SCIENCE CHINA
Information Sciences

• LETTER •

September 2016, Vol. 59 099103:1–099103:3
doi: 10.1007/s11432-016-5550-8

New algorithms for verifiable outsourcing of bilinear pairings

Yanli REN[1*], Ning DING[2,3], Tianyin WANG[4], Haining LU[5] & Dawu GU[2]

[1]*School of Communication and Information Engineering, Shanghai University, Shanghai* 200444*, China;*
[2]*School of Electronic Information and Electrical Engineering, Shanghai Jiao Tong University, Shanghai* 200240*, China;*
[3]*State Key Laboratory of Cryptology, P. O. Box 5159, Beijing* 100878*, China;*
[4]*School of Mathematical Science, Luoyang Normal University, Luoyang* 471022*, China;*
[5]*School of Information Security and Engineering, Shanghai Jiao Tong University, Shanghai* 200240*, China*

Received June 5, 2015; accepted December 29, 2015; published online August 23, 2016

Citation Ren Y L, Ding N, Wang T Y, et al. New algorithms for verifiable outsourcing of bilinear pairings. Sci China Inf Sci, 2016, 59(9): 099103, doi: 10.1007/s11432-016-5550-8

Dear editor,

Outsourcing of computation has received widespread attention with the development of cloud computing and the proliferation of mobile devices [1]. Despite the huge benefits that it provides, it also encounters some security concerns and other challenges. First, the computational tasks often involve private information that should not be disclosed to the cloud servers since these cannot be fully trusted. Second, the cloud servers may return an invalid result, but the outsourcer may fail to detect the error [1].

Chevallier-Mames et al. [2] presented the first algorithm for secure outsourcing of bilinear pairings based on an untrusted server, in which the outsourcer could detect any failure with probability 1 if the server returned an incorrect result. However, with their algorithm, the outsourcer must execute some other expensive operations, including scalar multiplications and modular exponentiations, and these computations can involve difficulties comparable to those faced by bilinear pairings in some scenarios [3, 4]. Subsequently, other approaches to the delegation of bilinear pairings [5, 6] have suffered from similar problems. Chen et al. [7] proposed the first efficient outsourcing algorithm for bilinear pairing in the one-malicious version of the two-untrusted-program model, in which the outsourcer carries out only five point additions and four multiplications without any expensive operations. This algorithm is suitable for the computationally limited client, but its checkability is only 1/2, and the outsourcer may accept a faulty result returned by a malicious server with probability 1/2. Recently, Tian et al. [8] presented two outsourcing algorithms for bilinear pairing based on two servers. One of these is more efficient than the algorithm in [7], with the outsourcer needing to execute only four point additions and three multiplications with the same checkability. The other is more flexible and is based on two untrusted servers with an improved checkability $[1-1/(3s)]^2$, where s is a small integer. As we know, it is also possible for the outsourcer to be cheated by the server without the error being successfully detected.

Our Contributions. We first propose a secure verifiable outsourcing algorithm of single bilinear pairing based on two untrusted servers, which improves checkability for the outsourcer compared with the previous ones. We also present the first outsourcing algorithm for multiple bilinear pairings. In the proposed two algorithms, the out-

* Corresponding author (email: renyanli@shu.edu.cn)

The authors declare that they have no conflict of interest.

sourcer could detect any failure with probability 1 if one of the servers returns the fault result.

The proposed VBP algorithm. As in [1], a subroutine named *Rand* is used to speed up the computations. The inputs for *Rand* are a prime q, two cyclic additive groups G and $\hat{G}$ of order q, and a bilinear map $e : G \times \hat{G} \to G_T$, where G_T is a cyclic multiplicative group of order q, and the output is a random vector of the form $(a^{-1}, b^{-1}, aP, b\hat{P}, a_2P, b_2\hat{P}, e(aP, b\hat{P}), e(a_2P, b\hat{P})^{-1}, e(aP, b_2\hat{P})^{-1})$, where $a, b, a_2, b_2 \in Z_q$.

We propose a new outsourcing algorithm VBP for bilinear pairing, where T outsources its bilinear pairing computations to U_1 and U_2 by invoking the subroutine *Rand*. The input of VBP is $A \in G$, $B \in \hat{G}$, and the output is $e(A, B)$. Both A and B are computationally blinded to U_1 and U_2. The proposed VBP algorithm is described as follows.

