以多項式為基礎的智慧卡認證協議之安全缺漏

謝文恭　涂承澔
中國文化大學資訊管理學系

摘要

智慧卡認證協議已被廣泛研究並運用到許多方面，在資訊通訊安全實務領域扮演十分重要的角色。為了提供更安全且有效率之遠端身份鑑別，Guo等人於2013年提出利用Chebyshev多項式之智慧卡認證協議，主張可防止內線攻擊、重送攻擊等，並主張其協議的計算成本與相關的協議相當。然而，本研究中，我們發現Guo等人所自訂之智慧卡認證協議，在認證過程所傳送的訊息中，並未提供完整之訊息認證碼，導致其使用者與伺服器雙方所建立的會期金鑰可能被竄改而無法發現。為確認其缺陷，我們透過演算法分析，成功竄改通訊雙方交換之訊息而不被發現，進而促使伺服器與使用者雙方的會期金鑰不一致，成功破壞其有效通訊。同時，因其登入訊息中具備與使用者相依之常數值，我們發現使用者之位置有被攻擊者追蹤之風險。另外，透過比較2013年類似之新認證協議，我們也發現Guo等人所訂之協議，可能耗費不必要的成本，有進一步降低成本的空間。

關鍵詞：Diffie-Hellman 金鑰交換、智慧卡、Chebyshev 多項式、金鑰協議

1. 前言

現代社會中，智慧卡泛用於很多方面，而我們所探討之 Guo 等人[1]的文章中也將其作為與伺服器之身分驗證使用。Guo 等人[1]將 Chebyshev 多項式以類似 Diffie-Hellman 金鑰交換演算法[2]方式建立認證協議，具備許多 Diffie-Hellman 金鑰交換演算法之優點。然而，我們發現 Guo 等人[1]所訂之協議有安全上的漏洞，包括使用者可能會被追蹤及智慧卡與伺服器雙方會期金鑰協議可能失敗等狀況。前者乃因登入訊息中包含使用者相關之常數值，後者則導因於缺乏完整之訊息認證碼。

Guo 等人[1]協議之安全漏洞，特別是會期金鑰協議失敗之關鍵，乃隱藏於忽視下列基本原則：區段加解密演算法只有保密效果，而無認證效果。特別當被加密之明文有許多區段時，確保某一段密文未被竄改，並不能確保其他區段密文不被竄改。

本文將檢視 Diffie-Hellman 金鑰交換演算法、Chebyshev 多項式及 Guo 等人所訂之認證協議，並探討 Guo 等人認證協議之缺失。同時，本文將指出如何竄改密文區段的攻擊方法。該攻擊將導致 Guo 等人認證協議中，智慧卡與伺服器雙方會期金鑰協議失敗之狀況。

2. Diffie-Hellman 金鑰交換演算法

Guo 等人[1]所自訂之協議中的演算法，其過程與 Diffie-Hellman 金鑰交換演算法[2]性質極其雷同，故在此略做說明 Diffie- Hellman 金鑰交換演算法，其中定義符號如下:

X_A: A 者私密金鑰
Y_A: A 者公開金鑰
X_B: B 者私密金鑰
Y_B: B 者公開金鑰
g、q: Diffie-Hellman 金鑰交換演算法之公開參數
SK: A、B 兩者共同金鑰

首先 A 與 B 兩人想建立秘密通訊，A 隨機選擇 X_A 做為私密金鑰，再計算公開金鑰 $Y_A = g^{X_A} \bmod q$；同樣的 B 也隨機選擇 X_B 為私密金鑰，再計算公開金鑰 $Y_B = g^{X_B} \bmod q$，然後互相交換彼此公開金鑰，當雙方接收到對方之公開金鑰後，A 以 B 的公開金鑰 Y_B 配合私密金鑰 X_A 計算 $SK =(Y_B)^{X_A} \bmod q$，B 也以 A 的公開金鑰 Y_A 配合私密

