proposed in [6] is not authenticated, while our proposed protocols are authenticated using certifications signed by a trusted authority.

OT extension. Analogous to hybrid encryption systems, where two entities use public-key cryptography to share a symmetric-key and then use a symmetric encryption (e.g. AES) for further data communication, OT protocols can also be extended. In OT extension, entities generate few "seed" OTs based on public-key schemes, and then extend these base OTs to any number of OTs required, using symmetric-key schemes. [7] , [8] are two efficient examples for OT extension. Based on [6], we believe that our schemes can be very useful, efficient and simple OTs for being employed as seed OTs in OT extension.

Paper organization. The rest of this paper is organized as follows. In Section 2, we will propose our three authenticated OT schemes. In Section 3, we will discuss about the security of our proposed schemes. In Section 4, a comparison between our schemes and some other OT protocols will be presented. Finally, we conclude the paper in Section 5.

2 The proposed protocols

In this section, we propose three authenticated OT protocols. These protocols are based on three authenticated KAS. In fact we exploit Station-to-Station (STS) KAS [1], MTI KAS [2] and Girault KAS [3]. In our protocols, U and V are sender and receiver, respectively, where U owns two secrets m_0 and m_1. At the end of the protocol, V obtains either m_0 or m_1 while U learns nothing. U and V agree on $H(\cdot)$, a secure hash function, and a symmetric-key encryption algorithm such as AES-128.

In STS-based OT and MTI-based OT, suppose that p is a large prime number and all the operations are in $\mathbb{Z}_p$ and g is a generator of the multiplicative group $\mathbb{Z}_p^*$.

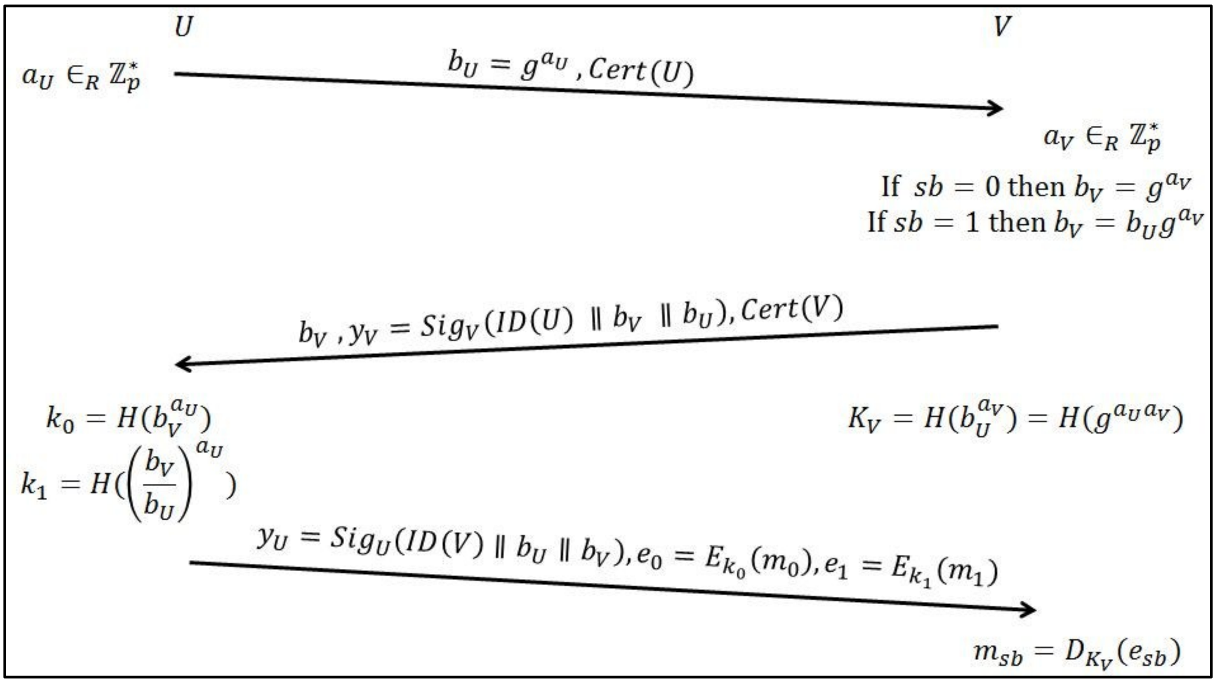
U
V
$a_U \in_R \mathbb{Z}_p^*$
$b_U = g^{a_U}, Cert(U)$
$a_V \in_R \mathbb{Z}_p^*$
If $sb = 0$ then $b_V = g^{a_V}$
If $sb = 1$ then $b_V = b_U g^{a_V}$
$b_V, y_V = Sig_V(ID(U) \parallel b_V \parallel b_U), Cert(V)$
$k_0 = H(b_V^{a_U})$
$k_1 = H(\left(\frac{b_V}{b_U}\right)^{a_U})$
$K_V = H(b_U^{a_V}) = H(g^{a_U a_V})$
$y_U = Sig_U(ID(V) \parallel b_U \parallel b_V), e_0 = E_{k_0}(m_0), e_1 = E_{k_1}(m_1)$
$m_{sb} = D_{K_V}(e_{sb})$

