Petr Švenda <xsvenda@fi.muni.cz>

Srovnání standardu AES s algoritmy 3DES a IDEA

P995 – Projekt, zadal: V. Matyáš

1. Specifikace algoritmů

3DES
Autoři: Původní návrh Lucifer (IBM) , úpravy (délka klíče, S-box) NIST
Popis: ANSI X9.52
Licence: bez omezení

Jedná se o symetrický šifrovací algoritmus s délkou klíče 128 nebo 192 bitů (112 nebo 168 bitů při 56 bitové délce klíče pro DES) a šifrovaným blokem délky 64 bitů založeným na algoritmu DES (specifikace ANSI X9.52).

Operace šifrování a dešifrování algoritmu 3DES vzniknou složením operací algoritmu DES. Nechť $E_K(I)$, resp. $D_K(I)$ reprezentuje šifrování resp. dešifrování bloku I algoritmem DES s použitím 64 (56) bitového klíče K.

Šifrování: $O = E_{K3}(D_{K2}(E_{K1}(I)))$

Dešifrování: $O = D_{K1}(E_{K2}(D_{K3}(I)))$

Standard specifikuje následující kombinace klíčů K_1, K_2 a K_3:
1. Klíče K_1, K_2 a K_3 jsou nezávislé. Celková délka klíčů 192 (168) bitů.
2. Klíče K_1 a K_2 jsou nezávislé, $K_1 = K_3$. Celková délka klíčů 128 (112) bitů.
3. $K_1 = K_2 = K_3$. Celková délka klíčů 64 (56) bitů. Ekvivalentní s DES.

IDEA

Autoři: X. Lai, J. Massey
Popis: http://www.ascom.ch/Web/systec/security/idea.html
Licence: bez omezení pro nekomerční užití, patent Ascom Systec Ltd.

Jedná se o symetrický šifrovací algoritmus s délkou klíče 128 bitů a šifrovaným blokem délky 64 bitů. Šifrovací klíč (Cipher Key) se rozdělí na osm 16 bitových klíčů, které jsou prvními osmi podklíči (subkeys). V dalším kroku je původní 128 bitový klíč cyklicky posunut o 25 bitů doleva a rozdělen na dalších osm 16 bitových podklíčů. Tento krok se opakuje, dokud není vytvořeno 52 podklíčů s1, s2, ... s52. Vstupní 64 bitový blok je rozdělen na čtyři 16 bitové bloky p1, p2, p3, p4. Při šifrování se užívá operace násobení modulo ((2^16)+1) a sčítání modulo (2^16). Každá z osmi rund se skládá z následujících kroků (označení podklíču odpovídá první rundě):

p1 x s1 --> d1
p2 + s2 --> d2
p3 + s3 --> d3
p4 x s4 --> d4
d1 XOR d3 --> d5
d2 XOR d4 --> d6
d5 x s5 --> d7
d6 + d7 --> d8
d8 x s6 --> d9
d7 + d9 --> d10
d1 XOR d9 --> d11
d3 XOR d9 --> d12
d2 XOR d10 --> d13
d4 XOR d10 --> d14

Následuje výměna bloků d12 a d13, takže bloky d11, d13, d12 a d14 jsou v tomto pořadí použity jako vstup další rundy spolu s dalšími šesti podklíči s7 až s12. Po dokončení všech osmi rund jsou výsledkem 16 bitové bloky e1, e2, e3 a e4. K dokončení šifrování jsou provedeny ještě následující čtyři kroky:

e1 x s49 --> c1
e2 + s50 --> c2
e3 + s51 --> c3
e4 x s52 --> c4

Bloky c1, c2, c3, a c4 jsou spojeny do výsledného 64 bitového zašifrovaného bloku.
Pozn.: Pro účely algoritmu je klíč skládající se ze samých nul ekvivalentní klíči 2^16.
Inverzní algoritmus využívá stejnou posloupnost operací, inverzní podklíče k s1, ... s52 jsou definovány následovně:

