

# Kryptografické eskalační protokoly – část 1.

**Jan Krhovják**, Fakulta informatiky, MU, Brno  
([xkrhovj@fi.muni.cz](mailto:xkrhovj@fi.muni.cz))

V této části se seznámíme se základními vlastnostmi kryptografických eskalačních protokolů, s možnostmi jejich nasazení a s principy a problémy návrhu prvních eskalačních protokolů. V dalších částech se pak budeme věnovat jak jejich různým modifikacím a zlepšením, tak také dalším (zcela odlišným) typům protokolů. Cílem tohoto seriálu není podat vyčerpávající přehled ani přesný a úplný popis všech eskalačních protokolů, ale spíše informovat o některých poměrně zajímavých protokolech (a bezpečnostních problémech, které musely být při jejich návrhu zváženy a vyřešeny).

## Motivace

V dnešní době, kdy se počítače stávají běžnou součástí každodenního života a elektronické obchodování již není jen předmětem teoretických rozprav několika nadšenců v akademické sféře, je vzdálená autentizace uživatelů naprosto klíčovým faktorem mnoha používaných systémů a potřeba kvalitních a bezpečných autentizačních protokolů vzrůstá každým dnem stále více.

K velmi oblíbené metodě autentizace uživatelů patří použití nějaké tajné, krátké, a tudíž i snadno zapamatovatelné informace, jako je například PIN či heslo, jejíž znalostí prokazuje uživatel svou identitu. Naneštěstí klasické kryptografické protokoly navržené pro vzdálenou autentizaci vyžadují použití kvalitních a předem ustavených kryptografických klíčů, které bývají dlouhé řádově stovky bitů a pro člověka jsou zcela nezapamatovatelné.

Využití krátkých PINů či hesel v těchto protokolech bylo dlouhou dobu pokládáno za nemožné, protože jejich verifikace přes nezabezpečený kanál je vždy vystavovala riziku slovníkového útoku. Počátkem devadesátých let minulého století však bylo objeveno několik metod, jak se tohoto útoku vyvarovat. Vznikla nová třída autentizačních tzv. eskalačních protokolů, které byly založeny pouze na použití PINů či hesel a umožňovaly navíc i ustavení kvalitních kryptografických klíčů.

Protože po uživateli je celkově vyžadována jen znalost nějaké tajné informace, jsou tyto protokoly snazší a méně náročné při zavádění do praxe, než například metody založené na použití biometrik či tokenů. Tam kde již nějaká forma bezpečnostní infrastruktury existuje, může dodatečná implementace těchto protokolů sloužit jako další nezávislý autentizační mechanismus, který významně přispěje a posílí zabezpečení systému jakožto celku.

Praktická aplikace tohoto typu protokolů pokrývá veškeré případy komunikace přes nezabezpečený kanál, kde by dlouhotrvající uchovávání kryptografických klíčů bylo nebezpečné či nepraktické. Příkladem může být jejich nasazení v dnes poměrně rozšířeném autentizačním systému Kerberos, ale také použití jako náhrada zastaralých internetových protokolů umožňujících vzdálený login pomocí hesel zasílaných v otevřené podobě. Zajímavou aplikací může být i zabezpečení vzdáleného přístupu ke kryptografické čipové kartě podporující technologii Java Card™ či vytvoření efektivnějšího modelu budoucích sítí bankomatů, kde by již autentizace pomocí zákaznickova PINu neprobíhala přes několik mezilehlých síťových přepínačů sdílejících stejné předem ustavené tajné šifrovací klíče.

## Úvod a terminologie

Zajímavý způsob vytváření (a distribuce) kvalitních kryptografických klíčů nám nabízejí některé kryptografické protokoly. Jejich typickým příkladem je známá Diffie-Hellmanova (DH) metoda ustavení klíčů [DH76], kde sdílený klíč vzniká na základě náhodných dat vygenerovaných oběma stranami.

Chtějí-li si dvě strany<sup>1</sup> protokolu ustavit sdílený klíč sezení, domluví se nejprve nějakým způsobem na velkém prvočíselném modulu  $\beta$  a na generátoru  $\alpha$  multiplikatívni grupy  $Z_\beta^*$ . Obě tyto hodnoty jsou veřejné<sup>2</sup> a jako  $\beta$  je doporučeno volit *bezpečné prvočíslo* tvaru  $\beta = 2\gamma + 1$ , kde  $\gamma$  je velké prvočíslo [PH78, vOW96]. Poté si přes nezabezpečený kanál zašlou následující zprávy (někdy označované jako *DH hodnoty*):

$$\begin{array}{ccc} \boxed{A} & & \boxed{B} \\ \alpha^{r_A} \bmod \beta & \rightarrow & \\ & \leftarrow & \alpha^{r_B} \bmod \beta \\ & \dots & \end{array}$$

Čísla  $r_A$  i  $r_B$  jsou jednotlivými stranami generovány náhodně<sup>3</sup> z intervalu  $\langle 2, \beta - 1 \rangle$ . Sdílený klíč sezení  $K_S$  je pak každou stranou získán tak, že je přijatá DH hodnota umocněna na náhodně vygenerované číslo příslušné strany a  $K_S = \alpha^{r_A r_B} \bmod \beta$ .