Fig. 1. The proposed STS-based OT

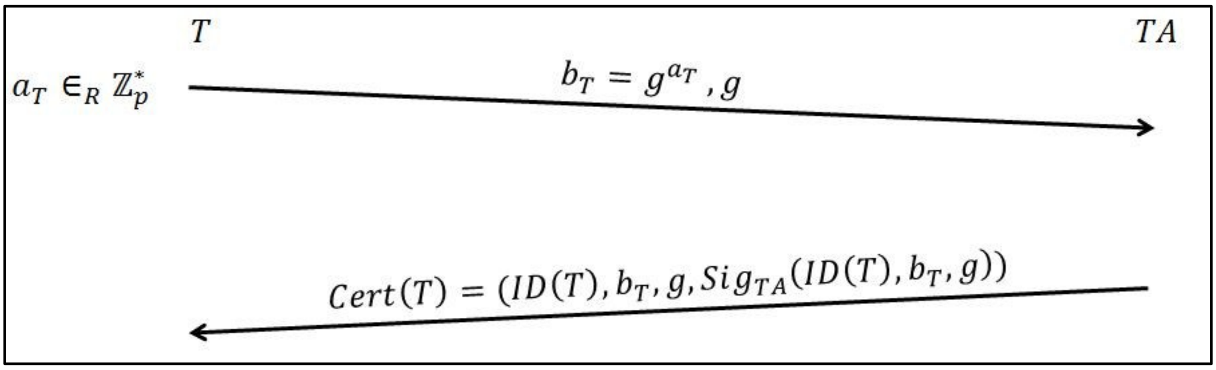
T
TA
$a_T \in_R \mathbb{Z}_p^*$
$b_T = g^{a_T}, g$
$Cert(T) = (ID(T), b_T, g, Sig_{TA}(ID(T), b_T, g))$

Fig. 2. The proposed MTI-based OT, public key generation phase

A. STS-based OT

Fig. 1, shows our STS-based oblivious transfer scheme. U chooses a_U, a random element of $\mathbb{Z}_p^*$ and sends $b_u = g^{a_u}$ along with her certificate $Cert(U)$ to V, where

