

Kryptografický modul šifruje symetrické klíče o délce 128 bitů pomocí schématu RSAES-PKCS1-v1_5 (viz PKCS#1). Vlivem chyby došlo k zašifrování jednoho a téhož (vč. doplňku) 128b klíče dvakrát po sobě pro stejnou hodnotu modulu N a dva různé veřejné exponenty e_1 a e_2 . Tím vznikly dva šifrové texty c_1 a c_2 . Najděte hodnotu přenášeného symetrického klíče, máte-li zadáno N , e_1 , e_2 , c_1 , c_2 :

In[1]:= $n =$

```
118 965 598 285 481 192 129 081 634 483 841 743 956 300 799 488 329 922 815 351 539 010 437 554 089 `.  
318 860 156 025 632 310 440 710 382 693 789 411 013 100 317 752 318 762 982 643 578 653 389 773 `.  
295 480 843 102 759 393 358 161 262 015 134 849 938 906 968 092 326 049 472 737 955 561 082 223 `.  
167 084 692 048 807 385 485 725 618 552 857 193 638 122 060 692 505 467 478 944 805 145 300 591 `.  
376 251 274 273 515 157
```

$e_1 = 2^{16} + 1$

$e_2 = 2^{16} + 3$

$c_1 =$

```
60 816 213 980 760 845 529 676 546 373 691 035 175 401 040 528 348 158 807 195 272 197 637 877 750 `.  
268 815 192 637 665 579 710 465 353 000 124 830 228 342 142 959 747 841 208 633 969 729 286 623 `.  
652 239 721 778 238 171 921 897 294 835 049 218 545 845 737 877 823 335 871 348 762 323 757 325 `.  
297 979 746 201 061 155 760 770 746 669 672 920 423 745 450 026 412 283 339 843 572 857 310 640 `.  
294 435 173 721 835 662
```

$c_2 =$

```
84 767 346 041 250 025 447 719 209 431 665 689 335 068 944 082 385 429 176 256 973 885 135 943 447 `.  
486 083 620 603 665 199 185 292 422 798 119 203 294 136 383 555 178 389 553 236 514 089 629 189 `.  
369 156 853 238 970 935 276 397 921 283 293 937 232 921 105 847 001 399 268 459 493 760 984 368 `.  
056 157 991 024 835 422 512 210 973 658 980 986 792 514 634 625 885 894 463 672 288 737 551 800 `.  
677 117 035 834 947 094
```

Out[1]= 118 965 598 285 481 192 129 081 634 483 841 743 956 300 799 488 329 922 815 351 539 010 437 554 089 `.
318 860 156 025 632 310 440 710 382 693 789 411 013 100 317 752 318 762 982 643 578 653 389 773 295 `.
480 843 102 759 393 358 161 262 015 134 849 938 906 968 092 326 049 472 737 955 561 082 223 167 084 `.
692 048 807 385 485 725 618 552 857 193 638 122 060 692 505 467 478 944 805 145 300 591 376 251 274 `.
273 515 157

Out[2]= 65 537

Out[3]= 65 539

Out[4]= 60 816 213 980 760 845 529 676 546 373 691 035 175 401 040 528 348 158 807 195 272 197 637 877 750 `.
268 815 192 637 665 579 710 465 353 000 124 830 228 342 142 959 747 841 208 633 969 729 286 623 652 `.
239 721 778 238 171 921 897 294 835 049 218 545 845 737 877 823 335 871 348 762 323 757 325 297 979 `.
746 201 061 155 760 770 746 669 672 920 423 745 450 026 412 283 339 843 572 857 310 640 294 435 173 `.
721 835 662

```
Out[5]= 84 767 346 041 250 025 447 719 209 431 665 689 335 068 944 082 385 429 176 256 973 885 135 943 447 `
        486 083 620 603 665 199 185 292 422 798 119 203 294 136 383 555 178 389 553 236 514 089 629 189 369 `
        156 853 238 970 935 276 397 921 283 293 937 232 921 105 847 001 399 268 459 493 760 984 368 056 157 `
        991 024 835 422 512 210 973 658 980 986 792 514 634 625 885 894 463 672 288 737 551 800 677 117 035 `
        834 947 094
```

Víme, že $(c1 = x^{e1} \bmod N)$, $(c2 = x^{e2} \bmod N)$, kde x je nešifrovaná hodnota, kterou potřebujeme zjistit.

```
In[6]:= k1 = ExtendedGCD[e1, e2][[2, 1]]
        k2 = ExtendedGCD[e1, e2][[2, 2]]
```

```
Out[6]= 32 769
```

```
Out[7]= -32 768
```

Díky REA víme, že $(e1 * k1 + e2 * k2 = 1)$. Můžeme tedy odvodit:

$x =$

$x^{e1} =$

$x^{(e1 * k1 + e2 * k2)} =$

$x^{(e1 * k1)} * x^{(e2 * k2)} =$

$(x^{e1})^{k1} * (x^{e2})^{k2} =$

$c1^{k1} * c2^{k2}$

Víme tedy, že $x = c1^{k1} * c2^{k2} \bmod N$. Jelikož je $k2$ záporné, vypočteme si nejprve pro zjednodušení multiplikativní inverzi k $c2$ v mod N :

```
In[10]:= c2Inv = ExtendedGCD[c2, n][[2, 1]];
         x = Mod[c1^k1 * c2Inv^(-k2), n]
```

```
Out[11]= 7 602 152 184 510 448 905 231 255 858 678 485 305 145 097 698 310 457 618 471 911 340 491 239 369 555 `
        251 143 890 284 350 819 330 535 362 626 821 586 063 635 758 771 450 591 288 536 024 261 651 464 279 `
        214 775 838 547 608 629 951 723 396 819 932 031 019 611 722 843 885 502 098 531 371 328 612 266 801 `
        335 041 852 969 282 766 921 754 649 835 655 191 659 267 740 176 922 537 183 340 520 680 102 383 130 `
        076
```

Pro klid naší duše můžeme provést kontrolu:

```
In[12]:= Mod[x^e1, n] == c1
        Mod[x^e2, n] == c2
```

```
Out[12]= True
```

```
Out[13]= True
```

Jelikož víme, že klíč je šifrován pomocí standardu PKCS#1, můžeme postupovat

podle návodu na získání původního octet stringu. Nejprve vytvoříme funkci I2OSP, která převede číslo x do EM (encoded message) skládajícího se z jednotlivých oktetů. EM je ve formátu $EM = 0x00 \parallel 0x02 \parallel PS \parallel 0x00 \parallel M$, ze kterého získáme původní octet string M :

```
In[18]:= i2osp[number_] :=
  Module[{octets = {}}, xx = number; While[xx > 0, AppendTo[octets, Mod[xx, 256]];
    xx = IntegerPart[xx / 256]];
  Reverse[AppendTo[octets, 0]]];
octetsList = i2osp[x];
sequences = SequenceSplit[octetsList, {0}];
m = Last[sequences]

Out[21]= {197, 161, 156, 20, 184, 11, 233, 127, 21, 112, 51, 206, 30, 89, 21, 220}
```

Pozn.: Pokud by EM obsahovalo více bajtů 0x00, bylo by potřeba vyzkoušet všechny kombince!