1 runda	s49*	s50#	s51#	s52*	s47	s48
2 runda	s43*	s45#	s44#	s46*	s41	s42
3 runda	s37*	s39#	s38#	s39*	s35	s36
4 runda	s31*	s33#	s32#	s34*	s29	s30
5 runda	s25*	s27#	s26#	s28*	s23	s24
6 runda	s19*	s21#	s20#	s22*	s17	s18
7 runda	s13*	s15#	s14#	s16*	s11	s12
8 runda	s7*	s9#	s8#	s10*	s5	s6

Koncová transformace.....s1* s2# s3# s4*

sXX* = inverze vzhledem k násobení klíče sXX modulo ((2^16)+1)
sXX# = inverze vzhledem ke sčítání klíče sXX modulo (2^16)

AES
Autoři: J. Daemen, V. Rijmen
Popis: http://csrc.nist.gov/encryption/aes/
Licence: bez omezení

Tento algoritmus (původním jménem Rijndael) byl vybrán v roce 2000 jako Advanced Encryption Standard (AES). Jedná se o symetrický šifrovací algoritmus s délkou klíče 128-, 192- a 256-bitů aplikovaný na bloky délky 128- 192-, 256-bitů. Velikost klíče a bloku je nezávislá. Šifrovací klíč (Cipher Key) se expanduje na rozšířený klíč (Expanded Key), jehož části se následně užívají pro šifrování bloku v jednotlivých rundách. Počet rund přímo závisí na délce klíče a délce bloku, pohybuje se v rozmezí od 10 do 14. Expanze se liší v závislosti na délce klíče. Stav bloku po jednotlivých transformacích se nazývá State. Šifrování se skládá z iniciální operace AddRoundKey, (celkový počet rund – 1) obyčejných rund (Round) a finální rundy (FinalRound). Definice inverzního algoritmu je odlišná, ale umožňuje částečné využití původních operací. Poslední runda se od předchozích liší z důvodu snahy o zmenšení odlišnosti inverzního algoritmu.
V pseudo C notaci :

```
Round (State, RoundKey)
{
ByteSub(State);
ShiftRow(State);
MixColumn(State);
AddRoundKey(State, RoundKey);
}

FinalRound(State, RoundKey)
{
ByteSub(State);
ShiftRow(State);
AddRoundKey(State, RoundKey);
}
```

ByteSub: Jednotlivé slabiky v bloku jsou nahrazeny svými ekvivalenty z nelineární převodní tabulky.
ShiftRow: Skupiny slabik bloku jsou podrobeny cyklickému posunu. Velikost posunu závisí na velikosti šifrovaného bloku.
MixColumn: Skupiny čtyř slabik jsou podrobeny násobení definovanou maticí. To umožní každé slabice ovlivnit ostatní slabiky skupiny.
AddRoundKey: Blok je podroben operaci XOR spolu s klíčem (RoundKey) pro tuto rundu získaného z rozšířeného klíče (Expanded Key).

Inverzní algoritmus využívá operace inverzní vzhledem k ByteSub, ShiftRow, MixColumn. Operace AddRoundKey je nezměněna (je sama k sobě inverzní).

2. Srovnání algoritmů AES, 3DES a IDEA

- Porovnání návrhu

	AES	IDEA	3DES
délka klíče (bity)	128, 192, 256	128	64, 128, 192 (56, 112, 168)
délka bloku (bity)	128, 192, 256	64	64
S-box	ano (známa motivace)	ne	ano (bez motivace)
Feistel	ne	ne	ano
operace	XOR, cyklický posun, maticové sčítání a násobení v GF(2^8)	XOR, cyklický posun, násobení mod (2^16)+1, sčítání mod 2^16	XOR, permutace
slabé klíče	ne	ano	ano

- výkon na 8 bitovém CPU (Motorola 6805), *G. Keating–AES candidates on 6805 core[2]*

Paměťová náročnost (byte)	**AES**	**AES^{-1}**	**IDEA**	**DES**
RAM	50	54	NA	117
ROM šifrování	879	976	NA	680
ROM šifrování + příprava klíče	879	1049	NA	1036

Pozn.: Hodnoty pro 3DES by se od hodnot pro DES měli jen málo lišit.

