Open Source a šifrování dat OFTE

Ing. Pavel Machula, Ph.D.

MWare CZ, s.r.o.

pavel.machula@mware.cz

ABSTRAKT

Článek je zaměřen na porovnání možností *Open Source* nástrojů určených pro šifrovaní dat v reálném čase (OFTE) v prostředí Linux® a prezentaci praktických zkušeností z jejich nasazení v oblasti serverových řešení. Součástí je i stručný popis začlenění problematiky šifrování dat do legislativního rámce ČR. Definice a popis jednotlivých systémů je převzat z veřejně dostupných zdrojů sítě Internet

Klíčová slova

open-source, kódování, šifrování, šifrovací mód, hashovací funkce, souborový systém, jádro, zranitelnost, legislative, OFTE, LUKS

1. ÚVOD

Šifrování informací provází člověka již od nepaměti. Je neoddělitelně spojeno s vývojem matematických metod, způsobem záznamu informací a zpracování. Od dob užití prvních šifrovaných zpráv až po současnost prodělal celý obor složitý vývoj, který akceleruje zejména v posledních 30-ti letech, a to především díky vývoji a nasazení výpočetní techniky, s tím i spjatou nutností rychle a efektivně zabezpečit obsah informací v rámci jejich uložení a přenosu.

V současné době existuje velké množství aplikací s podporou šifrování dat, jak v rámci komerčního software, tak i v oblasti *open-source*[8]. Míra a kvalita zabezpečení obsahu šifrovaných informací obecně závisí především na užitém šifrovacím algoritmu, úrovni aplikace, zabezpečení operačního systému a v neposlední řadě i na legislativních omezeních platných pro daný region. Za nejslabší článek z pohledu šifrování bývá tradičně označen člověk nebo vliv lidského faktoru.

2. ŠIFROVÁNÍ DAT

2.1 Základní pojmy

Pojem šifrování bývá často nesprávně zaměňován s termínem kódování. Z matematického pohledu představuje kódování předpis, který každému prvku z množiny A právě jedno slovo z množiny B. Kód k je tedy zobrazení k: A→B*. Pokud je kódování prováděno jednoznačně, tzn. pokud znakům primární abecedy přísluší různá kódová slova, říkáme, že kódování je prosté. Z pohledu informatiky pak za kód lze považovat způsob vyjádření informace. Kódy lze klasifikovat dle typu jejich aplikace např. na: kódy pro snižování nadbytečnosti a kompresi dat, kódy pro zabezpečení dat proti chybám, kódy pro přizpůsobení informačního zdroje na kanál, šifrovací kódy, kódy pro zrovnoměrnění spektra datových signálů a pod.

Šifrování představuje souhrn kódovacích metod určených pro utajení smyslu zpráv do podoby, která je čitelná jen se speciální znalostí. Šifrování bývá zjednodušeně definováno jako převod otevřeného textu do zašifrované podoby, dešifrování pak jako proces inverzní. Pojem šifra označuje kryptografický algoritmus použitý v procesech šifrování a dešifrování.

Symetrická šifra, někdy též nazývaná konvenční nebo obousměrná je takový šifrovací algoritmus, který používá k šifrování i dešifrování jediný „tajný“ klíč. Výhodu užití symetrických šifer je nízká výpočetní náročnost, nevýhodou nutnost sdílení tajného klíče. Symetrické šifry lze z pohledu způsobu zpracování otevřeného textu rozdělit na blokové šifry, dochází k rozdělení otevřeného textu na bloky shodné velikosti např. 64/128 bit, které jsou dále zpracovány a proudové šifry, které zpracovávají otevřený text po jednotlivých bitech. Mezi symetrické blokové šifry patří např. AES [9], Twofish [10], IDEA [11], Serpent [12], DES [13], mezi proudové šifry např. FISH [14], RC4 [15].

