Téma 2 – Principy kryptografie

Substitučně-permutační sítě a AES

V on-line světě každý den odešleme i přijmeme celou řadu šifrovaných zpráv. Obvykle se tak děje bez toho, abychom si to jakkoli uvědomovali. Zatímco ještě před několika lety byla i nešifrovaná komunikace na internetu celkem běžná, dnes, pokud si chceme zobrazit webovou stránku nebo někomu zavolat, obvykle šifrování využíváme. Za způsoby, jakými jsou data šifrována, přitom často nejsou schovány nijak složité úvahy. Cílem tohoto tématu bude tyto postupy přiblížit a trochu si je osahat.

V první části se budeme věnovat substitučně-permutačním sítím (substitution-permutation network, SPN), na kterých je založená mimo jiné v dnešní době nejrozšířenější bloková šifra AES.

Historické metody

Potřeba utajovat zasílané vzkazy je stará tisíce let. Například z dob antiky pochází slavná Caesarova šifra, která spočívá v posunutí každého písmena v abecedě o tři, tedy v nahrazení (neboli substituci) jednoho písmena druhým podle pevně stanovené tabulky.

Později byla vymyšlena Vigenèrova šifra, u které vznikne zašifrovaný text postupným přičtením jednotlivých písmen klíče k jednotlivým písmenům původního textu. Pokud se tedy v původním textu (budeme mu říkat *otevřený*) objevuje stejné písmeno vícekrát, může být v zašifrovaném textu zakódované pokaždé pomocí jiného znaku.

Dalším tradičním způsobem je proházení (permutace) pozic písmen obsažených v textu.

Substitučně-permutační sítě v podstatě nepřinášejí nic nového. Pouze všechny tyto tři dávno známé metody šikovným způsobem propojují.

Moderní blokové šifry

Aby matematici mohli šifry lépe popisovat a zkoumat, zavedli pojem „bloková šifra“. Tu si můžeme představit jako sadu dvou černých krabiček. První slouží pro šifrování – pokud do ní vložíme otevřený text a šifrovací klíč, dostaneme zašifrovaný text. Druhá krabička pak umožňuje s pomocí zašifrovaného textu a šifrovacího klíče vytvořit zpátky otevřený text.

Šifrovacím klíčem je vždy nějaká posloupnost znaků („heslo“) předepsané délky. Bloková šifra vždy očekává vstup pevně zadané délky a stejně dlouhý je i její výstup. Této délce se říká *velikost bloku*. Například AES používá bloky o velikosti 128 bitů neboli 16 znaků. Pokud chceme zašifrovat delší text, můžeme ho například rozdělit na části odpovídající velikosti bloku a ty zašifrovat zvlášť.

Dvojková soustava

Než budeme pokračovat, musíme udělat krátkou odbočku. Moderní kryptografie předpokládá využití počítačů, a tak jim přizpůsobuje využívané prostředky. Čísla jsou obvykle zapisována ve dvojkové soustavě. Při početních operacích se typicky počítá s bloky dat (proměnnými) omezené velikosti.

Každé číslo je možné v desítkové soustavě zapsat jako součet jednociferných násobků mocnin deseti, například $2945 = 2\cdot 10^3 + 9\cdot 10^2 + 4\cdot 10^1 + 5\cdot 10^0$. Podobně ve dvojkové soustavě je možné každé číslo zapsat jako součet mocnin dvojky.

Pokud chceme převést číslo z desítkové soustavy do dvojkové, stačí najít vhodné mocniny dvou, jejichž součet odpovídá původnímu číslu. Například:

$$(42)_{10} = 4\cdot 10^1 + 2\cdot 10^0 = 1\cdot 2^5 + 0\cdot 2^4 + 1\cdot 2^3 + 0\cdot 2^2 + 1\cdot 2^1 + 0\cdot 2^0 = (101010)_2$$

Převod z dvojkové soustavy do desítkové probíhá podobně.

Protože čísla ve dvojkové soustavě jsou dlouhá a jejich zápis pro člověka není příliš dobře čitelný, často se místo dvojkové soustavy používá soustava šestnáctková. Způsob převodu je opět úplně stejný. Abychom měli pro šestnáctkovou soustavu dostatečné množství číslic, přidávají se za číslici 9 ještě písmena A až F, například $(42)_{10} = (2\mathrm{A})_{16}$. Všimněte si, že 16 je mocninou 2, převod mezi dvojkovou a šestnáctkovou soustavou je proto velmi jednoduchý. Každá hexadecimální číslice totiž odpovídá čtveřici binárních číslic, např. $(2\mathrm{A})_{16} = (10\ 1010)_2$.