Čas (cycles)	**AES**	**AES^{-1}**	**IDEA**	**DES**
šifrování	9464	13538	NA	17458
příprava algoritmu	0	2278	NA	12320

Pozn.: Hodnoty pro 3DES by měli být vzhledem k DES zhruba trojnásobné.

- výkon na 32 bitovém CPU (K6-II @350Mhz), instrukční sada 386, kódy [4], [6], [7]

	Rychlost Mbits/s				
(klíč/blok)	**AES (128/128)**	**AES^{-1} (128/128)**	**IDEA (128/64)**	**IDEA^{-1} (128/64)**	**3DES (128/64)**
Příprava klíče (C)	171	28	31	5.8	7.6
Šifrování (C)	80	81	22	22.5	13.4
Příprava klíče (Java)	8.5	5.9	7.4	1.3	1.9
Šifrování (Java)	17.9	17.9	7.5	7.5	3.2

Pozn.: Příprava klíče pro 3DES v Jave je pro 192bitový klíč, pro 128bitový klíč by měla být rychlost přípravy o 1/3 vyšší (Cryptix neumožňuje přípravu 128bitového klíče).

3. Výkon algoritmu AES

- **Testovací konfigurace:** AMD K6-II @350 Mhz, 128MB RAM, Windows 98
  Intel Pentium II @350 Mhz, 128 MB RAM, Windows 98
- **Překladače:** Microsoft Visual C++ 6.0, Sun Java2 SDK 1.3.0_02
- **Instrukční sada:** Výkon algoritmu v jazyce C byl měřen po přeložení pro procesory s instrukční sadou 386 i pro procesory s instrukční sadou Pentium Pro, která poskytuje výrazně rychlejší instrukce pro rotaci bitů.
- **Zdrojové kódy C:**
  AES – reference code [3], AES implementation [4]
  IDEA – Cryptlib [5]
  3DES – Des implementation [6]

- **Zdrojové kódy Java:**
  AES, 3DES, IDEA - Cryptix 3.2.0 [7]
- **Metodika:** Výkon algoritmu byl určen opakovaným voláním vybrané metody (příprava klíče, šifrování) a následným přepočtem výsledné doby vzhledem k počtu opakování. Rychlost operace pak byla určena dle následujícího vzorce:
  *rychlost = počet_opakování*délka_bloku/celkový_čas*
- **Délka bloku:** Algoritmus AES byl testován s délkou bloku 128 bitů (specifikace AES), 192 bitů a 256 bitů. Pokud není uvedeno jinak, byl použit blok s délkou 128 bitů.

Výsledky:

Raw test: V tomto testu není v případě šifrování a dešifrování zohledněna doba nutná pro přípravu klíče (bloky jsou šifrovány jakoby jedním předem připraveným klíčem).

Jazyk C, AES implementation [4], instrukční sada 386, rychlost v Mbits/s

- *128 bitů blok*

	K6-II @350 Mhz			**Pentium II @350 Mhz**		
Délka klíče	**128 bitů**	**192 bitů**	**256 bitů**	**128 bitů**	**192 bitů**	**256 bitů**
Příprava klíče pro zašifrování	171	163	112	129	123	101
Zašifrování	80	68	58	120	101	80
Příprava klíče pro dešifrování	28	23	20	31	27	23
Dešifrování	81	68	57	120	101	80