$$Cert(U) = (ID(U), ver_U, Sig_{TA}(ID(U), ver_U))$$

ver_U is a verification algorithm for the signature scheme of U and Sig_{TA} is the signature of the TA which is verifiable for everyone. V chooses a_V at random from $\mathbb{Z}_p^*$. If his select bit $sb = 0$, then he computes $b_V = g^{a_V}$, otherwise he computes $b_V = b_U g^{a_V}$. Then he computes $K_V = H(b_U^{a_V}) = H(g^{a_U a_V})$ and $y_V = Sig_V(ID(U) \parallel b_V \parallel b_U)$. He sends b_V and y_V along with his certificate $Cert(V)$ to U. Then U verifies y_V using ver_V. If the signature y_V is not valid, she rejects. Otherwise she computes $k_0 = H(b_V^{a_U})$, $k_1 = H(\left(\frac{b_V}{b_U}\right)^{a_U})$ and $y_U = Sig_U(ID(V) \parallel b_U \parallel b_V)$. Then she encrypts m_0 and m_1 with k_0 and k_1, respectively and forms $e_0 = E_{k_0}(m_0)$ and $e_1 = E_{k_1}(m_1)$ where $E_\lambda(\rho)$ is the symmetric encryption of massage ρ with key λ. Now, U sends e_0 and e_1 along with y_U to V. V verifies y_U using ver_V.

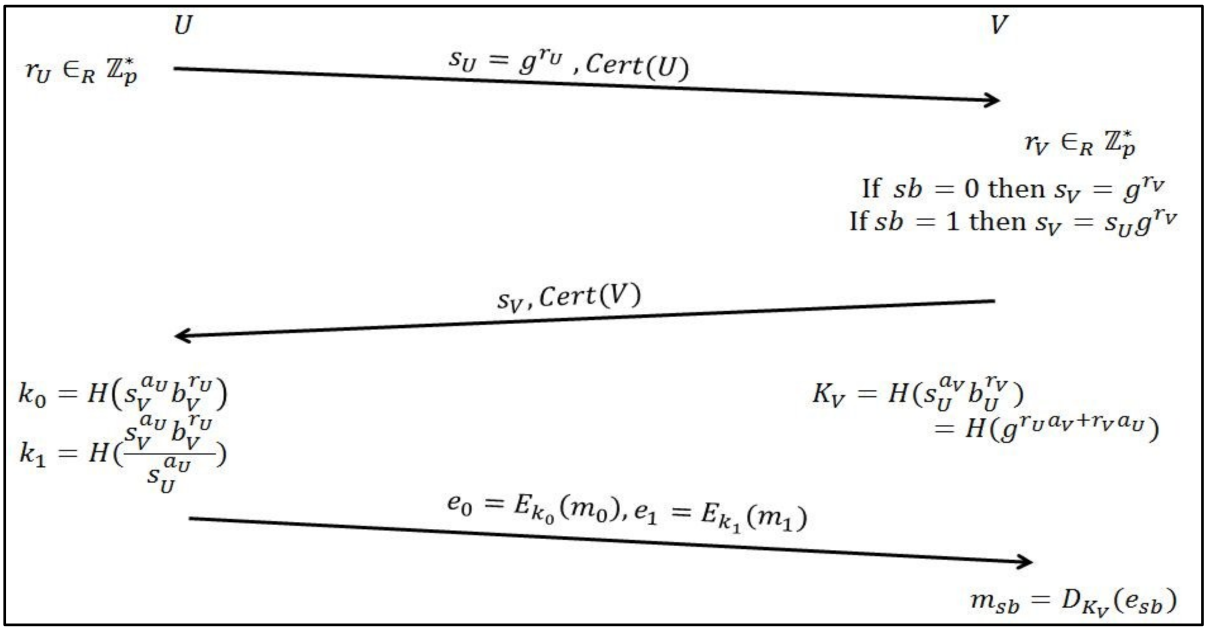
U
V
$r_U \in_R \mathbb{Z}_p^*$
$s_U = g^{r_U}, Cert(U)$
$r_V \in_R \mathbb{Z}_p^*$
If $sb = 0$ then $s_V = g^{r_V}$
If $sb = 1$ then $s_V = s_U g^{r_V}$
$s_V, Cert(V)$
$k_0 = H(s_V^{a_U} b_V^{r_U})$
$k_1 = H(\frac{s_V^{a_U} b_V^{r_U}}{s_U^{a_U}})$
$K_V = H(s_U^{a_V} b_U^{r_V})$
$= H(g^{r_U a_V + r_V a_U})$
$e_0 = E_{k_0}(m_0), e_1 = E_{k_1}(m_1)$
$m_{sb} = D_{K_V}(e_{sb})$