金鑰 X_B 計算 $SK=(Y_A)^{X_B} \bmod q$，即完成此次秘密通訊所需 Session Key SK 之交換如圖 1：

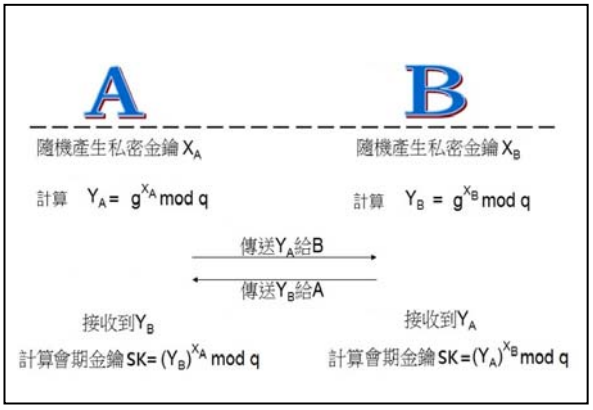

圖 1 Diffie-Hellman 金鑰交換過程

3.Chebyshev 多項式

Guo 等人[1]在文中利用了 enhanced chebyshev 多項式[3]，$T_n(x)=(2xT_{n-1}(x))(\mathrm{mod}N)$，其中 $n\geq 2$,x 屬於$(-\infty,+\infty)$，且 N 為極大的質數。請注意，根據 $T_n(x)$ 之定義將有 $T_r \cdot s(x)=T_r(T_s(x))=T_s(T_r(x))$。根據 Xiao 等人[4]，$T_n(x)$ 可發展類似 Diffie-Hellman 金鑰交換演算法如下:

1.A 與 B 選用亂數 x 屬於[-1,1]，且 x 不需要被保密。

2.A 選用一個極大正整數 r 計算 $X=T_r(x)$，然後傳送 X 給 B。

3.B 選用一個極大正整數 s 計算 $Y=T_s(x)$，然後傳送 Y 給 A。

4.A 可計算會期金鑰 $k=T_r(Y)=T_r(T_s(x))$，B 也可計算會期金鑰 $k=T_s(X)=T_s(T_r(x))$。

接著，A 與 B 可使用共同金鑰 k 於雙方的通訊中。

4. Guo 等人認證協議之過程

在此先列出相關符號定義如下:

U　:使用者

S　:伺服器

ID　:使用者身份

PW　:使用者密碼

T_1　:使用者之時間戳記

T_2　:伺服器之時間戳記

ΔT　:時間門檻

||　:符號串聯運算

h( )　:公開無碰撞單向雜湊函數

E_K( ) :以金鑰 K 安全對稱式加密演算

D_K( ) :以金鑰 K 安全對稱式解密演算

SK　:共同暫時金鑰

4.1 參數生成階段

1.伺服器 S 以 enhanced chebyshev 多項式選出公開金鑰$(x,T_s(x))$，其中 s 為私密金鑰。

2.伺服器 S 選用單向安全雜湊函數 h( )。

3.伺服器 S 選用對稱式加解密演算法，E_K( ) 為加密，D_K( )為解密，k 為對稱式金鑰。

4.2 註冊階段

使用者須到伺服器註冊，將會進行下列 4 項步驟如圖 2：

1.使用者 U 選用一個密碼 PW 與隨機亂數 b 來做 h(PW || b)之運算，在安全的通道上使用者 U 送出他的身分 ID 與 h(PW || b)值給伺服器 S 註冊。

