指數之私密加速運算於分散式伺服器

林志鴻
中興大學資訊科學與工程學系
s9856034@cs.nchu.edu.tw

陳昱圻
中興大學資訊科學與工程學系
phd9809@cs.nchu.edu.tw

陳國璋
中興大學資訊科學與工程學系
s9756013@cs.nchu.edu.tw

洪國寶
中興大學資訊科學與工程學系
gbhorng@cs.nchu.edu.tw

摘要

在許多常見的密碼系統或簽章中廣為運用指數運算。然而，指數運算所需的計算量比起其他運算相對的多。這使得如何加速指數運算成為重要且受注意的議題。

尤其是在行動通訊越來越流行的現在，使用者擁有的裝置計算能力可能不足或是受到電力限制難以有效率的進行複雜運算。由於輔助伺服器可能不可信任，則介於使用者與伺服器之間安全的加速計算協定是需要的。Van Dijk 等人於 2006 提出由一個不可信任伺服器輔助加速指數運算的協定。

本文我們延伸 Van Dijk 等人的架構提出建立於分散式伺服器上一個安全的指數運算加速協定。我們的協定可允許部份伺服器出錯。最後，此協定也類似 Van Dijk 等人的協定可應用在不同的情況並屬於無條件安全且比 Van Dijk 等人來的有效率。

關鍵詞：指數運算、伺服器輔助、分散式伺服器、行動通訊。

Abstract

Exponentiation is a commonly used operation in most of cryptosystems and signature schemes. How to speed up exponentiation becomes important. Because the mobile communication is more and more popular, the computational ability of the device owned by a user is weak or restricted, for example the power cannot provide complex computation efficiently. Van Dijk et al. in 2006 presented a speeding-up exponentiation protocol using an untrusted server to aid computing.

In this paper, we extend Van Dijk et al.'s construction, and propose a secure speeding-up exponentiation protocol with distributed servers. Taking the factor of fault tolerance into consideration, our protocol allows a number of dishonest servers. Our protocol is unconditional secure and more efficient than Van Dijk et al.'s.

Keywords: exponentiation, sever-aided, distributed servers, Mobile Communications.

1. 前言

在許多常見的密碼系統(如 RSA[17]及 ElGamal[7]等)或簽章法(ElGamal 簽章法[7]與 Schnorr 簽章法[18][20]等)廣泛使用指數運算[6]。然而，指數運算所需要得計算量比起其他運算(如加減乘除)相對的多，這使得如何加速指數運算成為一個重要的研究議題[2][4][10]。

在行動通訊越來越普及的時代，使用者擁有的裝置時常是具備可能不足的計算能力或是受到電力限制的行動裝置，以至於難以有效率的進行複雜運算。然而，強大的伺服器應該拿來輔助使用者來提昇效率。因此，伺服器輔助技術(Server-aided technique)的概念被提出並受到熱烈的探討[3][5][11][12][13][14][15]。但考慮安全的議題，如此強力的伺服器通常假設為不可信任的(untrusted)。

由於近數十年以來雲端計算(cloud computing)相關技術與研究廣受矚目使得利用不可信任伺服器加速運算的議題被廣為研究[10]。介於使用者與不可信任伺服器間之一個安全且可提昇計算的協定是絕對必要的。在 2006 年 Van Dijk 等人[21] 提出了一個藉由單一不可信任伺服器輔助運算來加速 Fixed Base-Variable Exponent 指數運算的協定(簡稱為 V-FBVE 協定)。他們利用此協定作為基礎並延伸至其他不同的類型如 Variable-Base 及 Variable-Exponent。他們宣稱此方法不需倚賴任何困難假設即可達到無條件安全，並且可以應用於許多基於離散對數[1][7][19][22]的密碼學機制上。然而，他們僅只使用單一不可信任伺服器。

本文我們延伸 Van Dijk 等人的架構來提出一個安全的指數運算加速協定建立於分散式

伺服器環境。考慮容錯情形時，我們的協定可以允許一定數量的伺服器出錯(或潛在惡意)而不影響計算結果。最後，由於我們的協定類似於 V-FBVE 協定故亦可應用在不同情況，以及他仍然屬於無條件安全且比 V-FBVE 協定來的稍有效率。

本文第二章描述基本環境架構及 Van Dijk 等人所提出協定；第三章我們提出安全的指數運算加速協定建立於分散式伺服器環境以及分析與討論，最後為本文結論。

2. 相關研究

2.1 環境架構

在 Van Dijk 等人的文章中[21]，此環境構分別由使用者與伺服器兩實體所組成。使用者定義為一個要解決某一問題所使用可信任但較弱的裝置。不幸的是使用者沒有足夠的能力解決問題故需要協助。伺服器為網路上提供服務的強力裝置可以協助使用者解決問題。基於安全問題，我們通常假設伺服器為不可信任故使用者必須確認由伺服器所提供的結果之完整性並完成問題。然而，此一驗證行為所需的計算成本理所當然的不能太高。