Nedostatkem tohoto protokolu je zranitelnost útokem typu *man in the middle*. Tento problém řeší autentizované verze protokolu [DvOW92], které umožňují vzájemné ověření identity jednotlivých komunikujících stran (avšak vyžadují k tomu předem ustavené kryptografické klíče, certifikáty apod.).

My se v dalším zaměříme především na tzv. *eskalační protokoly*, které jsou založeny na použití dat s nízkou entropií (jako například hesel). Tyto kryptografické protokoly náleží k mimořádné třídě metod, které zajišťují autentizované ustavení klíčů přes nezabezpečený kanál a jsou založeny na použití hesel způsobem, který je nevystaví riziku off-line útoku hrubou silou (a tedy ani slovníkovému útoku). Jako eskalační je nazýváme proto, že málo kvalitní hesla eskalují na kvalitní kryptografické klíče.

Pokud neřekneme jinak, budeme předpokládat, že obě strany protokolu (tj. klient i server) mají předem ustaveno společné tajné heslo. Při vlastním popisu protokolů budeme také velmi často vynechávat zasílané identifikátory jednotlivých stran (ačkoliv jejich opomenutí by v reálných implementacích mohlo vést k nejrůznějším útokům). Šifrování (a dešifrování) budeme značit jako  $E_K$  (a  $D_K$ ), kde  $K$  je použitý klíč. Bude-li  $K$  tajný symetrický klíč, bude i šifrování/dešifrování prováděno nějakým symetrickým algoritmem. Bude-li  $K$  jeden z páru veřejný/soukromý klíč, bude i šifrování/dešifrování (resp. verifikace/podepisování) prováděno nějakým asymetrickým algoritmem.

---

<sup>1</sup> Většinou je budeme označovat pojmy *strana A* a *strana B*. Při použití terminologie klient/server, pak bude strana A chápána vždy jako klient a strana B jako server.

<sup>2</sup> Při popisu tohoto či jemu podobných protokolů budeme předpokládat, že již byly ustaveny.

<sup>3</sup> Předpokládáme tedy na obou stranách existenci kryptograficky bezpečného generátoru (pseudo)náhodných sekvencí/čísel.

## (DH)EKE

Historicky první protokol spadající do této kategorie je označován jako *EKE* (*encrypted key exchange*) [BM92] a jedná se o zcela originální kombinaci asymetrické a symetrické kryptografie. Základní část protokolu vypadá následovně:

$$\begin{array}{ccc} \boxed{A} & & \boxed{B} \\ E_P(VA) & \rightarrow & \\ & \leftarrow & E_P(E_{VA}(K_S)) \\ & \dots & \end{array}$$

Sdílený tajný klíč (v tomto případě tedy heslo  $P$ ) je použit k zašifrování veřejného klíče strany A (z jednorázově vygenerovaného páru soukromý/veřejný klíč), který je v zašifrované podobě doručen straně B, dešifrován a použit spolu s heslem k zašifrování touto stranou jednorázově vygenerovaného klíče sezení. Ten je pak odeslán zpět straně A a pomocí hesla a soukromého klíče dešifrován. V další části protokol pokračuje výměnou několika klíčem sezení zašifrovaných zpráv, které zajišťují ochranu proti útokům přehráním (např. pomocí náhodných čísel či časových razítek) a ověřují, zda ustavení klíče sezení proběhlo korektně.

EKE může být použit jak se systémy umožňujícími distribuci veřejného klíče (obzvláště dobře funguje s DH metodou ustavení klíčů [DH76]) tak s asymetrickými kryptosystémy (po vyřešení specifických problémů lze použít například RSA či ElGamal). Z bezpečnostního hlediska je u tohoto protokolu zcela zásadní, aby zpráva, která má být pomocí hesla zašifrována (např. výše zmíněný veřejný klíč), byla nerozlišitelná od náhodného čísla. V opačném případě (tj. kdyby zpráva měla určitou strukturu, kontrolní součet apod.) by bylo snadné provést off-line útok hrubou silou.

Při implementaci EKE pomocí RSA není například možné efektivně zakódovat veřejný klíč  $(n, e)$  tak, aby byl nerozlišitelný od náhodného čísla – útočník může vždy testovat, zdali má modulo  $n$  malé prvočíselné dělitele. Proto může být pomocí hesla zašifrován pouze exponent  $e$ , ke kterému je ještě před zašifrováním s pravděpodobností  $1/2$  přičtena hodnota  $1$  (protože všechny přípustné hodnoty  $e$  jsou liché). Implementace založená na kryptosystému ElGamal tímto nedostatkem netrpí, protože veřejné klíče jsou zde generovány jako  $\alpha^r \bmod \beta$  a mají tedy rovnoměrné rozložení na intervalu hodnot  $\langle 1, \beta - 1 \rangle$ .

Kdybychom u výše uvedeného příkladu s RSA ponechali před zašifrováním exponenty vždy liché, mohl by útočník při každém pokusu o dešifrování  $E_P(VA)$  pomocí zkoušených hesel  $P'$  zredukovat prostor možných hesel přibližně na polovinu. Prostě by jednoduše vyřadil všechna taková hesla  $P'$ , která by po provedení operace  $D_P(E_P(VA))$  vrátila jako výsledek sudé číslo. Takováto redukce prostoru hesel na polovinu zdánlivě nevypadá nijak nebezpečně. Uvážíme-li však, že při dalších sezeních jsou vždy vygenerovány nové hodnoty klíčů, lze tímto způsobem dále pokračovat v dělení/redukci prostoru hesel (celkově se tedy bude snižovat logaritmicky). Obecně se tento typ útoku, při němž dochází k postupnému dělení/redukci prostoru klíčů (hesel), nazývá *partition attack*.