Šifrovací mód představuje způsob užití blokových šifer v případě, kdy má otevřený text více než jeden blok. Některé z šifrovacích módů využívají tzv. inicializační vektor, tj. blok bitů, který je využíván jako „počátek“ pro daný šifrovací mód. Inicializační vektor mívá shodnou velikost jako použitá bloková šifra. Mezi šifrovací módy patří např. ECB, CBC, LRW, XEX, CMC, EME [16][17].

Asymetrické šifrování je založeno na tzv. jednocestných funkcích, pro šifrování a dešifrování používá odlišné klíče veřejný a soukromý. Výhodou užití asymetrického šifrování je možnost volné publikace veřejného klíče. Podstatnou nevýhodou užití asymetrického šifrovaní je velmi vysoká výpočetní náročnost, takže se naprosto nehodí k šifrovaní větších objemů dat. Aplikace tedy používají většinou tzv. hybridní šifrování, kdy se asymetrická šifra používá pouze k šifrování tajného klíče blokových šifer, přičemž otevřený text je šifrován blokovou šifrou. Mezi algoritmy užité pro asymetrické šifrování patří např. RSA [18], Diffie-Hellman [19].

Hashovací funkce [3] představuje reprodukovatelnou metodu určenou pro převod vstupních dat na číslo pevně dané délky. Charakteristickou vlastností hashovacích funkcí je, že malá změna na vstupu vede k velké změně na výstupu, tj. vytvoří na první pohled rozdílný otisk někdy nazývaný též jako *fingerprint* či *hash*. Hashovaní funkce se používají např. pro kontrolu integrity dat, jejich porovnání a indexování. Mezi často užívané hashovaní algoritmy patří např. RIPEMD-160 [20], SHA-512 [21], Whirlpool [22].

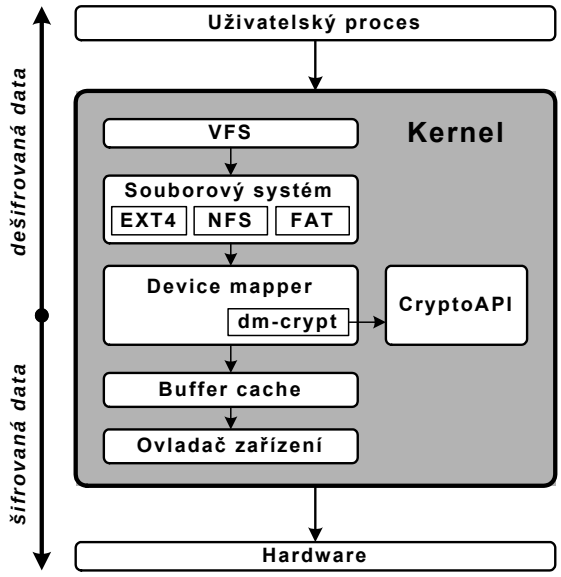
Digitální podpis [2] bývá realizován v podobě otisku hashovaní funkce, který je dále zašifrován privátním klíčem autora dokumentu. Často bývá často distribuován s dokumentem. Na základě příslušného veřejného klíče lze dešifrovat otisk původního dokumentu a porovnat jej s aktuálním otiskem. Digitální podpis tak zajišťuje především integritu a autentifikaci.

2.2 Souborový systém

Pro šifrování dat uložených na disku musí být nezbytně splněny následující podmínky:

- Data na disku musí zůstat důvěrná.
- Musí být zajištěna integrita dat.
- Jak čtení, tak ukládání dat musí být dostatečně rychlé operace bez ohledu na to, kde na disku jsou data uložena.
- Užitá šifrovací metoda nesmí zbytečně plýtvat diskovým prostorem.

Obecně lze data uložená na disku šifrovat jak na úrovni souborového systému, tj. šifrováním obsahu jednotlivých souborů, tak na úrovni diskových oddílů (dále jen jako Diskové šifrování). Diskové šifrování je technicky realizováno jako mezivrstva mezi blokovým zařízením a souborovým systémem.