使用者開始解決問題前所進行的運算被定義為“offline work”。另一方面使用者開始解決問題所執行的運算則定義為“online work”。我們的環境架構類似於Van Dijk等人的架構。我們的環境架構使用n台分散式伺服器並將其定義為$S_1,\ldots,S_n$。

使用單一不可信任伺服器加速運算需符合以下需求：

1. Efficiency:使用者的線上計算成本低於親自計算整個問題的成本。
2. Completeness:若伺服器是誠實的，則使用者可獲得問題的正確解答。
3. Soundness:使用者面對任何伺服器皆可有充分的高機率獲得正確解答或是發現伺服器是不誠實的。

除此之外，我們希望對伺服器隱藏部份或整個問題。

有別於單一伺服器，我們使用分散式伺服器的架構故在考慮容錯情形下我們將第二及第三需求整合並將需求重新定義如下：

1. Efficiency: 使用者的線上計算成本低於親自計算整個問題的成本。
2. Fault-tolerance:若伺服器中超過一定門檻數為誠實的，則使用者可獲得問題的正確解答並有高機率可知道有伺服器不誠實。

2.2 Van Dijk 等人之指數運算加速協定

方便敘述起見，我們將本文會用到的符號列於此處。

符號：

G:一循環群。

g: G 的一個 base($g \in G$)

a: 指數

使用者所解問題→ g^a

n: G 的 order($g^n = 1$, for all $g \in G$)

s: 安全參數($s \leqq n$)

w: 衍生參數($n=wq$, $w<s$)

q: 衍生參數($n=wq$, $q \geqq s$)

現在簡單的描述 V-FBVE 協定[21]，此協定由三個階段所組成：使用者設定階段、伺服器輔助階段以及使用者驗證還原階段。

- 使用者建立階段：使用者輸入(g, a)而使用者準備計算他無法親自計算的g^a。首先計算$b=a \mod w$及$e=(a-b)/w$並隨機選取$m \leftarrow\{0,\ldots,s-1\}$及$r \leftarrow ZZ_n$。使用者將($g$, e, d)送給伺服器，此處$d=em+r$。
- 伺服器輔助階段：伺服器計算$x=g^e$, $y=g^r$將(x, y)傳回給使用者。
- 使用者驗證還原階段：使用者經由確認$x^m g^r=y$是否成立來驗證伺服器回傳的正確性，接受後可還原$g^a=x^w g^b$。

根據他們的分析，成功攻擊此協定的機率為$1/s = 2^{-\log s}$。當以乘法來估計計算成本時，使用者的計算成本為$\frac{3}{2}\lceil \log s \rceil + \lceil \log w \rceil + 2$的乘法，而伺服器為$2\lceil \log n \rceil$的乘法。我們在此省略完整的計算量估計過程。

3. 指數之運算加速於分散式伺服器

3.1 我們的方法

我們將介紹兩個安全的指數之運算加速協定於分散式伺服器，我們稱之為基本與進階深度法。

Ⅰ基本法

我們使用t台伺服器，依序稱為$S_1,\ldots,S_t$。如同Van Dijk等人的協定我們的協定也由三個階段所組成。

- 使用者建立階段：使用者輸入(g, a)而使用者準備計算他無法親自計算的g^a。首先計算$b=a \bmod w$及$e=(a-b)/w$並隨機選取亂數$r \leftarrow ZZ_n$。令$d=e+r$，傳送(g, d)到所有伺服器$S_1,\ldots,S_t$。
- 伺服器輔助階段：伺服器計算$R=g^d$，並將R傳回給使用者。
- 使用者驗證還原階段：使用者接收所有的R比較其值後選出現次數最多且大於t/2 設定為R'用於復原的計算。最後，還原$g^a=R'^w g^b/g^{rw}$。

