Odvozování klíčů (2) – KDF a PRF

Klíče hrají v moderní počítačové a internetové kryptografii zásadní roli. Vývojáři aplikací často stojí před problémem, jak je vytvářet. Nejlépe je generovat všechny náhodně, ovšem takovýto postup s sebou nese problémy jejich přenosu. Proto se vytváří klíčové hierarchie, kdy se nižší klíče mohou odvozovat od klíčů vyšších atd. až ke klíči, který je všeho původcem (master key input, KI).

Nejbezpečnější KDF pro náhodné klíče

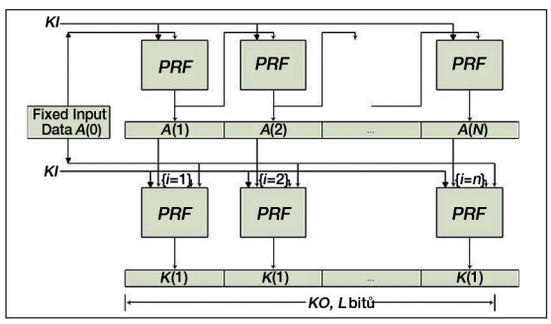
Ještě upozorněme na to, že podstatný rozdíl je v tom, zda vstupní master klíč *KI* je heslo (password), nebo náhodný binární řetězec. Pro každý z nich platí jiné postupy! V minulém čísle Sdělovací techniky začaly být rozplétány normy týkající se odvozování klíčů od náhodného klíče. Byla popsána dvě schémata KDF (key derivation function), která umí ze vstupního klíče *KI* (key input) generovat libovolné množství odvozených klíčů. Tato schémata za stavební blok využívala pseudonáhodnou funkci (PRF), která bude charakterizována právě nyní. Ještě ale uveďme poslední (nejsložitější a nejbezpečnější) schéma KDF pro připomenutí, jak je v něm PRF použita [1] – viz *obr. 1*. Vstup *KI* je vždy klíč pro PRF. Ostatní vstupy jsou „datové“.

Připomeňme krátce, že i je čítač, *KO* je (nekonečný) výstupní klíčový materiál, z něhož je možné odebírat binární řetězce jako klíče, a „*A*(0) = Fixed Input Data“ je určitá nepříliš dlouhá konstanta vztahující se ke kontextu využití klíčů (návěští, identifikátory, nonce atd.). Je zřejmé, že první linie použití pseudonáhodné funkce vytváří posloupnost bloků *A*(*i*), *i* = 1, 2, …, která se použije v druhé linii ke generování klíčového materiálu (*KO*). Jako PRF se používají dvě funkce – první je založena na blokové šifře a druhá na hašovací funkci.

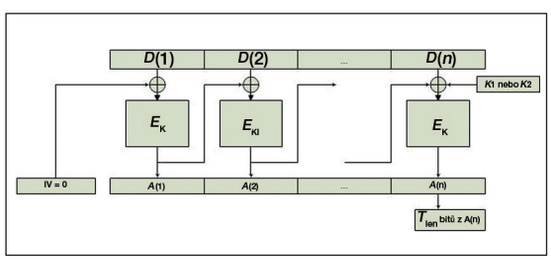
PRF na bázi blokové šifry

Tuto funkci definuje norma [2] a je to tzv. modus CMAC blokové šifry. Připomíná mnohem známější modus CBC, který je určen pro šifrování. Avšak CMAC jako PRF může mít kromě klíčového vstupu ještě dost dlouhý datový vstup, zatímco výstup musí mít pevnou stanovenou délku, a to buď celou délku bloku použité blokové šifry (*T*len = *b* bitů) nebo menší (*T*len ≤ b). CMAC sice jakoby šifruje v módu CBC, ale jako výstup použije jen poslední blok nebo jeho část. Další změna je v tom, že na poslední datový blok ještě před šifrováním načte trochu pozměněný klíč použité blokové šifry. Postup CMAC přesněji ukazuje *obr. 2*.

Definice CMAC se mírně liší pro blokové šifry s blokem délky *b* = 128 a *b* = 64 bitů, pro jiné zatím nejsou specifikovány hodnoty, které jsou uvedeny v *tabulce 1*. Také je nutné vypořádat se s tím, že datový vstup nemusí být zarovnán na bloky. Tak tedy, jsou-li data zarovnána na bloky přesně, nic se nedoplňuje a je zde vstupní posloupnost *b*-bitových bloků *D*(1), …, *D*(*n*). Jestliže datům chybí alespoň jeden bit do plného posledního bloku, doplní se jeden jedničkový bit a poté potřebný počet (0 až *b* – 2) nulových bitů do plného bloku. Tím je hotova také vstupní posloupnost *b*-bitových bloků *D*(1), …, *D*(*n*). Z obrázku je zřejmé, že na poslední blok se přičítá hodnota klíče *K*1 nebo *K*2. Tyto klíče vznikají z klíče *KI* jednoduchou úpravou *KI*. Ukažme jejich definici pro blok délky *b* = 128 (v [2] je podobná definice pro *b* = 64). Důležité je, že *K*1 se použije tehdy, když se poslední blok nedoplňuje, a *K*2, když se doplňuje.