(1) T runs *Rand* twice to create two vectors: $(a^{-1}, b^{-1}, aP, b\hat{P}, a_2P, b_2\hat{P}, e(aP, b\hat{P}), e(a_2P, b\hat{P})^{-1}, e(aP, b_2\hat{P})^{-1})$ and $(a_1^{-1}, b_1^{-1}, a_1P, b_1\hat{P}, a_3P, b_3\hat{P}, e(a_1P, b_1\hat{P}), e(a_3P, b_1\hat{P})^{-1}, e(a_1P, b_3\hat{P})^{-1})$, where $a, b, a_1, b_1, a_2, b_2, a_3, b_3 \in Z_q^*$.

(2) T queries U_1 in random order as follows:
$U_1(A - aP, B - b\hat{P}) \to \alpha_{11} = e(A - aP, B - b\hat{P})$,
$U_1(A - aP + a_3P, b_1\hat{P}) \to$
$\alpha_{12} = e(A - aP + a_3P, b_1\hat{P})$,
$U_1(a_1P, B - b\hat{P} + b_3\hat{P}) \to$
$\alpha_{13} = e(a_1P, B - b\hat{P} + b_3\hat{P})$.
Similarly, T queries U_2 in random order as follows:
$U_2(A - aP + a_1P, B - b\hat{P} + b_1\hat{P}) \to$
$\alpha_{21} = e(A - aP + a_1P, B - b\hat{P} + b_1\hat{P})$,
$U_2(A - aP + a_2P, b\hat{P}) \to$
$\alpha_{22} = e(A - aP + a_2P, b\hat{P})$,
$U_2(aP, B - b\hat{P} + b_2\hat{P}) \to$
$\alpha_{23} = e(aP, B - b\hat{P} + b_2\hat{P})$.

(3) T computes:
$\alpha_{12} \cdot e(a_3P, b_1\hat{P})^{-1} = e(A - aP, b_1\hat{P})$,
$\alpha_{13} \cdot e(a_1P, b_3\hat{P})^{-1} = e(a_1P, B - b\hat{P})$,
$\alpha_{22} \cdot e(a_2P, b\hat{P})^{-1} = e(A - aP, b\hat{P})$,
$\alpha_{23} \cdot e(aP, b_2\hat{P})^{-1} = e(aP, B - b\hat{P})$.
Next, T randomly chooses $t_1, t_2 \in Z_q^*$ and queries U_1 in random order as follows:
$U_1(e(A - aP, b\hat{P}), t_1b^{-1}) \to \alpha_{14} = e(A - aP, t_1\hat{P})$,
$U_1(e(aP, B - b\hat{P}), t_2a^{-1}) \to \alpha_{15} = e(t_2P, B - b\hat{P})$.
Similarly, T queries U_2 in random order as follows:
$U_2(e(A - aP, b_1\hat{P}), t_1b_1^{-1}) \to \alpha_{24} = e(A - aP, t_1\hat{P})$,
$U_2(e(a_1P, B - b\hat{P}), t_2a_1^{-1}) \to \alpha_{25} = e(t_2P, B - b\hat{P})$.

(4) Finally, T verifies that both U_1 and U_2 generate the correct outputs, that is, $\alpha_{14} = \alpha_{24}$, $\alpha_{15} = \alpha_{25}$, $\alpha_{21} = \alpha_{11} \cdot e(A - aP, b_1\hat{P})\, e(a_1P, B - b\hat{P})\, e(a_1P, b_1\hat{P})$. If not, T outputs "error"; else, T can compute $e(A, B) = \alpha_{11} \cdot e(A - aP, b\hat{P}) e(aP, B - b\hat{P}) e(aP, b\hat{P})$.

Comparison. We compare the outsourcing algorithms for single bilinear pairing with input privacy in Table 1, where s is a small integer and PA, SM, M, MInv, and MExp denote the computation of point addition and scalar multiplication in G or $\hat{G}$, multiplication, modular inverse, and modular exponentiation in G_T, respectively.

From Table 1, we conclude that the BJN algorithm [2] has no advantage for single bilinear pairing since the outsourcer has to execute 10 modular exponentiations in G_T and 6 point multiplications in G or $\hat{G}$, although it is based on a single untrusted server. Compared with the algorithms in [7] and [8], the proposed algorithm VBP has improved checkability and can efficiently verify the correctness of outsourcing results. Specifically, the checkability of our algorithm is 1, compared with 1/2 and $[1 - 1/(3s)]^2$ for the algorithms of [7] and [8], respectively, although at the penalty of a small increase in computational cost.