Některé kryptosystémy jako například ElGamal mohou ale minimální dělení připustit – zde je pomocí hesla šifrováno nějaké celé číslo z intervalu  $\langle 0, \beta - 1 \rangle$ . Pokud jej zakódujeme do  $n$  bitů, tak budeme při zkoušení hesel schopni vyřadit všechna taková hesla  $P'$ , která po provedení operace  $D_P(E_P(VA))$  vrátí jako výsledek číslo z intervalu  $\langle \beta, 2^n - 1 \rangle$ . Zřejmě pokud

se  $\beta$  bude blížit  $2^n - 1$ , tak bude možno vyloučit jen malý počet hesel. Naopak, hodnoty  $\beta$  blízké  $2^{n-1}$  povedou k velké redukci prostoru hesel.

Dále je také potřeba zvážit doplnění vstupních dat na odpovídající délku bloku podporovanou příslušným symetrickým algoritmem. Doplnění nulami s sebou přináší opět riziko útoku, a proto by tyto doplňující bity měly být raději náhodné.

Poslední dva výše zmíněné problémy lze efektivně vyřešit následující úpravou. Předpokládejme, že šifrujeme nějaké číslo modulo  $\beta$  a délka šifrovacího bloku je  $m$  bitů, kde  $2^m > \beta$ . Dále necht'  $x = \lfloor 2^m / \beta \rfloor$  označuje číslo, kolikrát se interval  $\langle 0, \beta - 1 \rangle$  vejde do jednoho bloku. Pak vždy zvolíme náhodnou hodnotu  $j \in \langle 0, x - 1 \rangle$  a pomocí nemodulární aritmetiky přičteme k původnímu vstupnímu číslu hodnotu  $j\beta$ . Pokud je hodnota vstupního čísla menší než  $2^m - x\beta$ , použijeme  $j \in \langle 0, x \rangle$ .

Implementace základní části EKE pomocí DH metody ustavení klíčů se pak od obecného popisu protokolu mírně odklání – v prvních dvou krocích jsou přenášeny hodnoty chráněny heslem a na jejich základě si pak nezávisle strana A i B vygeneruje klíč sezení (např. výběrem určitých bitů z  $\alpha^{rArB} \bmod \beta$ ):

$$\begin{array}{ccc}
 \boxed{\text{A}} & & \boxed{\text{B}} \\
 E_P(\alpha^{rA} \bmod \beta) & \rightarrow & \\
 & \leftarrow & E_P(\alpha^{rB} \bmod \beta) \\
 & \dots &
 \end{array}$$

Útočník bez znalosti hesla nemůže provést útok *man in the middle* ani výpočet diskrétního logaritmu (v případě malých hodnot  $\beta$ ). Tuto implementaci EKE budeme většinou označovat jako DHEKE.

V [BM92] je dále diskutována volba vhodných (a)symetrických kryptosystémů, správných parametrů  $\alpha$ ,  $\beta$  a obrana proti kryptoanalytickým útokům.

## Reference

- [BM92] S. M. Bellare and M. Merritt. Encrypted Key Exchange: Password-based protocols secure against dictionary attacks. In *Proceedings IEEE Computer Society symposium on Research in Security and Privacy*, pages 72–84, May 1992.
- [DH76] W. Diffie and M. E. Hellman. New Directions in Cryptography. In *IEEE Transactions on Information Theory*, volume IT-22, pages 644–654, November 1976.
- [DvOW92] W. Diffie, P. C. van Oorschot, and M. J. Wiener. Authentication and Authenticated Key Exchange. In *Designs, Codes and Cryptography*, pages 107–125, 1992.
- [PH78] S. C. Pohlig and M. E. Hellman. An improved algorithm for computing logarithms over GF(p) and its cryptographic significance. In *IEEE*, volume IT-24, pages 106–110, 1978.
- [vOW96] P. C. van Oorschot and M. J. Wiener. On Diffie-Hellman Key Agreement with Short Exponents. In *Advances in Cryptology – Eurocrypt 96*, volume 1070 of *Lecture Notes in Computer Science*, pages 332–343. Springer, 1996.

## Kryptografické eskalační protokoly – část 2.

**Jan Krhovj**ák, Fakulta informatiky, MU, Brno  
([xkrhovj@fi.muni.cz](mailto:xkrhovj@fi.muni.cz))

V této části se zaměříme především na popis několika modifikací protokolu (DH)EKE, kterým jsme se zabývali v minulém čísle Crypto-Worldu (a jehož znalost předpokládáme). Dále se také seznámíme s dalším významným zástupcem kryptografických eskalačních protokolů (SPEKE) a s několika jeho modifikacemi.

### AEKE

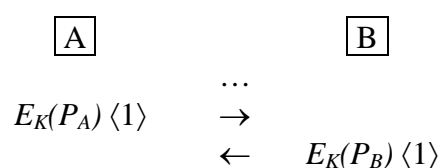
Nevýhodou protokolu EKE je, že jednotlivé strany si musí uchovávat svá sdílená hesla uložená v otevřené podobě. Protokol *AEKE* (*augmented encrypted key exchange*) [BM93] je takovým rozšířením a vylepšením protokolu DHEKE, které zajišťuje, že si server již uchovává hesla pouze jednocestně zašifrována – někdy je budeme označovat jako *verifikační hodnoty*. Útočník, který by získal přístup k souboru takto zašifrovaných hesel, by sice stále mohl vystupovat jako falešný server, ale nemohl by se jejich přímým použitím vydávat serveru za libovolného uživatele (nejprve by musel provést slovníkový útok).