Obr. 1 KDF vytvořená z PRF ve dvojitém iteračním modu



Obr. 2 PRF vytvořená z blokové šifry v modu CMAC

Nejprve se vytvoří základ klíče, $L = E_K(00\ldots0)$. Je to výsledek zašifrování nulového bloku danou blokovou šifrou s klíčem *K*. Výsledkem je 128bitový blok, jehož bity se označí od nejvýznamnějšího k nejméně významnému takto: *L* = (*L*127, *L*126, …, *L*0). Hodnoty *K*1 a *K*2 se liší podle toho, jaké hodnoty mají dva nejvýznamnější bity *L*127 a *L*126 (viz *tabulka 1*). V podstatě jde o to, že bity *L* se posunou a na pár bitů se naxoruje jednička.

PRF na bázi hašovací funkce

Tuto funkci definuje norma [3]; je to známý kód HMAC, tj. PRF(*KI*, *D*) = HMAC(*KI*, *D*). Každý HMAC má vstupní klíč *KI* a data *D* a výstupem je *h*-bitový výstup použité hašovací funkce (*H*). Je-li *KI* stejně dlouhé jako inicializační vektor (*B* bajtů) hašovací funkce *H*, jednoduše HMAC(*KI*, *D*) = *H*(*KI* xor opad, *H*(*KI* xor ipad, *D*)), kde ipad (resp. opad) je konstantní řetězec B bajtů s hodnotou 0×36 (resp. 0×5c). Jestliže je *KI* kratší než *B* bajtů, doplní se nulovými bajty. Když je delší než *B* bajtů, vezme se místo *KI* hodnota *H*(*KI*), která se popř. doplní nulovými bajty do *B* bajtů.

Vlastimil Klíma,
vlastimil.klima@knzsro.cz

Tabulka 1 Definice pomocných klíčů pro modus CMAC

bit:	127	126	125	124	…	8	7	6	5	4	3	2	1	0
L	0	0	*L*125	*L*124	…	*L*8	*L*7	*L*6	*L*5	*L*4	*L*3	*L*2	*L*1	*L*0
*K*1	0	*L*125	*L*124	…	*L*8	*L*7	*L*6	*L*5	*L*4	*L*3	*L*2	*L*1	*L*0	0
*K*2	*L*125	*L*124	…	…	*L*7	*L*6	*L*5	*L*4	*L*3	*L*2	*L*1	*L*0	0	0
L	0	1	*L*125	*L*124	…	*L*8	*L*7	*L*6	*L*5	*L*4	*L*3	*L*2	*L*1	*L*0
*K*1	1	*L*125	*L*124	…	*L*8	*L*7	*L*6	*L*5	*L*4	*L*3	*L*2	*L*1	*L*0	0
*K*2	*L*125	*L*124	…	…	*L*7	*L*6	¬*L*5	*L*4	*L*3	*L*2	*L*1	¬*L*0	1	1
L	1	0	*L*125	*L*124	…	*L*8	*L*7	*L*6	*L*5	*L*4	*L*3	*L*2	*L*1	*L*0
*K*1	0	*L*125	*L*124	…	*L*8	*L*7	¬*L*6	*L*5	*L*4	*L*3	*L*2	¬*L*1	¬*L*0	1
*K*2	*L*125	*L*124	…	…	*L*7	¬*L*6	*L*5	*L*4	*L*3	*L*2	¬*L*1	¬*L*0	1	0
L	1	1	*L*125	*L*124	…	*L*8	*L*7	*L*6	*L*5	*L*4	*L*3	*L*2	*L*1	*L*0
*K*1	1	*L*125	*L*124	…	*L*8	*L*7	¬*L*6	*L*5	*L*4	*L*3	*L*2	¬*L*1	¬*L*0	1
*K*2	*L*125	*L*124	…	…	*L*7	¬*L*6	¬*L*5	*L*4	*L*3	*L*2	¬*L*1	*L*0	0	1

LITERATURA

[1] NIST Special Publication 800-108: Recommendation for Key Derivation Using Pseudorandom Functions (Revised). Computer Security Division, ITL, CS, October 2009.

[2] NIST SP 800-38B, Recommendation for Block Cipher Modes of Operation – The CMAC Mode for Authentication. May 2005.

[3] FIPS 198-1, The Keyed-Hash Message Authentication Code (HMAC). Revision expected to be published in 2008.