2.假如 ID 為合法的，伺服器 S 運算 $IM=E_{KS}$(ID || h(PW || b))，其中 KS 為伺服器 S 的主金鑰。

3.伺服器 S 將資料{IM, h( ), E_k( ), x, $T_s(x)$}儲存進新的智慧卡裡。

4.使用者 U 將隨機亂數 b 儲存至智慧卡裡。

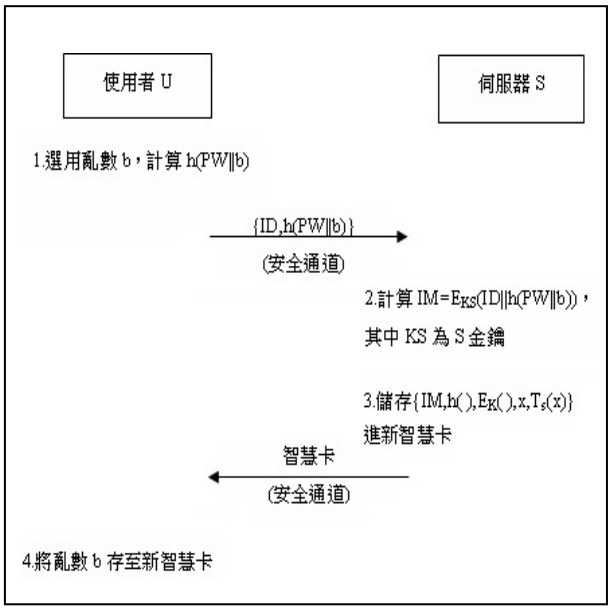

圖 2 註冊階段

4.3 認證階段

在上述註冊階段完成後，接著將進行認證階段。如圖 3，在認證階段使用者與伺服器將進行下列步驟：

圖 3 認證階段

1.使用者 U 將智慧卡插入讀卡機並輸入密碼 PW。

2.智慧片使用亂數 u 計算 $KA=T_uT_s(x)$ ，再計算 $E_{KA}(h_{PW}||IM||T_1)$，其中 $h_{PW}=(ID||h(PW||b))$ ，T_1 為目前系統時間，智慧卡傳送 $\{IM,T_u(x),E_{KA}(h_{PW}||IM||T_1)\}$給伺服器 S。

3.伺服器 S 收到$\{IM,T_u(x),E_{KA}(h_{PW}||IM||T_1)\}$，使用隨機產生之私密金鑰 s 計算 $KA=T_s(T_u(x))=T_uT_s(x)$，再進一步解密取得$(h_{PW}||IM||T_1)$ ，並檢驗是否 $T^*-T_1 \leqq \Delta T$。若不成立，則終止。

4.伺服器 S 使用主金鑰 KS 來解密 IM，取得 ID'與 h'(PW||b) ，計算 $h'_{PW}=h(ID'||h'(PW||b))$並比對是否 $h_{PW}=h'_{PW}$，如果不相等將停止，反之則認定使用者 U 為合法的。

5.伺服器 S 使用一個亂數 s'並計算 $T_{s'}(x)$，伺服器 S 計算 $E_{KA}(T_{s'}(x)||h(ID||T_2)||T_2)$，其中 T_2 為系統現在的時間，然後傳給智慧卡。

6.智慧卡收到後，使用 KA 來解密此訊息取得 T_2，然後比對是否在可接受的延遲時間內，進一步再計算 $h'(ID||T_2)$ 並比對是否 $h'(ID||T_2)=h(ID||T_2)$ ，如果此等式成立，則伺服器為真。

7.使用者 U 與伺服器 S 互相計算共同會期金鑰，各自的 $SK=T_uT_{s'}(x)=T_{s'}T_u(x)$ 。

5. Guo 等人認證協議之缺失

1.由認證階段的過程來觀察，可以發現 Guo 等人所訂協議中智慧卡與伺服器雙方分別進行 3 次 $T_n(x)$運算，應可降低。因為 Guo 等人所訂協議與 Diffie-Hellman 金鑰交換演算法過程性質雷同，且正常的 Diffie-Hellman 演算法中通訊雙方分別只使用 2 次的指數運算，故 Guo 等人所訂之協議可能耗費不必要的成本，有進一步降低成本的空間。2013 年，Shieh 等人[5]提出之智慧卡認證協議，利用 Diffie-Hellman 金鑰交換演算法同時達成交互認證與金鑰交換，演算法中通訊雙方分別只使用 2 次的指數運算。$T_n(x)$運算類似指數運算，相當耗費成本，如在營運上長期進行將會是筆可觀的支出。

2.使用者的所在位置可被追蹤。在認證階段的第 2 步驟裡，請注意智慧卡傳送的訊息內容為$\{IM,T_u(x),E_{KA}(h_{PW}||IM||T_1)\}$。我們可以看到智慧卡每次都傳遞固定之 IM 常數值，這很有可能會被有心人士追蹤其使用者的所在位置。請注意，由註冊階段步驟 2 可知，$IM=E_{KS}(ID \| h(PW \| b))$是利用使用者 ID 與 PW 計算所獲之資料。故該 IM 值為使用者相依資料。而且，因使用者 ID 每人不同，