The proposed VMBP algorithm. We propose another outsourcing algorithm to compute multiple bilinear pairings $\prod_{i=1}^{n} e(A_i, B_i)$ $(n \geqslant 2)$, which is called VMBP algorithm. The input of VMBP is $A_1, \ldots, A_n \in G$, $B_1, \ldots, B_n \in \hat{G}$, and the output is $\prod_{i=1}^{n} e(A_i, B_i)$. All of A_i, B_i $(1 \leqslant i \leqslant n)$ are computationally blinded to U_1 and U_2. A subroutine named *Rand'* is used to speed up the computations. The input for *Rand'* is the same as *Rand*, and the output is a random vector of the form $(a^{-1}, b^{-1}, aP, b\hat{P}, a_2P, b_2\hat{P}, e(aP, b\hat{P})^n, e(a_2P, b\hat{P})^{-1}, e(aP, b_2\hat{P})^{-1})$, where $a, b, a_2, b_2 \in Z_q$. The proposed VMBP algorithm is described as follows:

(1) T firstly runs *Rand'* for two times to create two blinding vectors as below: $(a^{-1}, b^{-1}, aP, b\hat{P}, a_2P, b_2\hat{P}, e(aP, b\hat{P})^n, e(a_2P, b\hat{P})^{-1}, e(aP, b_2\hat{P})^{-1})$, and $(a_1^{-1}, b_1^{-1}, a_1P, b_1\hat{P}, a_3P, b_3\hat{P}, e(a_1P, b_1\hat{P})^n, e(a_3P, b_1\hat{P})^{-1}, e(a_1P, b_3\hat{P})^{-1})$, where $a, b, a_1, b_1, a_2, b_2, a_3, b_3 \in Z_q^*$.

(2) T queries U_1 in random order as follows:
$U_1(A_1 - aP, B_1 - b\hat{P}) \to$
$\alpha_{111} = e(A_1 - aP, B_1 - b\hat{P})$,
$\ldots$,
$U_1(A_n - aP, B_n - b\hat{P}) \to$
$\alpha_{11n} = e(A_n - aP, B_n - b\hat{P})$,
$U_1(\sum_{i=1}^{n}(A_i - aP) + a_3P, b_1\hat{P}) \to$
$\alpha_{12} = e(\sum_{i=1}^{n}(A_i - aP) + a_3P, b_1\hat{P})$,
$U_1(a_1P, \sum_{i=1}^{n}(B_i - b\hat{P}) + b_3\hat{P}) \to$
$\alpha_{13} = e(a_1P, \sum_{i=1}^{n}(B_i - b\hat{P}) + b_3\hat{P})$.
Similarly, T queries U_2 in random order as
$U_2(A_1 - aP + a_1P, B_1 - b\hat{P} + b_1\hat{P}) \to$
$\alpha_{211} = e(A_1 - aP + a_1P, B_1 - b\hat{P} + b_1\hat{P})$,
$\ldots$,
$U_2(A_n - aP + a_1P, B_n - b\hat{P} + b_1\hat{P}) \to$
$\alpha_{21n} = e(A_n - aP + a_1P, B_n - b\hat{P} + b_1\hat{P})$,

Table 1 Comparison of outsourcing algorithms for single bilinear pairing

Algorithm	BJN [2]	Pair [7]	TZR1 [8]	TZR2 [8]	VBP
PA(T)	4	5	4	$O(\log s)$	8
M(T)	6	4	3	$O(\log s)$	14
MExp(T)	10	0	0	0	0
SM(T)	6	0	0	0	0
Rand	0	8SM+2Pair	8SM+2MExp +2MInv+5M	8SM+2Exp +4MInv+8M	8SM+6Pair +8MInv
Pair(U)	4	8	6	6	6
MExp(U)	0	0	0	0	4
Number of servers	One	Two	Two	Two	Two
Checkability	1	1/2	1/2	$\left(1-\frac{1}{3s}\right)^2$	1
Communication Rounds	4	2	2	2	4

$U_2(\sum_{i=1}^n(A_i-aP)+a_2P,b\hat{P})\rightarrow$
$\alpha_{22}=e(\sum_{i=1}^n(A_i-aP)+a_2P,b\hat{P})$,
$U_2(aP,\sum_{i=1}^n(B_i-b\hat{P})+b_2\hat{P})\rightarrow$
$\alpha_{23}=e(aP,\sum_{i=1}^n(B_i-b\hat{P})+b_2\hat{P})$.

(3) After receiving $\alpha_{111},\ldots,\alpha_{11n},\alpha_{12},\alpha_{13}$ and $\alpha_{211},\ldots,\alpha_{21n},\alpha_{22},\alpha_{23}$ from U_1 and U_2 respectively, T computes:

$\alpha_{11}=\prod_{i=1}^n\alpha_{11i},\alpha_{21}=\prod_{i=1}^n\alpha_{21i}$,
$\alpha_{12}\cdot e(a_3P,b_1\hat{P})^{-1}=e(\sum_{i=1}^n(A_i-aP),b_1\hat{P})$,
$\alpha_{13}\cdot e(a_1P,b_3\hat{P})^{-1}=e(a_1P,\sum_{i=1}^n(B_i-b\hat{P})$,
$\alpha_{22}\cdot e(a_2P,b\hat{P})^{-1}=e(\sum_{i=1}^n(A_i-aP),b\hat{P})$,
$\alpha_{23}\cdot e(aP,b_2\hat{P})^{-1}=e(aP,\sum_{i=1}^n(B_i-b\hat{P}))$.