Implementace AEKE pomocí schémat digitálních podpisů využívá páru soukromý/veřejný klíč, který je na základě hesla jednocestně vygenerován stranou A. Veřejný klíč je jakožto verifikační hodnota sdílen také se stranou B. Protokol začíná ustavením klíče sezení  $K_S$  metodou DHEKE, kde je k symetrickému šifrování namísto hesla použit výše zmíněný veřejný klíč. Strana A poté navíc podepíše  $K_S$  svým soukromým klíčem, podpis pomocí  $K_S$  symetricky zašifruje a zašle straně B. Ta podpis dešifruje a nakonec sdíleným veřejným klíčem ověří jeho korektnost.

### Interlock protocol

(A)EKE je vhodnou náhradou za Rivestův a Shamirův *interlock protocol* [RS84]. Ten je navržen tak, aby na komunikačním spoji detekoval aktivní útočníky. Davies a Price v [DP89] navrhli způsob jeho použití také k autentizaci, ale později byl na něj v [BM94] popsán útok. Z [BM94] v této části vycházíme.

Pro použití tohoto protokolu k autentizaci předpokládejme, že strany A a B sdílí dvě tajná hesla  $P_A$  a  $P_B$ . První fází protokolu je ustavení klíče  $K$  mezi těmito stranami, čehož je dosaženo standardní DH metodou. V druhé fázi protokolu pak pomocí tohoto klíče každá strana zašle své zašifrované heslo straně druhé, která jej porovná s uloženým heslem. Aby se zabránilo útoku *man in the middle*, rozdělila se zašifrovaná hesla na dvě části, které si měly jednotlivé strany střídavě vyměnit. Popis druhé fáze protokolu je uveden níže – v hranatých závorkách je vyznačeno číslo příslušné části zašifrovaného hesla:



$$E_K(P_A) \langle 2 \rangle \quad \rightarrow \\ \leftarrow \quad E_K(P_B) \langle 2 \rangle$$

Pokud by nějaký útočník vstoupil do procesu ustavení klíče<sup>1</sup> a následné autentizace, tak by nemohl dešifrovat první polovinu zprávy přijatou od strany A dokud nedorazí i její druhá polovina – dešifrováním jedné poloviny zprávy by nezískal správné heslo. To by mu mělo definitivně zabránit zfalšovat první zprávu určenou pro stranu B, čímž se mělo útoku *man in the middle* zabránit.

Předpokládejme však, že má útočník plnou kontrolu nad komunikačními spoji mezi A a B, a že se pokouší vydávat za stranu A. Nejprve si v první fázi protokolu se stranou A ustaví společný šifrovací klíč. V druhé fázi může straně A zaslat libovolné zašifrované heslo a počkat, až mu strana A pošle druhou polovinu svého zašifrovaného hesla. Po jeho dešifrování pak přeruší komunikaci s A (případně znemožní komunikaci mezi A a B) a sám začne komunikaci se stranou B. (Tomuto útoku nezabrání ani případná modifikace protokolu tak, že by druhou fázi dialogu začínala strana B.)

## MEKE

V [STW95] je popsána efektivnější varianta protokolu DHEKE, často označovaná jako *MEKE (minimal encrypted key exchange)*. Optimalizací došlo k redukci počtu zasílaných zpráv i prováděných kryptografických operací. Kromě popisu MEKE jsou diskutovány také kryptoanalytické útoky na (A)EKE a obrana proti nim.

Jako podstatné je u (A)EKE zdůrazněno především bezpečné ustavení klíče sezení  $K_S$  tak, aby jeho pozdější kompromitace neumožnila slovníkový útok na heslo. Proto by měl být klíč sezení raději vypočítán z původního  $K_S$  pomocí kryptografické hašovací funkce jako  $K_S = h(K_S)$ . Protokoly DHEKE i MEKE jsou proti podobnému typu útoku odolné.

## DWEKE

I přes svůj precizní návrh jsou mnohé používané protokoly stále náchylné k útokům, které umožňují se znalostí současně používaného hesla získat všechna v budoucnu ustavená hesla – tzv. *password chaining attacks*. Zásadní problém většiny těchto protokolů totiž je, že používají své heslo také k ochraně zpráv, které jsou použity k ustavení nového hesla. Útočník, který zná původní heslo, tak může s jeho pomocí snadno dešifrovat zprávu obsahující nové heslo. Výsledkem odhalení byť jen jediného hesla je pak kompromitace veškeré komunikace daného uživatele.

Původní návrh implementace EKE založené na DH metodě ustavení klíčů (tj. DHEKE) podporuje použití pevně dané hodnoty modulu  $\beta$ , která je buď z důvodu zvýšení rychlosti malá (několik set bitů) a nezamezuje těmto útokům, nebo je dostatečně velká (několik tisíc bitů) a za cenu snížení rychlosti jim zamezuje [BM92]. Protokol *DWEKE (dual-workfactor encrypted key exchange)* [Jas96] je vylepšenou variantou protokolu DHEKE a bez ztráty na rychlosti a efektivitě tomuto typu útoků zamezuje.