Pro čísla zapsaná ve dvojkové soustavě můžeme snadno definovat takzvané bitové operace, tedy početní operace, které provádíme na jednotlivých bitech (tj. pozicích ve dvojkové soustavě) čísla nezávisle na ostatních. Obvyklými operacemi pro dvě čísla jsou:

and & z čísel vybere pouze ty bity, na kterých je v obou číslech 1 (ostatní bity nastaví na 0), tedy např.

$$\begin{array}{rr} & 10100110 \\ \& & 01101011 \\ \hline & 00100010 \end{array}$$

or $\|$ vybírá ze zadaných čísel všechny bity, na kterých je jednička alespoň v jednom z nich, např.

$$\begin{array}{rr} & 10100110 \\ \| & 01101011 \\ \hline & 11101111 \end{array}$$

xor $\oplus$ na výstup vypíše 1 u těch bitů, kde se zadaná čísla liší, např.

$$\begin{array}{rr} & 10100110 \\ \oplus & 01101011 \\ \hline & 11001101 \end{array}$$

Kromě těchto operací se ještě využívá klasické sčítání (značíme +) na číslech omezené délky, ve kterém zanedbáváme přenos přes nejvyšší řád. Pokud tedy

budeme pracovat s 8-bitovými čísly, tak $10100110 + 01101011 = 00010001$. Hodnota odpovídá běžnému sčítání v desítkové soustavě, při kterém nás ale zajímá ze součtu pouze zbytek po dělení 2^8 (obecně 2^n pro n-bitová čísla).

Navíc ještě definujeme operace, které provádíme vždy pouze s jedním číslem:

not $\sim$ je operace, která prohazuje, na kterých bitech jsou jedničky a na kterých nuly, např. $\sim 10100110 = 01011001$.

shift left $\ll$ posouvá všechny bity o jednu pozici vlevo a na konec přidá nulu. Bit nejvíce vlevo se ztratí. Např. $\ll 10100110 = 01001100$. Tato operace odpovídá násobení 2.

shift right $\gg$ posouvá všechny bity o jednu pozici vpravo a na první pozici přidá nulu. To odpovídá dělení 2, např. $\gg 10100110 = 01010011$.

rot left $\lll$ neboli rotace vlevo funguje obdobně jako shift vlevo, ale bit z nejlevější pozice přesune na konec čísla, např. $\lll 10100110 = 01001101$.

rot right $\ggg$ neboli rotace vpravo opět připomíná shift vpravo, ale poslední bit přesune na začátek čísla, např. $\ggg 10100110 = 01010011$, $\ggg 01101011 = 10110101$.

V tomto dílu využijeme jen některé z těchto operací, ale v průběhu tématu se nám mohou hodit i další.

Úloha 1 [1b]: *Mějme libovolná čísla a a b. Dokážete získat výsledky operací or, xor a rotace vpravo pouze pomocí opakovaného využití operací and a not?*

Šifra OCTA

Vybaveni potřebným matematickým aparátem si můžeme konečně zadefinovat naši vlastní jednoduchou šifru OCTA založenou na substitučně-permutační síti a trochu ji prozkoumat.

OCTA využívá blok o velikosti osmi bitů a očekává klíč, jehož délka v bitech je násobkem osmi. Jak je u moderních blokových šifer obvyklé, bude dělat několikrát po sobě operace zapsané pomocí stejného předpisu. Při tvorbě šifrového textu tedy proběhne několik kol šifrování (těm říkáme *rundy*).

Na začátku každé rundy přixorujeme k aktuálnímu textu klíč. Abychom pořád nexorovali stejnou hodnotu, použijeme poprvé prvních osm bitů klíče (označíme K_1), v druhé rundě dalších osm bitů (K_2) a tak dále. Jakmile dojdeme až na konec klíče, začneme opět od začátku. Pokud by tedy měl klíč například 16 bitů, tak $K_1 = K_3 = K_5 = \ldots$.

Druhou operací v rámci rundy bude využití takzvaných S-boxů neboli substituce podle pevně stanovené tabulky. Text (tj. osm bitů) rozdělíme na čtyři skupiny po dvou bitech a hodnotu každé z nich změníme podle tabulky:

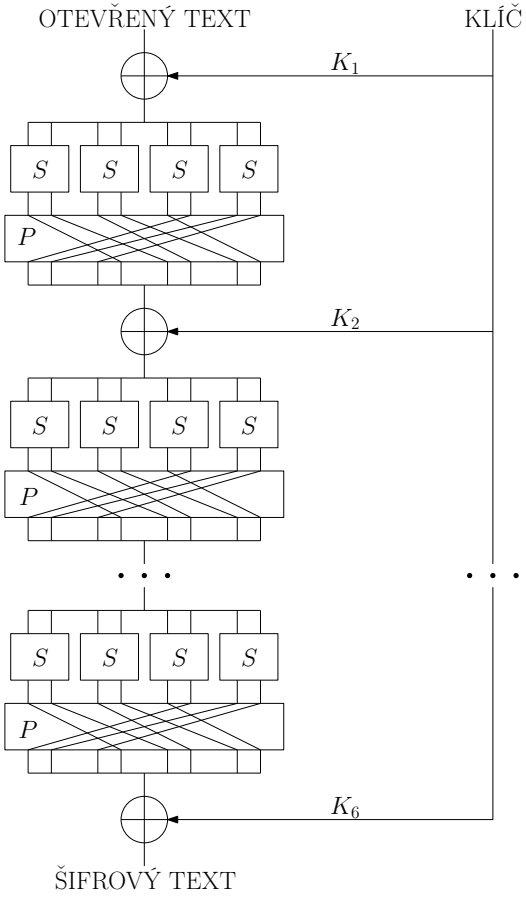
x	00	01	10	11
$S(x)$	10	00	01	11

Tedy například text 01110101 změníme na 00110000.

