Téma 2 – Principy kryptografie

Módy blokových šifer

V tomto čísle se vrátíme k blokovým šifrám a budeme řešit otázku, jak pomocí blokové šifry zašifrovat co nejlépe větší množství dat. Tento problém jsme řešili už v prvním čísle u šifry OCTA. Tam jsme zprávu rozdělili na několik kratších a ty zašifrovali nezávisle na sobě. V tomto čísle se zamyslíme, jaké takový postup má nevýhody a jak to případně udělat lépe.

Řešení 1. série

Na začátku bychom rádi připomněli úlohy z prvního čísla a vybrali z došlých příspěvků jejich pěkná řešení.

Úloha 1

Zadání:
Mějme libovolná čísla a a b. Dokážete získat výsledky operací or, xor a rotace vpravo pouze pomocí opakovaného využití operací and a not?

Řešení:
Správných řešení přišla celá řada. Například tak, že si napíšeme tabulku pravdivostních hodnot, můžeme nalézt a následně ověřit, že platí:

$$a \parallel b = \sim ((\sim x) \,\&\, (\sim y))$$

$$a \oplus b = (a \parallel b) \,\&\, (\sim (a \,\&\, b)) = \sim ((\sim x) \,\&\, (\sim y)) \,\&\, (\sim (a \,\&\, b))$$

Operaci rotace vpravo pomocí and a not vyjádřit nedokážeme. U operací and a or, a tudíž i u všech operací z nich složených, je hodnota výsledného bitu vždy závislá pouze na hodnotách bitů stejného řádu (na stejné pozici v binárním zápisu čísla). Možná i právě proto se jim říká bitové operace. U rotace vpravo je ale hodnota výsledného bitu závislá na bitu jiného řádu (obvykle o jedna vyššího).

Úloha 2

Zadání:
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?

Řešení:
Druhá úloha vyžadovala akorát správně pochopit schéma šifry OCTA. Mnozí řešitelé se pak pouštěli do dlouhých a složitých manuálních výpočtů. Pochvalu zaslouží Lucie Kunčarová a Doc.^{MM} Kateřina Rosická, které si šifru naprogramovaly. Mohou tak jednoduše dešifrovat i další texty.

Zašifrovaný text byl „VOMBAT CHLUPONOSY“.

Úloha 3

Zadání:
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?

Řešení:
Existuje klíč, který zvolený otevřený text zašifruje tak, že otevřený i šifrový text budou stejné? A existuje otevřený text a klíč, který ho zašifruje na FF?

Stačí si kupříkladu uvědomit, že poslední operace, ke které při šifrování dochází, je přixorování klíče K_6. Můžeme si tedy zvolit libovolný otevřený text i prvních pět rundovních klíčů a spočítat mezistav šifry před přixorováním posledního rundovního klíče. Pak už není těžké dopočítat K_6, abychom dosáhli kýženého výsledku.

Otázku, jestli existuje univerzální klíč, který zašifruje libovolný otevřený text na šifrový text, který bude s otevřeným textem shodný, necháme zatím otevřenou. Za její řešení nabízíme tři body.

Úloha 4

Zadání:
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?

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

Řešení:
Úkolem čtvrté úlohy bylo převážně dohledat na internetu informace o AES. Na detailní popis této šifry zde nemáme dostatek prostoru, takže pouze nastíníme řešení. Úpravy v poslední rundě AES jsou čistě z implementačních důvodů, aby bylo možné pomocí co nejpodobnějšího kódu šifrovat i dešifrovat zprávy. Jsou přitom udělány tak, aby to na vlastnosti šifry nemělo vůbec žádný vliv.

Podobně bychom mohli u šifry OCTA vynechat permutaci v poslední rundě. Pokud bychom navíc adekvátně zpermutovali poslední rundovní klíč, dostali bychom stejný výsledek, akorát zpermutovaný.

Problém

Zadání:
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?

Řešení:
Jako řešení problému otiskujeme bez větších redakčních úprav příspěvek Františka Bujnovského. Pokud vám připadá, že v jeho textu něco chybí nebo s něčím nesouhlasíte, nebojte se ho doplnit vlastním článkem.