Ⅱ進階深度法

由於基本法在"online work"上必須多計算g^{rw}增加了計算成本，為了克服此缺點我們提出了進階深度法。

我們設定有t台伺服器，依序稱為$S_1,\ldots,S_t$。如同基本法進階深度法也由三個階段所組成。

- 使用者建立階段：使用者輸入(g, a)而使用者準備計算他無法親自計算的g^a。首先計算$b=a \bmod w$及$e=(a-b)/w$並隨機選取t個截然不同的亂數$r_1,\ldots,r_t \leftarrow ZZ_n$。若$i=j$時令$d_j^i = e + r_i$否則令$d_j^i = -r_j$，$1 \leqq i,j \leqq t$；最後傳送$(g, d_j^i \mid 1 \leqq j \leqq t)$到$S_i$。
- 伺服器輔助階段：伺服器計算$x_j^i = g^{d_j^i}$，將x_j^i傳回給使用者。
- 使用者驗證還原階段：使用者接收$(x_j^i \mid 1 \leqq i,j \leqq t)$後計算$R_j^i = x_j^j x_j^i$，$j=\{1,\ldots,t\}, i=\{1,\ldots,t\}-\{i\}$，將$R_j^i$中出現次數最多且大於$t(t-1)/2$的值設定為$R$用於復原的計算。最後，還原$g^a=R^w g^b$。

3.2 協定之分析與討論

首先分析我們的協定之正確性，如下：

$$\begin{aligned} R^w g^b &= (R_j^j R_j^i)^w g^b \\ &= (x_j^j x_j^i)^w g^b \\ &= (g^{d_j^j} g^{d_j^i})^w g^b \\ &= (g^{e+r_j} g^{-r_j})^w g^b \\ &= (g^e)^w g^b \\ &= g^{ew+b} \\ &= g^a \end{aligned}$$

$R_j^j R_j^i$代表出現最多次的解中任意一組皆可滿足。

我們的協定在不考慮伺服器共謀的情況下與 V-FBVE 協定相似，我們的協定類似將 V-FBVE 協定延伸到多對多情況下使用固定 m 值；然而，在我們的協定中 e 值保持未知；因此，若他們的協定屬於無條件安全的則我們的協定理應也屬於無條件安全的。

若考慮伺服器共謀情形進行以下分析。

Theorem 1. 我們的協定可以允許一定數量的伺服器為不誠實的。

首先假設有 m 共謀欺騙者(或稱之不誠實伺服器)，而總伺服器數量為 n 台。若 m 的量不夠多則無法成功欺騙其餘的伺服器。當欺騙者進行欺騙時，他們僅能修改他們所收到的d_j^i。因此他們最多可以影響到 $m(n-1)+m(n-m)=m(2n-m-1)$ 組 $R_j^i = x_j^j x_j^i$ 的結果。由於我們是選取結果中相同的多數為最終結果，也就是只要欺騙用的結果不超過總結果的 1/2 即可。故設定我們的容錯條件為$n(n-1)/2-m(2n-m-1) \geqq 1$，在表 1 我們將列出符合此條件下能容錯數與所需伺服器的最低數量。然而，事實上一旦任意兩個欺騙者合作取得 e 則可以有效的完成欺騙。因此，進階深度法可以允許一定數量的伺服器為不誠實的。

表 1 總伺服器數與容錯數

總數	容錯數	總數	容錯數
5	1	39	11
8	2	43	12
11	3	46	13
15	4	49	14
18	5	52	15
22	6	56	16
26	7	60	17
29	8	63	18
32	9	67	19
36	10	70	20

接著，我們分析我們協定的計算量估計。由於此協定使用的底是固定的，故我們可以使用 squire-and-multiply 演算法進行預先計算g^{2^i}，$(1 \leq i \leq \lceil \log s \rceil)$。使用squire-and-multiply 演算法及 bit representation 計算 R^w 成本為$\frac{3}{2}\lceil \log w \rceil$，使用bit representation計算$g^b$的成本為$\frac{1}{2}\lceil \log w \rceil$。在使用$t$台伺服器數量的情況下

總和起來整體的”online work”成本為 $\frac{3}{2}\lceil \log w \rceil + \frac{1}{2}\lceil \log w \rceil + t(t\text{-}1)+1 = 2\lceil \log w \rceil + t(t\text{-}1)+1$。回顧Van Dijk等人的協定所需之成本為 $\frac{3}{2}\lceil \log s \rceil + \lceil \log w \rceil + 2$，由於 $w < s$ 故我們的協定比Van Dijk等人的協定更為有效率。因此我們滿足在 2.1 節所描述的需求。

最後，我們考慮實際應用的情形探討通訊成本，由於我們的”online work”通訊成本會受到伺服器增加而以$O(t^2)$方式成長。因此，如何降低通訊成本是我們未來所要努力的工作。

4. 結論

行動設備在近年來的使用比率已大幅提昇，雖然便於攜帶但計算能力未必足夠或者受到電力限制。作者提出使用分散式伺服器輔助計算加速 Fixed Base-Variable Exponent 指數運算的協定。此協定不僅可延伸應用於其他 Variable-Exponent 指數運算或 Variable-Base 指數運算之加速也滿足可允許一定數量不誠實的伺服器，並比 Van Dijk 等人的協定稍有效率。

致謝

本研究由國科會補助完成，計畫編號：NSC99-2221-E-005-078。

參考文獻

[1] L.M. Adleman and J. DeMarrais, ***“A subexponential algorithm for discrete logarithms over all finite fields,”*** Advances in Cryptology - Crypto’93 Proceedings, LNCS 773, pp. 147-158, 1994.

