Zašifrované klíče

Šifrování dat je natolik běžná technika, že nikoho nepřekvapuje. Naproti tomu o šifrování klíčů se mluví velmi málo. Člověk by neřekl, že okolo klíčů se může točit taková věda, ale je to tak. Klíčům se věnuje jiná pozornost než datům.

Ošoupaný klíč generála Yamamota

Na distribuci klíčů doplatila např. ve druhé světové válce japonská armáda, která v důsledku rozlehlosti této ostrovní říše nestačila distribuovat klíče tak často, jak potřebovala. Klíče se používaly dlouho přes jejich životnost a USA díky jejich rozluštění věděly o záměrech protivníka dost dlouho dopředu. Když japonský národní hrdina generál Yamamoto vzlétl na inspekční cestu, netušil, že vlivem příliš „opotřebovaného klíče“ japonských šifrátorů se jeho letový plán stává na stole Američanů plánem na jeho nebeský pohřeb. Dnes jsou používané šifry velmi silné, proto pojem „opotřebovanost“ klíče ztratil z hlediska informační bezpečnosti smysl. Ale zůstává zde stále lidský činitel, který může i dokonalé systémy učinit zranitelnými. Potom takové banální organizační opatření, kterým je pravidelná výměna klíčů, může minimalizovat napáchané škody, neboť data lze dešifrovat zpětně jen po omezenou dobu platnosti kompromitovaného klíče. A tak místo „opotřebovanosti klíče“ se používá pojem „doba platnosti klíče“, který přesně odráží jeho současný smysl a bezpečnostní účel.

Šifrování klíčů v praxi

Se šifrováním klíčů pro blokové šifry se lze běžně setkat např. v protokolu SSL/TLS. Je známo, že při připojování na server SSL/TLS, např. při přístupu k internetovému bankovnictví, počítač vygeneruje náhodný klíč pro blokovou nebo proudovou šifru (AES nebo RC4) a ten serveru pošle zašifrovaně pod jeho veřejným asymetrickým klíčem. To je nejčastější situace šifrování klíče. Ve větších systémech, kde je symetrických a asymetrických klíčů více, vzniká otázka, jak tyto množiny klíčů chránit. Nejjednodušší způsob, který každého napadne, je, že tuto množinu klíčů bude chápat jako soubor dat a ten prostě zašifruje určitým vyšším klíčem. Tento klíč se v aplikované kryptologii nazývá klíč pro šifrování klíčů a má zkratku KEK (Key Encryption Key). Klíče pro šifrování dat jsou prostě klíče nebo se označují jako klíče pro šifrování dat a mají zkratku DEK (Data Encryption Key). Tato terminologie vznikla na počátku devadesátých let minulého století a v praxi a různých standardech se stále používá.

Základní norma pro šifrování klíčů

Jeden z prvních a dosud nejpoužívanějších standardů pro šifrování klíčů stanovil úřad NIST v roce 2001 (AES Key Wrap Specification). Je základem mnoha dalších standardů, např. pro RFC 3394 nebo čerstvý RFC 5649 (z roku 2009). Standard hovoří o tom, že „balík klíčů“ se šifruje blokovou šifrou AES, ale lze použít i jinou blokovou šifru se 128bitovým blokem. Klíč KEK k AES může mít délku 128, 192 nebo 256 bitů. Vstupní „balík klíčů“ se tedy nešifruje AES třeba v modu CBC, ale je na to speciální, (zhruba) šestkrát silnější postup. Detailní popis i testovací vektory lze nalézt ve zmíněných normách, zde bude ukázán princip a vlastnosti tohoto způsobu šifrování. Stojí totiž za pozornost nejen proto, že je velmi zvláštní a velmi silný, ale i proto, že ho lze využít i jinde, kde na šifrování dat velmi záleží. Další jeho důležitou vlastností je, že navíc obsahuje silnou integritní kontrolu, konkrétně 64bitový kryptografický kontrolní součet (viz dále proměnná *A*). Po dešifrování je tedy navíc jisté, že data nebyla poškozena, a současně byl k dešifrování použit správný klíč KEK.

```
Algoritmus šifrování klíčů
Vstup: Klíč K (KEK), 64 bitové bloky dat P[1],
P[2],…, P[n].

A = (hex.) A6A6A6A6A6A6A6A6,
In[1] = P[1], In[2] = P[2], …, In[n] = P[n],
For  j = 0, 1,…, 5
{
    //vnitřní cyklus
    For  i = 1, 2 ,…, n
    {
        B = AES(K, A zřetězeno s In[i]),
        A = (horních 64 bitů B) ⊕ t,
        kde t je čítač t = (n*j)+i,
        Out[i] = dolních 64 bitů B,
    }
    In[1] = Out[1], In[2] = Out[2], …, Out[n] = P[n],
}
C[1] = Out[1], C[2] = Out[2], …, C[n] = Out[n],

Výstup: 64 bitové bloky šifrového textu A, C[1],
C[2],…, P[n].
```