- *192 bitů blok*

	K6-II @350 Mhz			**Pentium II @350 Mhz**		
Délka klíče	**128 bitů**	**192 bitů**	**256 bitů**	**128 bitů**	**192 bitů**	**256 bitů**
Příprava klíče pro zašifrování	136	174	123	109	146	108
Zašifrování	61	61	53	97	98	85
Příprava klíče pro dešifrování	23	24	20	26	28	23
Dešifrování	62	63	54	98	98	85

- *256 bitů blok*

	K6-II @350 Mhz			**Pentium II @350 Mhz**		
Délka klíče	**128 bitů**	**192 bitů**	**256 bitů**	**128 bitů**	**192 bitů**	**256 bitů**
Příprava klíče pro zašifrování	115	136	117	97	137	119
Zašifrování	51	51	50	83	83	83
Příprava klíče pro dešifrování	20	21	20	22	24	24
Dešifrování	52	51	51	83	83	83

Jazyk C, AES implementation [4], instrukční sada Pentium Pro, rychlost v Mbits/s

- *128 bitů blok*

	K6-II @350 Mhz			**Pentium II @350 Mhz**		
Délka klíče	**128 bitů**	**192 bitů**	**256 bitů**	**128 bitů**	**192 bitů**	**256 bitů**
Příprava klíče pro zašifrování	142	102	88	133	122	102
Zašifrování	80	67	57	116	99	85
Příprava klíče pro dešifrování	27	22	19	31	28	23
Dešifrování	76	64	55	120	101	88

- *192 bitů blok*

	K6-II @350 Mhz			Pentium II @350 Mhz		
Délka klíče	**128 bitů**	**192 bitů**	**256 bitů**	**128 bitů**	**192 bitů**	**256 bitů**
Příprava klíče pro zašifrování	120	109	100	113	145	111
Zašifrování	62	62	54	98	97	83
Příprava klíče pro dešifrování	23	24	20	26	28	23
Dešifrování	60	60	52	99	97	85

- *256 bitů blok*

	K6-II @350 Mhz			Pentium II @350 Mhz		
Délka klíče	**128 bitů**	**192 bitů**	**256 bitů**	**128 bitů**	**192 bitů**	**256 bitů**
Příprava klíče pro zašifrování	104	113	116	99	139	123
Zašifrování	50	51	50	82	82	82
Příprava klíče pro dešifrování	19	20	20	22	24	24
Dešifrování	51	51	50	82	81	82

Jazyk Java, Cryptix 3.2.0 [7], rychlost v Mbits/s

- *128 bitů blok*

	K6-II @350 Mhz			Pentium II @350 Mhz		
Délka klíče	**128 bitů**	**192 bitů**	**256 bitů**	**128 bitů**	**192 bitů**	**256 bitů**
Příprava klíče pro zašifrování	8.5	8.9	5.9	NA	NA	NA
Zašifrování	17.9	16.3	14.9	NA	NA	NA
Příprava klíče pro dešifrování	5.9	5.3	4.1	NA	NA	NA
Dešifrování	17.9	16.3	14.9	NA	NA	NA

Packet test (vliv přípravy klíče na výkon): Tento test zohledňuje vliv přípravy klíče a počtu bloků, pro které bude připravený klíč použit, na výkon šifrování a dešifrování.
(1 blok = 128 bitů, K6-II @350 Mhz (i.s. 386), Pentium II @350 Mhz (i.s. Pentium Pro))

- *128 bitů klíč*

	Rychlost Mbits/s *(v % teoreticky maximální rychlosti)*					
počet bloků	**1 blok**	**2 bloky**	**3 bloky**	**10 bloků**	**100 bloků**	**Max.**
K6-II @350 Mhz						
Zašifrování	55 *(69%)*	66 *(83%)*	71 *(89%)*	78 *(98%)*	79 *(99%)*	80 *(100%)*
Dešifrování	21 *(26%)*	34 *(42%)*	41 *(51%)*	64 *(79%)*	80 *(99%)*	81 *(100%)*
Pentium II @350 Mhz						
Zašifrování	55 *(37%)*	75 *(65%)*	85 *(73%)*	105 *(91%)*	115 *(99%)*	116 *(100%)*
Dešifrování	25 *(21%)*	41 *(34%)*	52 *(43%)*	86 *(72%)*	115 *(96%)*	120 *(100%)*