Třetí operací pak bude třikrát po sobě rotace celého textu vpravo (tedy rotace vpravo o tři bity).

Takových rund proběhne pět. Na závěr ještě jednou přixorujeme klíč.

Schéma celé šifry je zachyceno na Obrázku 1.



Obrázek 1: schéma šifry OCTA

Abychom pomocí naší šifry mohli šifrovat reálné texty, využijeme v souladu se skutečnou praxí reprezentaci znaků pomocí ASCII tabulky[1], která převádí zá-

[1] viz `https://www.asciitable.com/`

kladní tisknutelné znaky na osmibitové hodnoty. Pokud budeme chtít mít v otevřeném textu písmeno A, tak do něj tedy napíšeme $(41)_{16} = (01000001)_2$. Každou osmici bitů (tedy každé písmeno) pak budeme šifrovat zvlášť.

Zbývá otázka, jak bude vypadat funkce pro dešifrování zpráv. Můžeme si ale všimnout, že všechny kroky, které jsme v průběhu šifrování provedli, lze provést i zpětně (všimněte si, že přixorování je samo sobě inverzní operací). Dešifrovat zprávy tedy můžeme pomocí stejného algoritmu, ale pozpátku.

Úloha 2 [2b]: *Dostali jste zprávu 89 19 2C 8C F9 BC FA CC 6D 6C FC BD 19 59 19 C8 29, o které víte, že je zašifrovaná šifrou OCTA pomocí klíče 4D 41 4E 44 4D (tj. MANDM). Jaká byla původní zpráva?*

Využití šifry

Problém [2b+]: *Představme si reálné využití šifry. Chcete mít aktuální přehled o tom, kolik máte doma tabulek hořké čokolády. A tak si každou hodinu posíláte do školy zprávu o jejich počtu. Protože tato informace je ale důvěrná, posíláte ji zašifrovanou pomocí šifry OCTA s klíčem, který znáte jenom vy.*

Je k tomuto účelu OCTA vhodná? Pokud někdo bude odposlouchávat vaše zprávy, dokáže z nich něco zjistit? Jak by bylo možné zasílání zprávy vylepšit?

Zkuste se zamyslet i nad případnými dalšími nedostatky prezentované šifry. Kde všude by mohla v praxi narazit? Za pěkný rozbor lze získat pěkné bodové ohodnocení.

Úloha 3 [2b+]: *Můžete též zkusit zauvažovat nad volbou šifrovacích klíčů. Existuje klíč, který vámi zvolený otevřený text zašifruje tak, že otevřený i šifrový text budou stejné? A existuje klíč, který zašifruje libovolný otevřený text na šifrový text, který bude s otevřeným textem shodný?*

Najdete otevřený text a klíč, který tento text zašifruje na samé 1, tj. FF?

AES

Zatím jsme si hráli s jednoduchou šifrou OCTA. Jak to ale vypadá s reálnými šiframi jako třeba s AES? Od naší šifry se až tak moc neliší. AES používá 128-bitové bloky a klíče o velikosti 128, 192, nebo 256 bitů. Podle velikosti klíčů je využito 10, 12, nebo 14 rund.

Na začátku dojde k přixorování rundovního klíče. Samotná runda pak obsahuje využití S-boxů v kroku nazvaném *SubBytes* a potom dvě permutace *ShiftRows* a *MixColumns*. Runda je zakončena opět přixorováním rundovního klíče.

Detailní popis šifry můžete nalézt třeba na Wikipedii[2].

Úloha 4 [2b]: *V detailním popisu AES si můžete všimnout, že v poslední rundě je vynechaná permutace MixColumns. Proč tomu tak je? A proč není vynechaná i ShiftRows?*

[2] `https://en.wikipedia.org/wiki/Advanced_Encryption_Standard`

Pokud bychom i v naší šifře OCTA vynechali v poslední rundě permutaci, změnili bychom tím obtížnost jejího prolomení?

Poznámka na závěr

Protože se toto téma zabývá v praxi využívanými postupy, na část otázek a úloh může být jednoduché najít odpovědi na internetu. Takový postup je zcela regulérní a nebojte se jej využívat. Kdykoli budete něco tvrdit, tak ale nezapomínejte svá tvrzení zdůvodňovat. Argument „je to pravda, protože to psali na Wikipedii“ uznávat nebudeme. Ale pokud si na Wikipedii přečtete zdůvodnění a to potom využijete ve svém řešení včetně citace zdroje, je to naprosto v pořádku.

Kuba, Káťa a Lenka; `jakub.topfer@matfyz.cz`
e-mailová konference: `krypto@mam.mff.cuni.cz`