<sup>1</sup> Při použití samotné DH metody ustavení klíčů je útok *man in the middle* vždy možný.

Základní myšlenka DWEKE spočívá v použití silnější varianty DHEKE s dostatečně velkými hodnotami modulu  $\beta$  pro ustavení hesla, zatímco pro standardní autentizační zprávy se i nadále používá z hlediska výkonu efektivnější DHEKE s malými hodnotami  $\beta$ . Návrh a popis celého protokolu je velmi silně ovlivněn snahou o co nejsnazší začlenění do systému Kerberos.

## SPEKE

Protokol *SPEKE* (*simple password encrypted key exchange*) [Jab96] je svým návrhem a implementací velmi blízký protokolu DHEKE. I přes svou podobnost však mají tyto dva protokoly rozdílná omezení a nedostatky.

První fáze SPEKE je opět založena na DH metodě ustavení klíčů, ale namísto běžně používané fixní DH báze (generátoru)  $\alpha$  využívá SPEKE funkci  $f$ , která na základě svého jediného parametru (hesla) vytvoří nějakou bázi pro umocňování (tedy ne nutně generátor příslušné grupy). Výměna prvních dvou zpráv pak vypadá následovně:

$$\begin{array}{ccc} \boxed{\text{A}} & & \boxed{\text{B}} \\ f(P)^{r^A} \bmod \beta & \rightarrow & \\ & \leftarrow & f(P)^{r^B} \bmod \beta \\ & \dots & \end{array}$$

Z předaných hodnot si pak nezávisle strana A i B vygeneruje klíč sezení například jako  $K_S = f(P)^{r^A r^B} \bmod \beta$  či  $K_S = h(f(P)^{r^A r^B} \bmod \beta)$ . Funkci  $f$  je v případě použití bezpečného prvočísla  $\beta = 2\gamma + 1$  doporučeno definovat jako  $f(P) = P^{(\beta-1)/\gamma} \bmod \beta = P^2 \bmod \beta$  (tj.  $f(P)$  stejně jako 2 je řádu  $\gamma$ ), raději než  $f(P) = 2^P \bmod \beta$ . Protokol je tak odolnější, protože k případnému slovníkovému útoku již nestačí pouze jediný výpočet diskrétního logaritmu. Podrobnější informace vztahující se ke správné volbě  $f$  lze nalézt v [Jab96, Jab97].

V druhé fázi SPEKE pak obě strany opět ověřují, zdali ustavení klíče sezení proběhlo korektně. Kromě náhodných čísel lze k tomuto účelu použít také kryptografické hašovací funkce. Strana A nejprve zašle straně B zprávu  $h(h(K_S))$  a ta po jejím přijetí a ověření zašle straně A k ověření zprávu  $h(K_S)$ . Tento přístup je možný, protože klíč  $K_S$  vznikl z náhodných dat vygenerovaných oběma stranami a obsahuje také dostatek entropie.

SPEKE narozdíl od DHEKE v první fázi žádným způsobem nešifruje předávané zprávy, což útočníkovi dává možnost omezit prostor klíčů na malou množinu snadno předvídatelných hodnot – tzv. *subgroup confinement attack*. Podívejme se nyní na *man in the middle* verzi tohoto útoku popsanou v [vOW96]. Necht'  $\delta$  je známý malý prvočíselný dělitel  $\beta - 1$ , pak útočník umocní předávané hodnoty na  $(\beta - 1)/\delta$ . Tím se z nich stanou generátory malé podgrupy řádu  $\delta$  a útočník pak může klíč s pravděpodobností  $1/\delta$  uhádnout nebo hrubou silou snadno nalézt. Útok lze také modifikovat tak, že se útočník vydává za jednu ze stran protokolu a druhé straně pošle přímo generátor podgrupy malého řádu.

Použití bezpečných prvočísel počet malých podgrup pouze redukuje a jako protiopatření by tedy mělo být vždy testováno, zdali výsledný klíč do těchto podgrup nepatří (nebo na základě jejich prvků nevznikl).

## ASPEKE, BSPEKE a BEKE

Tato tři rozšíření protokolů jsou představena v [Jab97] a podobně jako AEKE zajišťují, že si server uchovává hesla pouze jednoduše zašifrována. ASPEKE je přímočará aplikace technik použitých k vytvoření AEKE na protokol SPEKE. BEKE a BSPEKE nahrazuje poslední část AEKE a ASPEKE dalším kolem DH metody ustavení klíčů, které umožňuje straně B ověřit, že strana A skutečně zná heslo.

Ověření znalosti hesla je u tohoto typu protokolů zcela nezbytné, protože jejich původní část zůstává až na použití jednoduše zašifrované verifikační hodnoty namísto otevřeného hesla naprosto beze změn (strana A si tuto verifikační hodnotu musí vždy z hesla dopočítat) a přímá znalost hesla tedy není prokázána – kdokoliv, kdo zná verifikační hodnotu, by mohl vystupovat za stranu A.