Fig. 3. Our main MTI-based OT

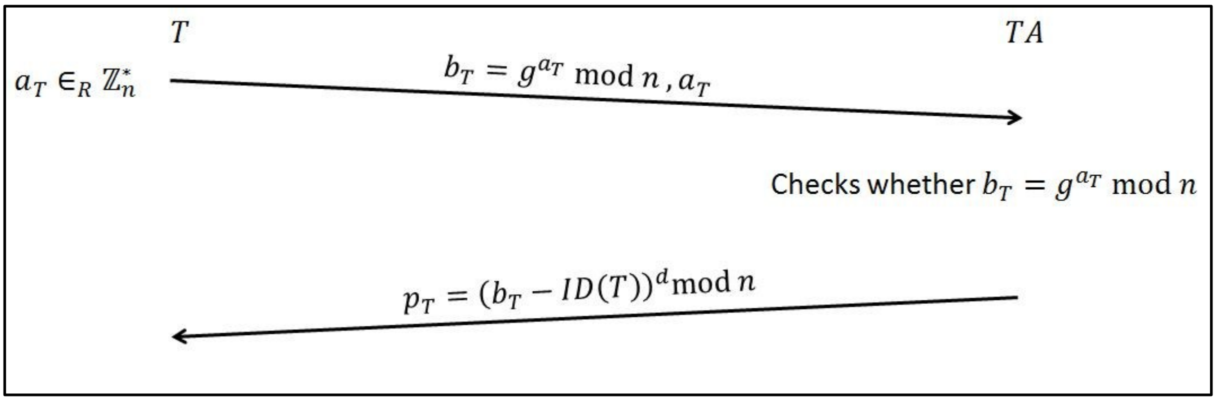
T
TA
$a_T \in_R \mathbb{Z}_n^*$
$b_T = g^{a_T} \bmod n, a_T$
Checks whether $b_T = g^{a_T} \bmod n$
$p_T = (b_T - ID(T))^d \bmod n$

Fig. 4. The Girault-based OT, public key generation phase

If the signature y_U is not valid he rejects; otherwise he decrypts $e_{sb}$with his key K_V. Note that he can decrypt both e_0 and e_1 but only one of them is meaningful. As it will be discussed in Section 3, the security of the scheme is based on intractability of the CDH problem.

B. MTI-based OT

The proposed MTI-based OT is shown in Fig. 2 and Fig. 3. Although MTI is a set of several key agreement schemes, we chose MTI/A0 which we believe is the most well-known one. Other MTI schemes will be exploitable to construct OT protocols using the same approach.

Public-key generation. First, each user T chooses a random element a_T from $\mathbb{Z}_p^*$ and computes $b_T = g^{a_T}$. Then T sends g and b_T to the TA. TA computes the user's certificate $Cert(T)$ from which b_T can be obtained and sends $Cert(T)$ to T. This phase can be operated offline.

The main MTI-based OT protocol. Since the MTI-based OT is very similar to the proposed STS-based OT, we abridge the explanation. Note that in the original MTI KAS, the mutual key of users U and V is computed as $K = g^{r_U a_V + r_V a_U}$, where r_T is a random element of $\mathbb{Z}_p^*$ chosen by user T in the beginning of the protocol. Hence in our MTI-based OT protocol $K_V = H(s_U^{a_V} b_U^{r_V}) = H(g^{r_U a_V + r_V a_U})$ and the keys generated by the sender are $k_0 = H(s_V^{a_U} b_V^{r_U})$ and $k_1 = H(\frac{s_V^{a_U} b_V^{r_U}}{s_U^{a_U}})$ where $s_T = g^{r_T}$. This protocol has been shown in Fig. 3.

C. Girault-based OT

Girault is a self-certifying KAS. We introduce our Girault-based OT protocol in two phases: "the public key generation" and "the main protocol". Girault combines features of both RSA and discrete logarithm problem. Suppose $n = pq$, where p and q are two large primes and g is a generator of the multiplicative group $\mathbb{Z}_n^*$. n and g are public but p and q are secret to the TA. On the other hand, TA chooses a public RSA exponent e and the corresponding secret exponent d where $d = e^{-1} \bmod \varphi(n)$.