Next, T randomly chooses $t_1,t_2\in Z_q^*$ and queries U_1 in random order as

$U_1(e(\sum_{i=1}^n(A_i-aP),b\hat{P}),t_1b^{-1})\rightarrow$
$\alpha_{14}=e(\sum_{i=1}^n(A_i-aP),t_1\hat{P})$,
$U_1(e(aP,\sum_{i=1}^n(B_i-b\hat{P})),t_2a^{-1})\rightarrow$
$\alpha_{15}=e(t_2P,\sum_{i=1}^n(B_i-b\hat{P}))$.

Similarly, T queries U_2 in random order as

$U_2(e(\sum_{i=1}^n(A_i-aP),b_1\hat{P}),t_1b_1^{-1})\rightarrow$
$\alpha_{24}=e(\sum_{i=1}^n(A_i-aP),t_1\hat{P})$,
$U_2(e(a_1P,\sum_{i=1}^n(B_i-b\hat{P})),t_2a_1^{-1})\rightarrow$
$\alpha_{25}=e(t_2P,\sum_{i=1}^n(B_i-b\hat{P}))$.

(4) Finally, T verifies that both U_1 and U_2 generate the correct outputs, that is to say,

$\alpha_{14}=\alpha_{24},\alpha_{15}=\alpha_{25}$,
$\alpha_{21}=\alpha_{11}e(\sum_{i=1}^n(A_i-aP),b_1\hat{P})$
$e(a_1P,\sum_{i=1}^n(B_i-b\hat{P}))e(a_1P,b_1\hat{P})^n$.

If not, T outputs "error"; otherwise, T computes:

$\prod_{i=1}^n e(A_i,B_i)=\alpha_{11}e(\sum_{i=1}^n(A_i-aP),b\hat{P})$
$e(aP,\sum_{i=1}^n(B_i-b\hat{P}))e(aP,b\hat{P})^n$.

Conclusion. We propose two verifiable outsourcing algorithms for single bilinear pairing and multiple bilinear pairings. The security model is based on two non-colluding servers, and the outsourcer can detect any failure with probability 1 if one of the servers misbehaves. The proposed algorithms improve the checkability without decreasing efficiency for the outsourcer.

Acknowledgements This work was supported by National Natural Science Foundation of China (Grant Nos. 61202367, 61572309, 61572246), Doctoral Fund of Ministry of Education of China (Grant No. 20120073110094), Innovation Program of Shanghai Municipal Education Commission (Grant No. 14YZ020), and Plan for Scientific Innovation Talents, HASTIT of Henan Province (Grant No. 13HASTIT042).

References

1 Chen X F, Li J, Ma J F, et al. New algorithms for secure outsourcing of modular exponentiations. IEEE Trans Paral Distr Syst, 2014, 25: 2386–2396
2 Chevallier-Mames B, Coron J, McCullagh N, et al. Secure delegation of elliptic-curve pairing. In: Proceedings of 9th IFIP WG 8.8/11.2 International Conference on Smart Card Research and Advanced Application, Passau, 2010. 24–35
3 Scott M, Costigan N, Abdulwahab W. Implementing cryptographic pairings on smartcards. In: Proceedings of 8th International Workshop on Cryptographic Hardware and Embedded Systems, Yokohama, 2006. 134–147
4 Galbraith S, Paterson K, Smart N. Pairings for cryptographers. Discret Appl Math, 2008, 156: 3113–3121
5 Tsang P, Chow S, Smith S. Batch pairing delegation. In: Proceedings of 2nd International Workshop on Security, Nara, 2007. 74–90
6 Chow S, Au M, Susilo W. Server-aided signatures verification secure against collusion attack. In: Proceedings of 6th ACM Symposium on Information, Computer and Communications Security, Hong Kong, 2011. 401–405
7 Chen X F, Susilo W, Li J, et al. Efficient algorithms for secure outsourcing of bilinear pairings. Theor Comput Sci, 2015, 562: 112–121
8 Tian H B, Zhang F G, Ren K. Secure bilinear pairing outsourcing made more efficient and flexible. In: Proceedings of 10th ACM Symposium on Information, Computer and Communications Security, Singapore, 2015. 417–426