Využití šifry OCTA ke kontrole zásob čokolády (4b)

František Bujnovský

Nejdříve je třeba si uvědomit, že je více způsobů, jak počet čokolád zašifrovat. Já bych jmenoval asi tři základní:

1. Doma je 5 tabulek hořké čokolády
2. Pět
3. 5

První způsob poskytuje odposlouchači (dále budu značit *OD*) velké množství materiálu, se kterým by mohl pracovat. Obecně pro *OD* je velice obtížné zjistit, kolik rund má šifra, jaký je přesně klíč, jaká je substituce a další... A tak se bude snažit (alespoň já bych to na jeho místě dělal) přiřadit určitému znaku písmeno. Je obecně známo, že samohlásky se v textu opakují častěji než souhlásky, a např. v tomto textu se opakuje třikrát písmeno *A* a čtyřikrát *O*. Mohl by si tedy natipovat, co je tenhle znak a co zase tenhle (možná by ho trochu mohla zmást mezera – některé šifry mezery nepíšou). Taky je podezřelé, že se mění pouze jeden údaj (pár znaků v tvrzení, zbytek zůstává stejný až občas na nějaké skloňování, ale to je zanedbatelné). Časem by *OD* mohlo dojít, že je to právě číslo, které se mění a zbytek zůstává stejný. Protože co jiného by se měnilo? Maximálně by to mohl být nějaký název, ale to je tak všechno, co mě napadá. A zde narážíme na nebezpečí číslic. Když se počet čokolád pohybuje do 10, je to v klidu, ale jak se začne zvyšovat, tak se v desítkové soustavě začnou čísla opakovat (jako v každé jiné soustavě, např. 1 se opakuje v číslech 12, 13, 14). Tedy pokud bychom k zašifrování používali pouze čísla zapsaná v číslicích, mohl by *OD* tohle pozorovat. Ze začátku by si mohl myslet (kdyby mu došlo, že je to číslo), že dvouciferné číslo je čtyřciferné číslo (nezašifrovaná 1 je v ASCII jako $(31)_{16}$), ale časem by mu došlo, že se dvojice znaků opakují a že existuje pouze deset druhů dvojic. Pak by začal tipovat a mohl by naši šifru prolomit. Ovšem na rozdíl od abecedy zde neexistují samohlásky a souhlásky, takže nelze tím, že by tam jedno číslo bylo víckrát než jednou, určit, že je to 1.[1] Tedy tenhle způsob je skoro neprolomitelný. Pouze pokud by se počet čokolád měnil vzestupně po pravidelných mezerách (např. +1) až do doby, kdy se k číslu musí přidat další cifra. V tomto případě by se dalo něco vypozorovat – určit, které znaky znamenají jaké číslo, atd.

[1] *Poznámka redakce: Toto tvrzení obzvláště doporučujeme vaší pozornosti. Souhlasíte s ním?*

Obtížnost šifry by se dala zvýšit odstraněním mezer mezi znaky. U naší šifry jde o to, že dva znaky značí jedno písmeno, číslo, mezeru, nebo jiné ($A = 41$ atd.). Pokud tedy necháváme za každou dvojicí mezeru, je skoro jasné, že máme pouze 10 různých číslic. Co kdybychom ale napsali číslo číslovkami? Obávám se, že je to v podstatě to samé, akorát ve větším měřítku. Budou se tu opakovat min. šestičlenné řetězce znaků (3 = tři = 54 52 49). Když *OD* spočítá počet použitých řetězců, mohl by přijít na to, že se jich opakuje pouze 10. Sice by se některá čísla v šifře nemusela vůbec objevit, ale i tak je podezřelé používat pořád stejné řetězce. Jak bychom to ale mohli ztížit? Např. šifře přidáme ještě něco jako +1. Ke všem údajům v pondělí přičteme 1, ke všem v úterý 2, atd. Nebo prostě v nějakém sledu měnit substituční tabulku. Není to nerozluštitelné, ale *OD* poté potřebuje více materiálu na luštění. Čím nepravidelněji by se tento systém měnil, tím těžší by bylo prolomit ho. Ovšem příjemce a odesílatel se v tomto případě musí domluvit, což je tady jednoduché, protože chodím každý den domů. V případě nějakých mezinárodních komunikací se ale systém nemůže moc měnit, protože oznámení o změně se nemůže poslat v šifře, u které hrozí, že byla prolomena, a tak by se nic nezměnilo – šifra by zůstala dál prolomená.