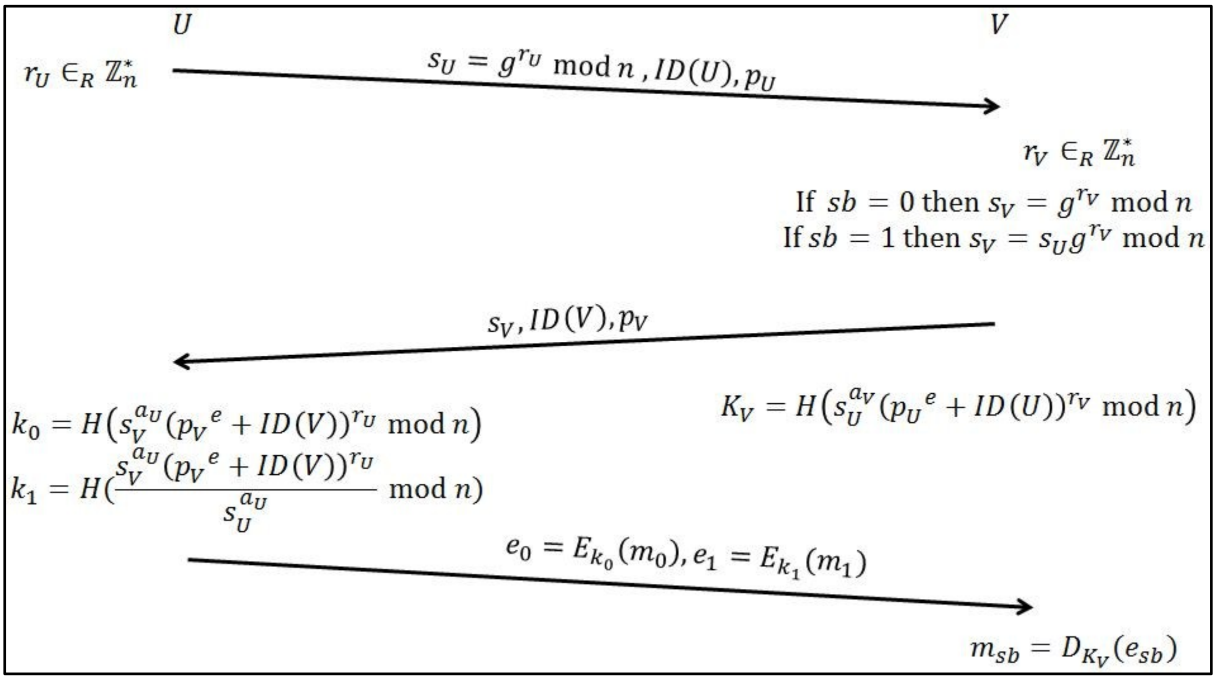
U
V
$r_U \in_R \mathbb{Z}_n^*$
$s_U = g^{r_U} \bmod n, ID(U), p_U$
$r_V \in_R \mathbb{Z}_n^*$
If $sb = 0$ then $s_V = g^{r_V} \bmod n$
If $sb = 1$ then $s_V = s_U g^{r_V} \bmod n$
$s_V, ID(V), p_V$
$k_0 = H\left(s_V^{a_U}({p_V}^e + ID(V))^{r_U} \bmod n\right)$
$K_V = H\left(s_U^{a_V}({p_U}^e + ID(U))^{r_V} \bmod n\right)$
$k_1 = H(\frac{s_V^{a_U}({p_V}^e + ID(V))^{r_U}}{s_U^{a_U}} \bmod n)$
$e_0 = E_{k_0}(m_0), e_1 = E_{k_1}(m_1)$
$m_{sb} = D_{K_V}(e_{sb})$

Fig. 5. Our main Girault-based OT

Public key generation. Each user T chooses a random number $a_T \in \mathbb{Z}_n^*$ and computes $b_T = g^{a_T} \bmod n$.Then T sends a_T and b_T to the TA through a secure channel. TA checks whether b_T is equal to $g^{a_T} \bmod n$ or not. If not, TA rejects; otherwise it computes $p_T = \left(b_T - ID(T)\right)^d \bmod n$ and sends p_T to T. This protocol has been shown in Fig. 4.