Obrázek 1. Zjednodušené schéma diskového šifrování

V Linuxu® se jednotlivé souborové systémy mapují na adresáře. To zajišťuje vrstva VFS (*Virtual File System*) [5]. Z uživatelského pohledu se tedy jakýkoliv připojený souborový systém jeví jako běžný adresář. VFS pak pracuje s konkrétními souborovými systémy na jednotlivých oddílech.

V závislosti na platformě a zvolené aplikaci pro diskové šifrování (dále jen aplikaci) je nutné mít vždy správně nainstalované a nakonfigurované nezbytné součásti operačního systému. V případě většiny platforem založených na OS Linux bývá vyžadována podpora CryptoAPI [23] popřípadě i *device-mapperu*[24]. CryptoAPI je rozšíření jádra obsahující implementace šifrovacích algoritmů, šifrovacích módů a hashovacích funkcí. *Device mapper* je ovladač pro linuxové jádro, který spravuje diskové oddíly, ale jejich ovládání ponechává na aplikaci v uživatelském prostoru. Je základem pro konstrukci softwarového RAIDu, LVM, EVMS a *dm-cryptu*.

Šifrování v reálném čase (*On-The-Fly encryption*) OFTE [25] umožňuje dostupnost souborů ihned po zadání klíče. Data jsou automaticky šifrována/dešifrována v okamžiku před načtením nebo uložením na disk bez jakéhokoliv zásahu uživatele. Data uložená na šifrovaném oddílu nemohou být správně načtena (dešifrována) bez zadání správného hesla/souboru nebo šifrovacích klíčů. Celý šifrovaný oddíl je připojen jako by tvořil abstraktní blokové zařízení. Šifrované svazky OFTE tak mohou být uloženy na diskových nebo logických oddílech, celých discích, stejně i jako v podobě samostatných souborů. OFTE je možné použít v rámci RAIDu a LVM.

2.3 Volba parametrů

Je nutné si uvědomit, že šifrování většího objemu dat v rámci souborového systému má přímý vliv na zatížení procesoru a snížení přístupové doby, tj. času, za který je blokové zařízení připraveno číst nebo zapisovat data. Volba správné aplikace, šifry a šifrovacího módu má zásadní vliv na zabezpečení uložených dat, výkon i stabilitu celého systému. Další důležitá kritéria jsou:

- Zda šifrovat stávající oddíl (již existující souborový systém)
- Zda má k šifrovaným datům přístup více uživatelů nebo zda se jedná o čistě privátní data.
- Požadavek na tzv. multi-platformní přístup, tj. zda data mají být přístupná jak z prostředí Linux®, tak i z prostředí Microsoft®, s tím souvisí i volba souborového systému (např. FAT32/NTFS),
- Zda existuje legislativní omezení pro šifrování dat nebo postih za odmítnutí sdělení hesla/klíče soudu či orgánům činným v trestním řízení.

Většina aplikací umožňuje výběr z několika blokových šifer, nebo i jejich vzájemnou kombinaci. Zvolená šifra má hlavní vliv na rychlost provádění operací při šifrování a dešifrování dat. Velmi také záleží na objemu aktuálně zpracovaných dat, kdy při různém objemu šifrovaných dat může z pohledu rychlosti zpracování dojít ke změně pořadí šifrovacích algoritmů. Pokud je k dispozici dostatečně výkonný hardware, je vhodné vždy užít kombinaci alespoň dvou šifer, např. AES-Twofish, Serpent-AES nebo Serpent-Twofish.
Neméně důležitou roli hraje i správný výběr hashovací funkce a šifrovacího módu. V případě hashovaní funkce by měla být její velikost rovna velikosti zvolené šifry. U některých v minulosti hojně využívaných hashovacích algoritmů byla objevena jejich zranitelnost, proto nejsou v současné době považovány za bezpečné a nelze je tedy doporučit pro užití v rámci diskového šifrování. Jedná se především o algoritmy: MD5[26], SHA-0[27], RIPEMD [28], SHA1 [27]. Za relativně bezpečné jsou v současné době považovány např. algoritmy: RIPEMD-160, SHA-512, Whirlpool.
Podobně byla i zranitelnost objevena u některých šifrovacích módů, kdy aplikací vhodného typu útoku může teoreticky dojít k prolomení šifry. Zásadně se nedoporučuje užití módu ECB, který šifruje každý blok nezávisle. Nedoporučuje se ani užití CBC a CBC-ESSIV.