Také by se to dalo ztížit tím, že bychom 30 minut po poslání pravé poslali falešnou šifru, ve které by byly nějaké jiné údaje (možná i napsané jiným systémem).

Když už *OD* šifru prolomí, je dobré, že ze samotného čísla nic nezjistí (tedy pokud nenapíšeme „Doma je 5 tabulek čokolády“). Může si myslet, že je to počet králíků, peníze, heslo nebo jiné...

K řešení správné šifry by ale bylo dobré vytvořit si tabulku ($A = 41$, B = 42, atd.), nebo naprogramovat si program, protože přepočítání každého znaku zvlášť je časově náročné. V tomto případě by ale samozřejmě mohl *OD* tabulku ukradnout, takže nejlepší bude si ji udržet v paměti.

Ještě bychom mohli změnit tabulku, ale jak už jsem říkal, je blbost zkoušet různé matematické operace, aby nám vyšla ona písmena, protože je strašně mnoho možností. Daleko lehčí je prostě jenom tipovat znaky, které se opakují. Měnit tabulku tedy moc nemá smysl, pokud náhodou *OD* nepoužívá nějaký program, který zkusí několik tisíc možností za sekundu, což je mezinárodně možné, ale v tomto případě spíše nepravděpodobné.

Zadání 3. série

Šifrování delších textů

Vraťme se ke zkoumání možností, jak zašifrovat delší texty. Způsobům, jak pomocí blokové šifry s pevnou délkou bloku zašifrovat delší zprávu se říká módy, ve kterých je tato šifra použita.

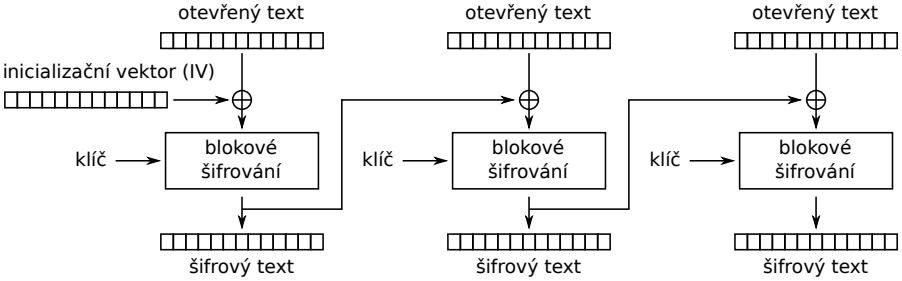
Zatím jsme vždy zprávy rozdělili na několik kratších a ty šifrovali nezávisle na sobě. Tomuto postupu se říká, že šifrujeme v módu *Electronic Codebook (ECB)*. Příspěvek Františka Bujnovského ukazuje na zásadní nedostatek: pokud se ně-

jaké bloky vstupního textu opakují, opakují se ty samé bloky i v šifrovém textu. Speciálně u šifry OCTA tedy šifrování pomocí ECB odpovídá permutaci znaků.

Problém 1 [3b]: *Vyberte si libovolný bitmapový obrázek (například formát BMP) a zašifrujte jeho obrazová data (tedy vše kromě hlaviček) pomocí šifry OCTA s klíčem MANDM (t.j. 4D 41 4E 44 4D) v módu ECB. Měl by opět vzniknout zobrazitelný obrázek. Jak se od toho původního liší? Původní obrázek i jeho zašifrovanou podobu nám zašlete.*

Pokud si nechcete hrát s obrázkovými formáty, tak si vzorový obrázek včetně popisu, jakou oblast je záhodno zašifrovat, můžete stáhnout z našeho webu ze stránky věnované tématku[2].