## Reference

- [BM92] S. M. Bellovin and M. Merritt. Encrypted Key Exchange: Password-based protocols secure against dictionary attacks. In *Proceedings IEEE Computer Society symposium on Research in Security and Privacy*, pages 72–84, May 1992.
- [BM93] S. M. Bellovin and M. Merritt. Augmented Encrypted Key Exchange: a Password-Based Protocol Secure Against Dictionary Attacks and Password File Compromise. In *Proceedings of the First ACM Conference on Computer and Communications Security*, pages 244–250, November 1993.
- [BM94] S. M. Bellovin and M. Merritt. An attack on the Interlock Protocol When Used for Authentication. In *IEEE Transactions on Information Theory*, volume 40, pages 273–275, January 1994.
- [DP89] D. W. Davies and W. L. Price. *Security for computer networks*. John Wiley & Sons, Inc., second edition, 1989.
- [Jab96] D. Jablon. Strong password-only authenticated key exchange. In *Computer Communication Review*, volume 26, pages 5–26. ACM SIGCOMM, October 1996.
- [Jab97] D. Jablon. Extended Password Key Exchange Protocols Immune to Dictionary Attacks. In *Proceedings of the Sixth Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises (WET-ICE '97)*, pages 248–255. IEEE Computer Society, June 1997.
- [Jas96] B. Jaspán. Dual-workfactor Encrypted Key Exchange: Efficiently Preventing Password Chaining and Dictionary Attacks. In *Proceedings of the sixth USENIX UNIX Security Symposium*, pages 43–50, July 1996.
- [RS84] R. L. Rivest and A. Shamir. How to Expose an Eavesdropper. In *Communications of the ACM*, volume 27, pages 393–395, 1984.
- [STW95] M. Steiner, G. Tsudik, and M. Waidner. Refinement and Extension of Encrypted Key Exchange. In *Operating Systems Review*, volume 29, pages 22–30. ACM SIGOPS, July 1995.
- [vOW96] P. C. van Oorschot and M. J. Wiener. On Diffie-Hellman Key Agreement with Short Exponents. In *Advances in Cryptology – Eurocrypt 96*, volume 1070 of *Lecture Notes in Computer Science*, pages 332–343. Springer, 1996.

## Kryptografické eskalační protokoly – část 3.

**Jan Krhovják**, Fakulta informatiky, MU, Brno  
([xkrhovj@fi.muni.cz](mailto:xkrhovj@fi.muni.cz))

V této závěrečné části si představíme poslední dva z námi vybraných eskalačních protokolů. Následovat bude krátké shrnutí, obsahující (mimo jiné) také stručný výčet několika dalších protokolů.

### SRP

*SRP (secure remote password)* [Wu97] je zcela nový typ protokolu, který (stejně jako některé z předchozích protokolů) zajišťuje, že si server uchovává hesla pouze jednocestně zašifrována. Narozdíl od protokolů jako AEKE a ASPEKE (které jsou založeny na použití digitálních podpisů) či BEKE a BSPEKE (které využívají přidané kolo DH metody ustavení klíčů) je SRP založen na obecné konstrukci zvané *AKE (asymmetric key exchange)*. Tato konstrukce oproti EKE žádným způsobem nevyužívá symetrickou kryptografii, což činí výsledné protokoly jednodušší a mnohdy i bezpečnější (není již třeba řešit žádné problémy spjaté s používáním hesel jakožto symetrických šifrovacích klíčů). Protokol SRP je speciální instancí AKE, a nabízí navíc vyšší výkon než srovnatelné protokoly jako například AEKE či BSPEKE.

Pro následující popis protokolu předpokládejme, že  $\alpha$  je generátor multiplikativní grupy  $Z_\beta^*$ , kde  $\beta$  je bezpečné prvočíslo, a že strana A na základě hesla  $P$  a soli<sup>1</sup>  $S$  vygeneruje veřejný klíč  $x = h(P, S)$  a verifikační hodnotu  $v = \alpha^x \bmod \beta$ . Hodnotu  $x$  pak smaže, zatímco hodnoty  $v, S$  doručí straně B (serveru).

V první části protokolu strana A zašle svůj identifikátor straně B, která jí nazpět zašle sůl  $S$  nezbytnou k výpočtu hodnoty  $x$ . Dále si strana A vygeneruje náhodné číslo  $a \in \langle 2, \beta - 1 \rangle$  a strana B náhodná čísla  $b, u \in \langle 2, \beta - 1 \rangle$ . Vlastní popis druhé části protokolu pak vypadá následovně:

$$\begin{array}{ccc} \boxed{\text{A}} & & \boxed{\text{B}} \\ C = \alpha^a \bmod \beta & \rightarrow & \\ & \leftarrow & u, D = v + \alpha^b \bmod \beta \\ & \dots & \end{array}$$

Strana A poté vypočítá společný klíč  $K_S = \alpha^{ab + bux} \bmod \beta$  jako  $(D - \alpha^x)^{a + ux} \bmod \beta$  a strana B jako  $(Cv^u)^b \bmod \beta$ . Klíč sezení je pak vypočítán jako  $K_S = h(K_S)$  a jeho znalost si v závěrečné části protokolu opět obě strany ověří.

Bezpečnost této metody stojí na důkazu z [Wu97], že existence techniky vedoucí v polynomiálním čase k získání klíče sezení použitého v SRP, by vedla také v polynomiálním

---

<sup>1</sup> Princip *solení* byl poprvé prezentován v [MT79] a aplikace této techniky v souvislosti s popisovanými protokoly je popsána například v [BM93, Jab97, Wu97]. Solení prakticky znemožňuje zjistit, zdali dvě (stejně) jednocestně zašifrovaná hesla vznikla ze stejného řetězce, a navíc útočnickovi brání vytvoření slovníku takto zašifrovaných hesel.