Obr. 1 Algoritmus šifrování klíčů s integritní kontrolou

```
Algoritmus dešifrování klíčů
Vstup: Klíč K (KEK), 64 bitové bloky dat A, C[1],
C[2],…, C[n].

In[1] = C[1], In[2] = C[2], …, In[n] = C[n],
For  j = 0, 1,…, 5
{
    //vnitřní cyklus
    For  i = 1, 2 ,…, n
    {
        čítač t = (n*j)+i,
        B = Inv_AES(K, (A ⊕ t) zřetězeno s In[i]),
        A = horních 64 bitů B,
        Out[i] = dolních 64 bitů B,
    }
    In[1] = Out[1], In[2] = Out[2], …, Out[n] = P[n],
}
P[1] = Out[1], P[2] = Out[2], …, P[n] = Out[n],

Je-li A odlišné od hodnoty (hex.) A6A6A6A6A6A6A6A6,
pak nastala chyba, jinak je následující výstup
platný.
Výstup: 64 bitové bloky otevřených dat P[1], P[2],…,
P[n].
```

Obr. 2 Algoritmus dešifrování klíčů s integritní kontrolou

Zvláštní technika šifrování s integritním kódem

Vstupní data nazývejme prostě daty, přičemž to mohou být klíče a jejich doprovodné informace, integritní kontroly apod. Tato vstupní data se doplní přesně stanoveným způsobem na 64bitové bloky P[1], P[2], …, P[n], kterých je nutné mít k dispozici alespoň n ≥ 2. Výsledkem zašifrování bude o jeden blok více, tj. n + 1 bloků. Jsou označovány A, C[1], C[2], …, C[n].

Na počátku bude A naplněn inicializační hodnotou A = (hex.) A6A6A6A6 A6A6A6A6. Algoritmus zašifrování se skládá z jádra, které volá šestkrát za sebou. V jádru se mění pouze postupně se zvyšující čítač (t). Vnější cyklus algoritmu má tedy proměnnou j = 0, 1, …, 5 a vnitřní cyklus převádí vždy vstup: A, In[1], In[2], …, In[n] na výstup: A (má hodnotu odlišnou od vstupního A), Out[1], Out[2], …, Out[n], přičemž výstup z jednoho vnitřního cyklu se stává vstupem následujícího (schéma na *obr. 1*).

Dešifrování klíčů a kontrola neporušenosti

Dešifrování klíčů a kontrola integrity probíhají tak, že se vstupními daty A, C[1], C[2], …, C[n] se prochází inverzním postupem zpět k hodnotám A, P[1], P[2], …, P[n]. Vyjde-li A = (hex.) A6A6A6A6A6A6A6A6, dešifrování je platné, jinak někde nastala chyba buď v datech, klíči nebo ve výpočtu. Povšimněme si, že algoritmus je reverzibilní, jak ukazuje schéma na *obr. 2*. Podtrhujeme, že algoritmus lze využít i jinde, kde na šifrování dat velmi záleží.

Vlastimil Klíma,
vlastimil.klima@knzsro.cz

LITERATURA

[1] Archiv autora a článků kryptologie pro praxi: http://cryptography.hyperlink.cz.