Proto se v praxi používají spíše jiné módy. Asi nejrozšířenějším je *Cipher Block Chaining (CBC)*. V tomto módu se k otevřenému textu vždy nejdříve přixoruje předchozí zašifrovaný blok a výsledek se následně zašifruje. Protože při šifrování prvního bloku žádný předchozí blok není k dispozici, vygeneruje se náhodný text o velikosti bloku, kterému se říká *inicializační vektor*, a místo předchozího zašifrovaného bloku se použije ten. Hodnota inicializačního vektoru se pak uloží společně se zašifrovanými daty. Schématicky je CBC mód zachycen na Obrázku 1.



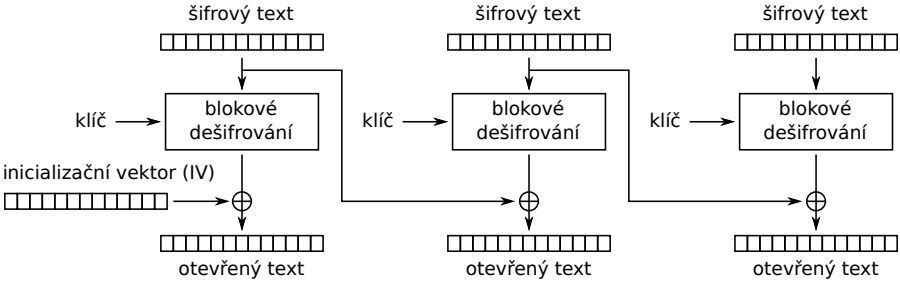
Obrázek 1: Šifrování pomocí CBC módu

Problém 2 [3b]: *Zašifrujete obrázek z Úlohy 1 také pomocí šifry OCTA se stejným klíčem v CBC módu. Jaký je výsledek tentokrát? Opět nám oba obrázky pošlete.*

Můžeme si všimnout, že nyní úprava jednoho bloku ovlivní i všechny následující bloky. Speciálně volba inicializačního vektoru má vliv na všechny bloky šifrového textu. Pokud tedy zašifrujeme ta samá data s jiným inicializačním vektorem, dostaneme zcela odlišný šifrový text. Obvykle se volí šifrovací vektor nový pro každé šifrování. Takže když zašifrujeme vícekrát ta samá data, nebude to na šifrových textech (snadno) poznat.

[2] `https://mam.mff.cuni.cz/problem/2203`

Dešifrování dat zašifrovaných pomocí CBC módu probíhá obráceně než šifrování, tedy nejdříve dešifrujeme šifrový text a následně přixorujeme předchozí blok šifrového textu, případně inicializační vektor. Proces dešifrování zachycuje Obrázek 2.



Obrázek 2: Dešifrování dat zašifrovaných pomocí CBC módu

Problém 3 [2+2b]: *Pokud zašifrujeme hodně dat pomocí šifry s malou velikostí bloku, musí se jednou stát, že dostaneme dva stejné bloky šifrového textu.*

Plynou z toho v případě využití CBC módu pro šifrování nějaká bezpečnostní rizika?

Kolik textu musíme průměrně zašifrovat blokovou šifrou s bloky délky 8 (to je například OCTA), než dostaneme dva stejné bloky šifrového textu? Předpokládejte, že máte náhodný vstupní text a ideální blokovou šifru.

Problém 4 [2b+]: *Nevýhodou CBC módu je, že k zašifrování bloku potřebujeme znát hodnotu předchozího bloku šifrového textu. Nemůžeme tedy výpočty provádět paralelně. Dokážete dohledat nebo vymyslet mód, který bude zachovávat výhody CBC módu a přitom bude umožňovat výpočty provádět (alespoň částečně) paralelně?*

Pokročilejší módy

Občas od módu blokové šifry nechceme pouze, aby zamaskoval situace, kdy máme v otevřeném textu více stejných bloků nebo kdy posíláme vícekrát tu samou zašifrovanou zprávu.

Existují pokročilejší módy, pomocí kterých je možné například ověřit, že zašifrovanou zprávu nikdo neupravoval. K těmto módům se ještě ke konci témátka dostaneme. Pokud by se ale někomu chtělo si nějaké najít a napsat o nich pěkný článek, rádi tento text uveřejníme a autora adekvátně bodově oceníme.

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