Tabulka 1. Zranitelnost šifrovacích módů

Zranitelnost	CBC	CBC-ESSIV	LRW	CMC, EME

content-leaks	ano	ano	ne	ne
watermark	ano	ne	ne	ne
malleable	ano	ano	ne	ne
movable	ano	ano	ne	ne
modification leak	ano	ano	ano	ne

Za vodné šifrovací módy pro diskové šifrování lze z pohledu poměru míry zabezpečení a výkonu považovat LRW a XEX. Šifrovací módy CMC a EME lze považovat za bezpečnější ovšem při daleko větším nároku na výpočetní výkon, kdy dochází k dvojímu šifrování každého bloku.
Volba správného hesla patří mezi kritické faktory každého šifrování. Obecně lze konstatovat, že správné heslo pro diskové šifrování by mělo z důvodu zaručení dostatečné entropie obsahovat minimálně 20 znaků obsahujících velká a malá písmena, čísla a speciální znaky. V hesle by se neměly vyskytnout slovníkové výrazy a nemělo by být odhadnutelné, tj. nemělo by mít žádnou souvislost s osobností uživatele.

2.4 Bezpečnostní rizika

Diskové šifrování s sebou přináší i nová bezpečnostní rizika, např. zranitelnost způsobenou narušením integrity na úrovni šifrovací vrstvy a směrem níže. Např. drobná změna šifrovaného textu v rámci jednoho bloku může v závislosti na zvoleném šifrovacím módu způsobit buď částečnou nebo i úplnou nedostupnost původně šifrovaných dat. Pokud je uvedený postup proveden úmyslně, lze jej označit za zvlášť nebezpečný DoS (*Deny of Service*) útok [2], který může mít naprosto fatální následky.

Změna nebo nedostupnost dat ve vyšších vrstvách, nemívá zpravidla tak závažný dopad. Účinným eliminačním nástrojem v tomto případě bývá užití programu určeného k opravě chyb v rámci příslušného souborového systému.

V obou případech platí málo zdůrazňovaná zásada, že se zavedením šifrování diskových dat je vždy bezpodmínečně nutné zajistit i jejich pravidelné zálohování včetně kontroly. Zálohování musí být prováděno bezpečným kanálem na bezpečné médium v souladu se stanovenou politikou RPO (*Recovery Point Objective*) [2] a RTO (*Recovery Time Objective*) [2]. Před provedením oprav detekovaných chyb souborového systému je vždy nutné provést mimořádnou zálohu stávajících dat.

Mezi často podceňovaná rizika patří tzv. falešný pocit bezpečí. Tento typ rizika nezahrnuje pouze riziko nepovoleného získání dat, nýbrž veškerá rizika spojená s ohrožením integrity dostupnosti a zabezpečení dat včetně všech výše uvedených.

3. DM-CRYPT

DM-Crypt [29] je nativní součástí Linuxu, využívá v zmíněné linuxové CryptoAPI i *device mapper*. Jedná se o nástroj určený výhradně pro platformu Linux vyžadující jádro verze 2.6 a výše. Zpravidla bývá dodáván jako základní balíček v rámci jednotlivých distribucí, není tedy nutné provádět kompilaci zdrojového kódu a následnou instalaci.