čase k rozbití DH metody ustavení klíčů. Z toho plyne, že SRP je alespoň tak bezpečný jako DH metoda ustavení klíčů.

Někdy je tento protokol nazýván SRP-3 a jeho vylepšená varianta SRP-6 [Wu02].

## PDM

Posledním protokolem, který si popíšeme, je *PDM (password derived moduli)* [PK01]. Tento protokol je opět založen na modifikaci DH metody ustavení klíčů, a jak již název napovídá, využívá heslo k vytvoření bezpečného prvočíselného modulu  $\beta$  (tj. modul musí být tvaru  $\beta = 2\gamma + 1$ , kde  $\gamma$  je velké prvočíslo).

Toho může být stranou A (klientem) dosaženo například využitím uživatelského hesla jako semínka generátoru pseudo-náhodných čísel, který bude k hledání odpovídající hodnoty  $\beta$  použit. Strana B (server) heslo nezná a má proto  $\beta$  jako verifikační hodnotu bezpečně uloženou. DH báze je u tohoto protokolu vždy  $\alpha = 2$ . Aby se při oboustranné autentizaci předešlo náročnému umocňování, tak má server navíc uloženy předpočítané hodnoty  $rB$  a  $2^{rB} \bmod \beta$ . Celý protokol pak vypadá následovně:

$$\begin{array}{ccc}
 \boxed{\text{A}} & & \boxed{\text{B}} \\
 2^{rA} \bmod \beta & \rightarrow & \\
 & \leftarrow & 2^{rB} \bmod \beta, R, h(2^{rArB} \bmod \beta) \\
 h(R, 2^{rArB} \bmod \beta) & \rightarrow & 
 \end{array}$$

$R$  je náhodně vygenerovaná hodnota. Při použití PDM výhradně k autentizaci mohou skutečně hodnoty  $rB$  a  $2^{rB} \bmod \beta$  zůstat pro daného klienta stejné. Pokud by ale cílem bylo i ustavení klíče sezení (založeného na hodnotě  $2^{rArB} \bmod \beta$ ), bylo by nezbytné náhodné číslo  $rB$  pro každý běh protokolu generovat znovu (s čímž by samozřejmě souvisel i opětovný výpočet  $2^{rB} \bmod \beta$ ).

Aby se předešlo redukci prostoru hesel, nesmí být žádná přenášená DH hodnota<sup>1</sup> větší než jakákoliv  $\beta'$  (vytvořené na základě zkoušených hesel  $P'$ ). Tento problém lze vyřešit zamítnutím použití takových náhodných čísel  $rA$  a  $rB$ , pro něž  $2^{rA} \bmod \beta$  a  $2^{rB} \bmod \beta$  jsou čísla větší, než nejmenší možná hodnota  $\beta$  vygenerovaná z nějakého hesla. Aby pravděpodobnost zamítnutí výše uvedených náhodných čísel byla co nejmenší (a předešlo se tak co nejvíce jejich opětovnému generování), bude se  $\beta$  volit z velmi úzkého intervalu prvočísel – v našem případě velmi blízkého mocnině dvou. Použijeme-li například 700bitová čísla, můžeme vhodný úzký interval získat fixním nastavením horních 64 bitů na binární hodnotu 1. Zbýlý prostor  $2^{636}$  čísel je pro vyhledávání bezpečných prvočísel stále dostatečně velký, a pravděpodobnost, že číslo  $2^{rA} \bmod \beta$  či  $2^{rB} \bmod \beta$  bude větší než nejmenší možná hodnota  $\beta$  je  $1/2^{64}$ .

V [PK99, PK01] je také uvedeno několik modifikovaných protokolů, které jsou určeny pro bezpečné stahování citlivých informací (například soukromých klíčů).

<sup>1</sup> Jako DH hodnoty v případě  $\alpha = 2$  označujeme  $2^{rA} \bmod \beta$  a  $2^{rB} \bmod \beta$ .

## Shrnutí

V úvodu tohoto seriálu jsme se seznámili s protokolem EKE [BM92], který k ustavení klíče sezení využívá sdíleného hesla v kombinaci se symetrickou i asymetrickou kryptografií a poskytuje ochranu proti off-line útokům hrubou silou. Myšlenka tohoto zcela originálního protokolu se stala základem celé třídy nově vznikajících protokolů.

Oproti původnímu EKE zaručují protokoly DHEKE [BM92], DWEKE [Jas96], MEKE [STW95] či SPEKE [Jab96] navíc *dopřednou bezpečnost* (forward secrecy), což znamená, že kompromitace hesla neumožní útočníkovi získat klíče předcházejících sezení. Navíc začíná být také brán zřetel na to, aby případná kompromitace klíče sezení neumožňovala útoky vedoucí k získání hesla. Největší nevýhodou všech výše uvedených protokolů však stále zůstává nutnost uchovávat hesla na straně serveru v otevřené podobě. Tento nedostatek jako první překonává protokol AEKE [BM93], který je rozšířením EKE a umožňuje serveru ukládat hesla jednocestně zašifrována. Nevýhodou této modifikace EKE je, že protokol už nezaručuje dopřednou bezpečnost. Protokol BSPEKE [Jab97] již podobnými nedostatky za cenu podstatného zvýšení výpočetní složitosti netrpí. Efektivnější řešení pak nabízejí protokoly SRP [Wu97] a PDM [PK01].