每一使用者智慧卡有惟一之 IM 常數值。因此，即使不知使用者 ID 與 PW，利用每次傳遞之 IM 常數值，有心人士便可追蹤同一張智慧卡之位置，最終達到追蹤使用者的所在位置的目的。

3. 會期金鑰協議失敗。認證過程中的第 5 步驟，我們可以看到伺服器回傳一段訊息 $E_{KA}(T_{s'}(x)||h(ID||T_2)||T_2)$ 給智慧卡。這邊我們發現一個安全上的缺失，在回送這段訊息上並沒有提供完整之訊息認證碼，接收訊息之智慧卡於解密後，只驗證其中 $h'(ID||T_2)=h(ID||T_2)$ 是否成立。因 Guo 等人自訂之協議使用對稱式區段加解密，為方便說明，我們可假設使用的加解密演算法為 DES 演算法，且令 $A=T_{s'}(x)$，$B=h(ID||T_2)$，$C=T_2$。若 A 之長度 n 大於 DES 之區段長度 8，即 $|A| = n > 8$，我們可以設 $A=A_1||A_2$，其中 $|A_1| = 8$，$|A_2| = n-8$。請注意，我們故意讓 A_1 之長度正好等於 DES 之區段長度 8，代表被加密明文之第一區段。則根據區段加解密規則，我們將有 $E_{KA}(A||B||C) = E_{KA}(A_1||A_2||B||C) = E_{KA}(A_1) \parallel E_{KA}(A_2||B||C)$ 如下圖 4 所示：

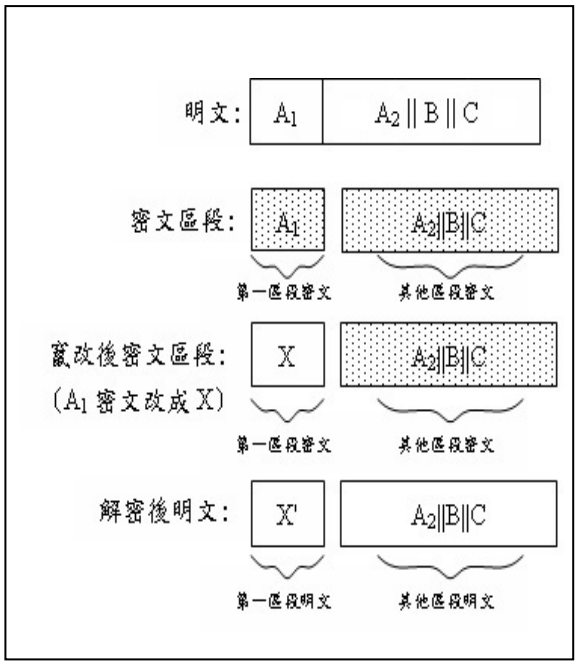

圖 4 駭客可能竄改的手段

請注意，圖 4 中 A=A1||A2 已被駭客竄改成為 X'||A2。如上述的作法，因解密後 B=h(ID||T2) 仍維持不變，竄改 A，即竄改 $T_{s'}(x)$資料，將不被智慧卡發現。最後，將導致使用者 U 與伺服器 S 各自計算之會期金鑰 SK 不相同，使會期金鑰協議失敗。

參考文獻

[1] C. Guo and C.-C. Chang, "Chaotic maps-based password-authenticated key agreement using smart cards," Communications in Nonlinear Science and Numerical Simulation, vol. 18, Issue 6, Jun. 2013, pp. 1433-1440.

[2] W. Diffie and ME. Hellman, "New direction in cryptography," IEEE Transaction on Information Theory, No. 22, Nov. 1976, pp. 644-654.

[3] L.-H. Zhang, "Cryptanalysis of the public key encryption based on multiple chaotic systems," Chaos Solutions & Fractals, vol. 37, No. 3, Aug. 2008, pp. 669-674.

[4] D. Xiao, X.-F. Liao, and S.-J. Deng, "A novel key agreement protocol based on chaotic maps," Information Science, vol. 177, No. 4, Feb. 2007, pp. 1136-1142.

[5] W.-G. Shieh and W.-B. Horng, "A security and efficiency improvement of Chung et al.'s remote authentication scheme for resource-limited environments," Journal of Convergence Information Technology, vol. 8, No. 2, Jan. 2013, pp. 795-803.