Výhodou nástroje *dm-crypt* je fakt, že je postaven na standardu LUKS (*Linux Unified Key Setup*) [30]. Zatímco některé nástroje pro šifrování disku používají různé, nekompatibilní, nedokumentované formáty, LUKS standardizuje na platformě nezávislé diskové šifrování. Standard LUKS používá tzv. dvojúrovňové šifrování, kdy vlastní data nejsou šifrována uživatelem zadaným klíčem, nýbrž náhodně vygenerovaným tzv. *master-key*. Pro derivaci klíče se používá funkce PBKDF2[31], která posiluje entropii slabých hesel. Kontrolní součet každého z klíčů je uložen v hlavičce. Maximální počet klíčů je 8. Do slotů pro uživatelské klíče se ukládá uživatelským klíčem zašifrovaný *master-key*. Použitím kteréhokoliv z klíčů je možné získat *master-key* a dešifrovat diskový oddíl. Podstatnou výhodou dvojúrovňového šifrování je revokace stávajících klíčů, kterýmkoliv z validních klíčů je možné revokovat ostatní, přičemž není nutné dešifrovat a znovu zašifrovat diskový oddíl s nově vygenerovaným *master-key*.

Konfigurace *dm-cryptu* se může částečně lišit v závislosti na linuxové distribuci. Nezbytnou součástí je správně zkompilované jádro včetně modulů:

- CONFIG_BLK_DEV_DM
- CONFIG_DM_CRYPT

Veškeré použité skripty jsou platné pro distribuci *Slackware* ver. 13 [32]. Základní informace o šifrovaných svazcích, jejich pojmenování a způsobu mapování je obsažena v souboru /etc/cryptsetup.

```
# /etc/cryptab
enc         /dev/sda2
swap        /dev/sda3  /dev/urandom        swap
```

Skript 1. Příklad souboru /etc/cryptab

Proto, aby se požadované změny projevily po nabootování systému je nutné upravit konfigurační soubor /etc/fstab.

```
# /etc/fstab
#/dev/sda3            swap      swap      defaults    0   0
/dev/mapper/swap      none      swap      sw          0   0
/dev/sda1             /         ext3      defaults    1   1
/dev/mapper/enc       /mnt/enc  ext3      defaults    1   2
```

Skript 2. Příklad souboru /etc/fstab

Výše uvedený příklad názorně demonstruje způsob, jak lze mj. elegantně nastavit automatické šifrovaní swapového oddílu. Ten je plněn pseudonáhodnými znaky zařízení /dev/urandom. Pro přístup k datům šifrovaného svazku /dev/sda2 je nutné provést autentikaci. Tu lze obecně provádět buď zadáním hesla z konzole, nebo i odkazem na autentikační soubor (např. na USB disku). Šifrovaný oddíl lze vytvořit např. následujícím skriptem:

```
#!/bin/sh

DEVICE="/dev/sda2"
DEVICENAME="enc"

CRYPTSETUP="/sbin/cryptsetup"
ECHO="/bin/echo"
MKFS="/sbin/mkfs.ext3"

$CRYPTSETUP -c aes-lrw-benbi -h sha512 -s 384 –y \
luksFormat $DEVICE
$CRYPTSETUP luksOpen $DEVICE $DEVICENAME
$MKFS /dev/mapper/$DEVICENAME
$CRYPTSETUP luksDump $DEVICE
```

Skript 3. Příklad vytvoření šifrovaného oddílu

Základní informace pro správnou konfiguraci a použití nástroje *dm-crypt* lze získat zadáním příkazu man cryptsetup nebo zde [29]. Za hlavní výhody tohoto nástroje osobně považuji možnost užití více hesel pro přístup k datům šifrovaného oddílu a možnost šifrování swapového oddílu. Naopak slabá místa spatřuji v nemožnosti specifikace oprávnění na principu ACL (*access control list*) pro provádění operací s klíči a stávající slabou interoperabilitu s platformou Microsoft, kde lze pro přístup k datům použít pouze nástroj FreeOFTE [33], který funguje pouze podobně jako např. WinZIP [34], nikoliv na principu *on-the-fly encryption*.