Existuje samozřejmě mnohem větší množství protokolů založených na použití hesla. Z nejvýznamnějších zmiňme například protokoly jako AMP [Kwo00], AuthA [BR00], OKE [Luc97], PAK [Mac02], S3P [RCW98] či SNAPi [MSP00]. Velkou nevýhodou mnoha z těchto (a jim podobných) protokolů je, že nejsou prezentovány společně s důkazy, které by prokázaly jejich bezpečnost. Na několik z nich již byly objeveny útoky [Pat97]. Některé z protokolů jsou navíc také patentovány.

Standardizací protokolů založených na použití hesla se zabývá pracovní skupina IEEE P1363 – viz draft IEEE P1363.2 [IEEE05]. Cílem tohoto právě vznikajícího dokumentu však není upřednostnění některých technik či protokolů před jinými, ale poskytnutí dostatečného množství různých metod, které se liší jak funkčností, tak také efektivitou. Podrobný popis jednotlivých metod pak slouží jakožto návod k jejich implementaci (a to jak na straně serveru, tak také na straně klienta). Celkově se tento draft skládá z hlavní části (obsahující popis jednotlivých metod), příloh (poskytujících doprovodné informace) a náčrtů jednotlivých postupů. Bohužel, jako mnoho jiných, je i tento draft prozatím značně nepřehledný a bez důkladné znalosti jednotlivých technik a protokolů téměř nečitelný.

## Reference

- [BM92] S. M. Bellare and M. Merritt. Encrypted Key Exchange: Password-based protocols secure against dictionary attacks. In *Proceedings IEEE Computer Society symposium on Research in Security and Privacy*, pages 72–84, May 1992.
- [BM93] S. M. Bellare and M. Merritt. Augmented Encrypted Key Exchange: a Password-Based Protocol Secure Against Dictionary Attacks and Password File Compromise. In *Proceedings of the First ACM Conference on Computer and Communications Security*, pages 244–250, November 1993.
- [BR00] M. Bellare and P. Rogaway. The AuthA Protocol for Password-based Authenticated Key Exchange. In *Contribution to the IEEE P1363 study group*, February 2000.

- [IEEE05] IEEE. P1363.2/D20 (Draft version 20) – Standard Specifications for Password-based Public Key Cryptographic Techniques, 2005.
- [Jab96] D. Jablon. Strong password-only authenticated key exchange. In *Computer Communication Review*, volume 26, pages 5–26. ACM SIGCOMM, October 1996.
- [Jab97] D. Jablon. Extended Password Key Exchange Protocols Immune to Dictionary Attacks. In *Proceedings of the Sixth Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises (WET-ICE '97)*, pages 248–255. IEEE Computer Society, June 1997.
- [Jas96] B. Jaspán. Dual-workfactor Encrypted Key Exchange: Efficiently Preventing Password Chaining and Dictionary Attacks. In *Proceedings of the sixth USENIX UNIX Security Symposium*, pages 43–50, July 1996.
- [Kwo00] T. Kwon. Ultimate Solution to Authentication via Memorable Password. In *Contribution to the IEEE P1363 study group for Future PKC Standards*, 2000.
- [Luc97] S. Lucks. Open key exchange: How to defeat dictionary attacks without encrypting public keys. In *Proceedings of the 5th International Workshop on Security Protocols*, volume 1361 of LNCS, pages 79–90. Springer, 1997.
- [Mac02] P. MacKenzie. The PAK suite: Protocols for Password-Authenticated Key Exchange. In *Contribution to the IEEE P1363 study group*, May 2002.
- [MSP00] P. MacKenzie, R. Swaminathan, and S. Patel. The PAK suite: Protocols for Password-Authenticated Key Exchange. In *Contribution to the IEEE P1363 study group*, August 2000.
- [MT79] R. Morris and T. Thompson. Password security: a case history. In *Communications of the ACM*, volume 22, pages 594–597. ACM Press, November 1979.
- [Pat97] S. Patel. Number theoretic attacks on secure password schemes. In *IEEE Symposium on Security and Privacy*, 1997.
- [PK99] R. J. Perlman and C. Kaufman. Secure Password-Based Protocol for Downloading a Private Key. In *Proceedings of the Internet Society Network and Distributed Systems Security Symposium*, 1999.
- [PK01] R. Perlman and C. Kaufman. PDM: A New Strong Password-Based Protocol. In *Proceedings of the 10th USENIX Security Symposium*, pages 313–321, August 2001.
- [RCW98] M. Roe, B. Christianson, and D. Wheeler. Secure Password-Based Protocol for Downloading a Private Key. Technical Report 445, University of Cambridge and University of Hertfordshire, July 1998.
- [STW95] M. Steiner, G. Tsudik, and M. Waidner. Refinement and Extension of Encrypted Key Exchange. In *Operating Systems Review*, volume 29, pages 22–30. ACM SIGOPS, July 1995.
- [Wu97] T. Wu. The Secure Remote Password protocol. In *Proceedings of the 1998 Internet Society Symposium on Network and Distributed Systems Security*, pages 97–111, November 1997.
- [Wu02] T. Wu. SRP-6: Improvements and Refinements to the Secure Remote Password Protocol. In *Submission to the IEEE P1363 Working Group*, October 2002.