The main Girault-based OT protocol. U chooses r_U at random from $\mathbb{Z}_n^*$ and computes $s_U = g^{r_U} \bmod n$ and sends s_U along with the $ID(U)$ and p_U to V. Then, V, the receiver, chooses r_V at random from $\mathbb{Z}_n^*$. If his select bit sb is 0, then he computes $s_V = g^{r_V} \bmod n$. Otherwise he computes $s_V = s_U g^{r_V} \bmod n$ and $K_V = H(s_U^{a_V}\left(p_U^e + ID(U)\right)^{r_V} \bmod n)$. Then V sends s_V and $ID(V)$ and p_V to U. The sender U computes $k_0 = H(s_V^{a_U}\left(p_V^e + ID(V)\right)^{r_U} \bmod n)$ and $k_1 = H(\frac{s_V^{a_U}\left(p_V^e+ID(V)\right)^{r_U}}{s_U^{a_U}} \bmod n)$ and encrypts m_0 and m_1 by the keys k_0 and k_1, respectively. U sends $e_0 = E_{k_0}(m_0)$ and $e_1 = E_{k_1}(m_1)$ to V. Finally, V decrypts e_{sb} and obtains m_{sb}. Our proposed Girault-based OT has been demonstrated in Fig. 5.

3 Security of our proposed schemes

In this section, we discuss the security of our OT schemes in the semi-honest model. Hence, we explain that in our schemes, the sender U cannot guess the select bit of the receiver V with the probability more than $1/2$ and V can just decrypt one of the ciphertexts. In the following discussion, by $CDH(g^a, g^b, g)$ we denote the computational Diffie-Hellman problem. This problem states that "given g, the generator of a multiplicative group G, g^a and g^b, compute g^{ab}."

Table. 1 COMPARISON BETWEEN OUR OT SCHEMES AND THREE SIMILAR DIFFIE-HELLMAN-BASED OT PROTOCOLS

Protocol	Computational Complexity of Sender					Computational Complexity of Receiver					Authentication
	Exponentiation	Hash	Encryption	XOR	Singing	Exponentiation	Hash	Decryption	XOR	Singing	
STS-OT	3	2	1	0	1	2	1	1	0	1	✔
MTI-OT	4	2	2	0	0	3	1	1	0	0	✔
Grault-OT	5	2	2	0	0	4	1	1	0	0	✔
Chou [6]	3	2	2	0	0	2	1	1	0	0	✘
Bellare [10]	4	2	0	2	0	3	1	0	1	0	✘
Naor [9]	3	2	0	2	0	3	1	0	1	0	✘

Security of the STS-based OT. In the STS-based OT, since a_V is secret, U cannot distinguish between g^{av} and $b_U g^{av}$. In other words, when a_V is chosen uniformly at random from $\mathbb{Z}_p^*$, for any b_U in G the distribution of $g^{av}$and $b_U g^{av}$ are the same.

On the other hand, to learn both m_0 and m_1, V has to compute $b_U^{a_U}$. Thus, he needs to know a_U. Hence, V needs to solve the CDH problem $CDH(b_U, b_U, g)$.

Security of the MTI-based OT. In the MTI-based OT, each user T has two random elements a_T and r_T which are secret. Similar to the STS-based OT, since r_V is secret, g^{rv} and $s_U g^{rv}$ are indistinguishable for U.

Likewise, to decrypt both of e_0 and e_1, V has to compute $s_U^{a_U}$ (or $b_U^{r_U}$) where he needs either a_U or r_U. Thus, V should solve $CDH(s_U, b_U, g)$.