Při užití tohoto nástroje důrazně doporučuji:

- Po každé změně šifrovací hlavičky (přidání, odebrání, změna stávajícího hesla) provést export (*luksDump*) hlavičky do souboru a ten uložit na bezpečném místě,
- Pravidelně provádět zálohu dat na externí disk s využitím jiného šifrovacího nástroje podobných kvalit.

4. TRUECRYPT

TrueCrypt [35] je *open source* nástroj pro OFTE šifrování pro operační systémy Microsoft Windows, Linux a Mac OS X. Umožňuje vytváření virtuálních disků v podobě souboru, který lze snadno připojit a pracovat s ním jako s jakýmkoliv jiným pevným diskem, nebo zašifruje celý diskový oddíl. Od verze 5.0 umí také šifrovat oddíl, ze kterého se bootuje operační systém (toto platí pouze pro verzi Windows). Od veze 6.0 byl implementován standard PKCS 11 [36].

Šifrovací schéma TrueCryptu je detailně popsáno na webových stránkách v sekci Dokumentace. V porovnání s výše zmíněným dm-cryptem TrueCrypt nepoužívá standard LUKS. Data jsou šifrována pouze jedním klíčem, který je součástí hlavičky a generuje se pomocí pseudonáhodné funkce [37]. Na kryptografickou sílu klíče má zásadní vliv sběr náhodných dat (tzv. *Random Pool*). Je proto velmi vhodné věnovat dostatečný čas sběru náhodných dat ve formě náhodných tahů myší v rámci dialogového okna při vytváření šifrovaného disku tak, aby generovaný klíč byl co nejvíc náhodný. Pro volbu hesla platí výše zmíněné doporučení o min. velikosti 20 znaků.

Tabulka 2. Šifrovací algoritmy programu TrueCrypt

Algoritmus	Autor	velikost klíče [bit]	velikost bloku [bit]
AES	J. Daemen, V. Rijmen	256	128
Serpent	R. Anderson, E. Biham, L. Knudsen	256	128
Twofish	B. Schneier, J. Kelsey, D. Whiting, D. Wagner, C. Hall, N. Ferguson	256	128
AES-Twofish	-	256,256	128
AES-Twofish-Serpent	-	256,256,256	128
Serpent-AES	-	256,256	128
Serpent-Twofish-AES	-	256,256,256	128
Twofish-Serpent	-	256,256	128

TrueCrypt umožňuje vytváření tzv. skrytých oddílů. Jedná se o vytvoření vnitřního TrueCryptového oddílu v rámci volného prostoru již existujícího (vnějšího) TrueCryptového oddílu. Při namountování vnějšího oddílu, není možné prokázat zda existuje vnitřní skrytý oddíl, protože volné místo na kterémkoliv svazku je vždy vyplněno náhodnými znaky, tzn. nelze detekovat žádnou část skrytého oddílu – např. hlavičku a pod. K vyplnění náhodnými znaky dochází již při vytváření oddílu. Heslo pro skrytý oddíl musí být podstatně odlišné od hesla pro vnější oddíl. Do vnějšího oddílu se doporučuje nakopírovat nějaká smysluplně vypadající data, která ve skutečnosti nemusí být skrytá. Novější verze TrueCryptu umožňují vytvoření skrytého bootovacího oddílu.

Hrozbu pro integritu a dostupnost dat skrytého oddílu představuje dodatečná alokace diskového prostoru v rámci vnějšího oddílu, kdy např. při uložení souboru dojde k fyzickému zápisu na místo stávajícího skrytého oddílu. V případě, že se zápisem přepíše hlavička skrytého oddílu je dopad fatální. Zde platí shodná doporučení jako v případě dm-cryptu.