[2] E. F. Brickell, D. M. Gordon, K. S. McCurley, and D. B. Wilson. ***“Fast exponentiation with precomputation,”*** in “Advances in Cryptology—Proceedings of Eurocrypt ’92,” Vol. 658, pp. 200–207, Springer-Verlag, Berlin/New York, 1992.

[3] P. B´eguin and J-J. Quisquater, ***“Fast server-aided RSA signatures secure against active attacks,”*** Advances in Cryptology - Crypto’95 Proceedings, LNCS 963, pp. 57-69, 1995.

[4] E. Brickell, D. M. Gordon, K. S. McCurley and D. Wilson, ***“Fast exponentiation with precomputation,”*** Advances in Cryptology - Eurocrypt’92 Proceedings, LNCS 658, pp. 200-207, 1993.

[5] J. Burns and C. J. Mitchell, ***“Parameter selection for server-aided RSA computation schemes,”*** IEEE Transactions on Computers, Vol. 43, No. 2, pp. 163-174, 1994.

[6] D. Chaum, ***“Blind signatures for untraceable payments,”*** Advances in Cryptology - Crypto'82 Proceedings, pp. 199-203, 1982.

[7] T. ElGamal, ***“A public key cryptosystem and a signature scheme based on discrete logarithms,“*** IEEE Transactions on Information Theory, Vol. 31, No. 4, pp. 469-472, 2002.

[8] R. Gennaro, C. Gentry, and B. Parno. ***“Non-interactive veriable computing: Outsourcing computation to untrusted workers.”*** In CRYPTO'10, volume 6223 of LNCS, pages 465-482. Springer, 2010.

[9] D. M. Gordon, ***“A Survey of Fast Exponentiation Methods,”*** J. Algorithms, vol. 27, no. 1, pp. 129–146, 1998.

[10] S. Hohenberger and A. Lysyanskaya, ***“How to securely outsource cryptographic computations,”*** Theory of Cryptography, LNCS 3378, pp.264-282, 2005.

[11] G. Horng, ***“Secure server-aided RSA signature computation protocol for smart cards,”*** Journal of Information Science and Enginearing, Vol. 16, No. 6, pp. 847-855, 2000.

[12] S. Kawamura and A. Shimbo, ***“Fast server-aided secret computation protocols for modular exponentiation,”*** IEEE Journal on Selected Areas of Communications, Vol. 11, No. 5, pp. 778-784, 1993.

[13] C. H. Lim and P. J. Lee, ***“Server(prover/signer)-aided verification of identify proofs and signatures,”*** Advances in Cryptology - EuroCrypt’95 Proceedings, LNCS 921, pp. 64-78, 1995.

[14] C. H. Lim and P. J. Lee, ***“Security and performance of server-aided RSA computation protocols,”*** Advances in Cryptology - Crypto’95 Proceedings, LNCS 963, pp. 70-83, 1995.

[15] T. Matsumoto, K. Kato and H. Imai, ***“Speeding up secret computation with insecure auxiliary devices,”*** Advances in Cryptology - Crypto’88 Proceedings, LNCS 403, pp. 497-506, 1989.

[16] A. Odlyzko, ***“Discrete logarithms: The past and the future,”*** Designs, Codes and Cryptography, Vol. 19, No. 2-3, pp. 129-145, 2000.

[17] R. Rivest, A. Shamir, and L. Adleman, ***"A method for obtaining digital signatures and public-key cryptosystems,"*** Communications of the ACM, Vol. 21, No. 2, pp. 120-126, 1978.

[18] P. de Rooij, ***"On Schnorr's preprocessing for digital signature schemes,"*** Journal of Cryptology, Vol. 10, No. 1, pp. 1-16, 1997.

[19] O. Schirokauer, D. Weber and Th. F. Denny. ***"Discrete logarithms: the effectiveness of the index calculus method,"*** Proceedings Algorithmic Number Theory II, LNCS 1122, pp. 337-361, 1996.

[20] C. P. Schnorr, ***"Efficient signature generation by smart cards. Journal of Cryptology,"*** Vol. 4, ISSUE 3, pp. 161–174, 1991.

[21] M. Van Dijk, D. Clarke, B. Gassend, G.E. Suh and S. Devadas, ***"Speeding up Exponentiation using an Untrusted Computational Resource,"*** Designs, Codes and Cryptography, Vol. 39, No. 2, pp. 253-273, 2006.