Security of the Girault-based OT. Same as the MTI-based OT, g^{rv} and $s_U g^{rv}$are indistinguishable for U, since r_V is a random secret.

On the other hand, to learn both secrets m_0 and m_1, V should learn either a_U or r_U. Thus, V should solve the CDH problem $CDH(s_U, b_U, g)$. Note that $b_U = p_U^e + ID(U) \bmod n$.

4 Comparison results

In this section, we compare our three proposed protocols with [6], [10], [9] in terms of computational complexity and authentication quality. These protocols are the most similar OT protocols to ours, since they have been also constructed by exploiting Diffie-Hellman KAS. As it is shown in Table. 1, none of the [6] , [10] and [9] support authentication while our proposed scheme does. The computational cost to achieve authentication is more exponential operations (up to two more exponential operation) in the Girault-based OT and MTI-based OT for both sender and receiver. To achieve authentication using STS-based OT, only a massage signing is needed for both parties.

5 Conclusion

In this paper, we introduced three authenticated oblivious transfer schemes by exploiting the most well-known Diffie-Hellman-based key exchange schemes namely, STS, MTI and Girault. Comparison among our proposed protocols and three other similar OT protocols shows that achieving authentication in Diffie-Hellman-based OT schemes using our method, costs up to either two exponentiations or one massage signing operation, for both sender and receiver. Note that for performance optimization, instead of intensive exponential operations, we can use elliptic curve computations. Our future work would be manipulating other key exchange schemes to gain more efficient OT protocols.

References

[1] W. Diffie, P. C. Van oorschot, and M. J. Wiener, "Authentication and authenticated key exchanges," *Designs, Codes and Cryptography*, vol. 2, no. 2, pp. 107-125, Jun. 1992.

[2] T. Matsumoto, Y. Takashima, and H. Imai, "On seeking smart public-key distribution systems," *The Transaction of the IECE of Japan*, vol. 69, pp. 99-106, 1986.

[3] M. Girault, "Self-certified public keys," in *Eurocrypt*, 1991, pp. 490-497.

[4] M. O. Rabin, "How to ecxhange secrets with oblivious trasnfer," Harvard University, 1981.

[5] S. Wiesner, "Conjugate coding," *SIGACT News*, vol. 15, no. 1, pp. 78-88, Jan. 1983.

[6] T. Chou and C. Orlandi, "The simplest protocol for oblivious transfer," in *4th International Conference on Cryptology and Information Security*, Guadalajara, Mexico, 2015, pp. 40-58.

[7] Y. Ishai, J. Kilian, K. Nissim, and E. Petrank, "Extending oblivious transfers efficiently," in *Advances in Cryptology- CRYPTO 2003*, 2003, pp. 145-161.

[8] V. Kolesnilov and R. Kumaresan, "Improved OT extension for transferring short secrets," in *CRYPTO 2013*, 2013, pp. 54-70.

[9] M. Naor and B. Pinkas, "Efficient oblivious transfer protocols," in *Proceedings of the Twelfth annual Symposium on Discrete Algorithms*, Washington DC, USA, 2001, pp. 448-457.

[10] M. Bellare and S. Micali, "Non-interactive oblivious transfer and applications," *Advances in Cryptology*, pp. 547-557, Aug. 1989.

[11] C. Hazay and Y. Lindell, "Efficient secure two-party protocols-techniques and constructions," *Information Security and Cryptography*, 2010.

[12] I. Damgard, B. Nielsen, and C. Orlandi, "Essentially optimal universally composable oblivious trasnfer," in *Information Security and Cryptology-ICISC*, Seoul, Korea, 2008, pp. 318-335.

[13] C. Peikert, V. Vaikuntanathan, and B. Waters, "A framework for efficient and composable oblivious transfer," *Advances in Cryptology*, pp. 554-571, 2008.

[14] W. Diffie and M. Hellman, "New directions in cryptography," *IEEE Transactions on Information Theory*, vol. 22, no. 6, pp. 644-654, Nov. 1976.