Šifrování/dešifrování probíhá transparentně při zápisu/čtení z disku na pozadí a uživatel se tak nemusí o nic starat. K souborům lze po připojení jednotky k souborovému systému počítače přistupovat běžným způsobem, což se stane až po zadání hesla, šifrovacího klíče. V případě, že je médium chráněno proti zápisu, tak disk bude připojen, ale nebude umožněno na něj zapisovat.

Narozdíl od standardní programové kompilace v prostředí Linux (*configure, make, make install*) se kompilace TrueCryptu podstatně liší. Nejprve je nutné stáhnout a rozbalit zdrojové kódy [35] pro Linux a vývojové prostředí vxWidgets[38]. Od verze 6.0 jsou vyžadovány a hlavičky knihovny pcks11[39]. Kompilace s podporou grafického rozhranní sestává z následujících kroků:

```
export PKCS11_INC=/usr/lib/pkcs11

make NOGUI=0  WX_ROOT=/usr/src/wxWidgets wxbuild
make NOGUI=0  WXSTATIC=1
```

Skript 4. Kompilace programu TrueCrypt

Základní informace pro správnou konfiguraci a použití lze získat zadáním příkazu *truecrypt --h*. Za hlavní výhody truecryptu považuji možnost užití v rámci různých typů operačních systémů a možnost použití skrytých oddílů. Jako nevýhodu bych označil nemožnost užití více hesel v rámci jednoho šifrovaného oddílu.

5. LEGISLATIVA ČR

Současná legislativa ČR neupravuje (tudíž ani neomezuje) šifrování dat jak pro fyzické, tak ani pro právnické osoby. Není upraveno, jaká data smí či nesmí

být šifrována a jaké šifrovací algoritmy, z pohledu síly zabezpečení, smí či nesmí být použity.

Obecně lze konstatovat, že jednotlivé zákony zpravidla nerozlišují významový rozdíl mezi pojmy kódování a šifrování (viz úvodní definice pojmů), často oba pojmy uvádí souběžně, což může v některých situacích způsobit nejednoznačný výklad v souvislosti s původním záměrem zákonodárce. Jako příklad lze uvést [6][7]:

- zákon č. 121/2000 Sb. (autorský zákon), §43 odst. 3,
- zákon č. 127/2005 Sb. (o elektronických komunikacích), §75 odst. 1, §97 odst. 6,
- zákon č. 336/2005 Sb. (informace o účastnících telefonní služby), §8 Obecné technické podmínky pro odposlech a záznam zpráv, odst. 4,
- zákon č. 137/2006 Sb. (o veřejných zakázkách), §149 odst. 3.

Orgány činné v trestním řízení nemají v rámci stávající legislativy legální způsob, jak přimět "zatvrzelého" pachatele ke sdělení přístupových hesel nebo šifrovacích klíčů, mj. s poukazem na ust. §33 ost. 1 vetu TrŘ, podle níž obviněný má právo vyjádřit se ke všem skutečnostem, které mu kladou za vinu, a k důkazům o nich, není však povinen vypovídat, jakož i vzhledem k ust. čl 37 ost. 1 Listiny základních práv a svobod č. 2/193 Sb., vč. znění zákona č. 162/1998 Sb., podle něhož každý má právo odepřít výpověď, jestliže by jí způsobil nebezpečí trestního stíhání sobě nebo osobě blízké [4].

Některé státy EU v rámci kampaně boje s terorismem výrazně omezily osobní svobody včetně práva nevypovídat. Konkrétním případem je Spojené království Velké Británie a Severního Irska, kde na základě zákona *Regulation of Invesstigatory Powers Act 2000* a dodatku s15 *Terrorism Act 2006* hrozí v případě odmítnutí výpovědi vztahující se k šifrované informaci ohrožující národní bezpečnost odnětí svobody na dobu 5 let nepodmíněně v ostatních případech na 2 roky nepodmíněně! Přijaté usnesení nemá retroaktivní charakter [40].