- *192 bitů klíč*

	Rychlost Mbits/s *(v % teoreticky maximální rychlosti)*					
počet bloků	**1 blok**	**2 bloky**	**3 bloky**	**10 bloků**	**100 bloků**	**Max.**
K6-II @350 Mhz						
Zašifrování	43 *(63%)*	53 *(77%)*	57 *(84%)*	64 *(94%)*	67 *(99%)*	68 *(100%)*
Dešifrování	18 *(26%)*	28 *(41%)*	34 *(51%)*	53 *(77%)*	66 *(97%)*	68 *(100%)*
Pentium II @350 Mhz						
Zašifrování	55 *(56%)*	71 *(72%)*	78 *(80%)*	91 *(93%)*	98 *(100%)*	98 *(100%)*
Dešifrování	22 *(22%)*	36 *(36%)*	45 *(45%)*	74 *(75%)*	98 *(99%)*	99 *(100%)*

- *256 bitů klíč*

	Rychlost Mbits/s ***(v % teoreticky maximální rychlosti)***					
počet bloků	**1 blok**	**2 bloky**	**3 bloky**	**10 bloků**	**100 bloků**	**Max.**
K6-II @350 Mhz						
Zašifrování	35 *(60%)*	43 *(75%)*	48 *(82%)*	55 *(94%)*	57 *(99%)*	58 *(100%)*
Dešifrování	15 *(26%)*	23 *(40%)*	29 *(51%)*	45 *(78%)*	56 *(99%)*	57 *(100%)*
Pentium II @350 Mhz						
Zašifrování	47 *(55%)*	60 *(71%)*	67 *(79%)*	79 *(93%)*	85 *(100%)*	85 *(100%)*
Dešifrování	18 *(20%)*	30 *(34%)*	39 *(44%)*	64 *(73%)*	85 *(97%)*	88 *(100%)*

Table-lookup (T-L) test: Srovnání výkonu algoritmu s předpočtenými tably oproti implementaci bez tabel. Metoda předpočtení tabel je popsána v [1].
(128 bitů blok, jazyk C, K6-II @350 Mhz, i.s. 386)

- výkon bez T-L – *reference code [3]*, výkon s T-L – *AES implementation[4]*

	Rychlost Mbits/s		
délka klíče	**128 bitů**	**192 bitů**	**256 bitů**
zašifrování (bez T-L/ T-L)	1.3 / 80	1.1 / 68	0.9 / 58

Závěr:

- Uvedené testy demonstrují, že algoritmus AES je podstatně rychlejší než algoritmy IDEA a 3DES.
- AES implementation [4] je kód optimalizovaný pro procesory Intel řady Pentium Pro. Oproti AMD K6-II zde Intel Pentium II dosahuje až 50% nárůstu výkonu při operaci šifrování.

Literatura, zdrojové kódy

[1] J. Daemen, V. Rijmen, "AES Proposal: Rijndael"
http://www.esat.kuleuven.ac.be/~rijmen/rijndael/
[2] G. Keating, "Performance Analysis of AES candidates on the 6805 CPU core"
http://www.ozemail.com.au/~geoffk/aes-6805
[3] Paulo Barreto, Vincent Rijmen, "Rijndael ANSI C Reference Code v2.0"
http://www.esat.kuleuven.ac.be/~rijmen/rijndael/
[4] B. Gladman, "AES implementation"
http://www.gladman.uk.net/
[5] Peter Gutmann, "CryptLib v3.0 beta"
http://www.sogot.de/cryptlib/
[6] Eric Young, "Des implementation v4.04"
ftp://ftp.psy.uq.oz.au/pub/Crypto/SSL/
[7] Cryptix 3.2.0
http://www.cryptix.org