6. ZÁVĚR

Je nesporné, že v současné době je ochrana a zabezpečení dat prioritou jak na osobní, tak korporátní úrovni. Šifrování představuje jeden z velmi účinných nástrojů před nepovolaným přístupem k datům. Před volbou šifrovacího nástroje a algoritmu je nezbytně nutné zkoumat legislativní omezení a formy případného postihu v rámci daného regionu.

Jako vhodné *open-source* šifrovací nástroje jak pro osobní, tak korporání užití lze doporučit *dm-crypt* a TrueCrypt. *Dm-crypt* se spíše hodí pro serverová řešení, kdy se předpokládá větší počet oprávněných uživatelů. TrueCrypt je vhodnější pro osobní užití a zálohování dat.

Kdo je připraven, není překvapen!

LITERATURA A ODKAZY

[1] Garfinkel, S., PGP: Prejty Good Privacy – šifrování pro každého, Praha, Computer Press / O'Reily, ISBN 80-7226-054-5

[2] Harris, S., CISSP Exam Guide, 3. vydání, Emerywille CA, McGraw-Hill/Osborne, ISBN 0-07-225712-1

[3] Singh, S., Kniha kódů a šifer, Praha, ARGO, 2003, ISBN 80-7203-499-5

[4] Smejkal, V. a kolektiv, Právo informačních a telekomunikačních systémů, 2., aktualiz. a rozš. vyd., Praha, C. H.. Beck, 2004, ISBN 80-7179-765-0

[5] Torvalds, L., Linux – Dokumentační projekt, Praha, Computer Press, 1998, ISBN 80-7226-114-2

[6] Informační systém ASPI, Aspi Publishing, http://www.aspionline.cz

[7] Informační systém Codexis Advokacie, Atlas Consulting, http://www.codexisadvokacie.cz

[8] http://en.wikipedia.org/wiki/Open_source

[9] http://en.wikipedia.org/wiki/AES

[10] http://en.wikipedia.org/wiki/Twofish

[11] http://en.wikipedia.org/wiki/International_Data_Encryption_Algorithm

[12] http://en.wikipedia.org/wiki/Serpent_(cipher)

[13] http://en.wikipedia.org/wiki/Data_Encryption_Standard

[14] http://en.wikipedia.org/wiki/Fish_(cryptography)

[15] http://en.wikipedia.org/wiki/RC4

[16] http://en.wikipedia.org/wiki/Block_cipher_modes_of_operation

[17] http://en.wikipedia.org/wiki/Disk_encryption_theory

[18] http://en.wikipedia.org/wiki/RSA

[19] http://en.wikipedia.org/wiki/Diffie-Hellman

[20] http://en.wikipedia.org/wiki/RIPEMD-160

[21] http://en.wikipedia.org/wiki/SHA-512

[22] http://en.wikipedia.org/wiki/Whirlpool(cryptography)

[23] http://en.wikipedia.org/wiki/Cryptographic_API

[24] http://en.wikipedia.org/wiki/Device_mapper

[25] http://en.wikipedia.org/wiki/On-the-fly_encryption

[26] http://en.wikipedia.org/wiki/MD5

[27] http://en.wikipedia.org/wiki/SHA-1

[28] http://en.wikipedia.org/wiki/RIPEMD

[29] http://en.wikipedia.org/wiki/Dm-crypt

[30] http://code.google.com/p/cryptsetup

[31] http://en.wikipedia.org/wiki/PBKDF2

[32] http://www.slackware.com

[33] http://en.wikipedia.org/wiki/Freeotfe

[34] http://en.wikipedia.org/wiki/WinZip

[35] http://www.truecrypt/org/

[36] http://en.wikipedia.org/wiki/PKCS11

[37] http://en.wikipedia.org/wiki/Pseudorandom_function

[38] http://www.wxwidgets.org

[39] http://www.rsa.com/rsalabs/node.asp?id=2133

[40] http://www.lightbluetouchpaper.org/2007/09/30/time-to-forget