Nick Palutsis
February 22, 2018
Computer Security

## Homework 3

### Task 1.2

```
f5e8 2deb 6e9f 1fcd ffb5 ecd3 ca00 a9c9
a8ed a39f 5c58 1248 2b2b 07dc 752c a516
8217 f03f 3649 ea79 21bc 428b c69d f671
186d c9a1 a932 d052 d310 b81b 2007 e485
a885 022d cada 5267 9519 c407 a328 af20
45da 3ea1 45e9 0060 e0b5 975b b78a aab4
83bf 0018 8441 434a 58d2 075d 9bee ccf6
5102 0325 de3d 89d6 97b3 e491 8272 c3df
98b9 2625 1790 3f7b 83c2 96d5 e93c 57b6
5e08 341a 2cae 3c2e a761 f72b 4022 1a44
eed2 a621 ed93 6977 95b5 eb90 90d0 db42
59db ef41 7f7f 0e0f 403f 7f0a 8c9d 272b
3890 6c31 8973 2b59 7f13 3026 86f3 3c2b
b875 4691 4896 7a99 44b6 310c 4be9 2aa2
7601 fa49 9bad dcf4 64d3 3fc8 de92 fa77
94b6 08b0 250c 4ce0 62ce 4bb1 5eff 3365
37c0 7b18 f1b9 11dc 34aa d76b 8fa0 2ff6
0dd2 26cf 79d6 f5b9 18fc c68d 7dc7 676d
844d 67fb bf37 7d8d a357 59f3 27ac 86b2
a1a7 9dd3 e36a 53a3 9cf2 9a0a 87e3 639c
1923 c46b 349c 89b6 dd5d 11b7 6e7f 8491
6f21 6b88 815a bdd7 7fe5 7c82 d026 4041
c17f 7114 f68f 7106 816a 2f9f 185f e5a8
468c f942 3f29 86a1 607a 9da6 2d72 601b
de78 2c92 4773 0eda f6c4 338a 9726 0982
8bf2 8225 d4e5 7b67 aa5d 44ad cc30 3386
b270 82db 081b f129 4616 4c27 f50f e92d
02d6 d6ca 6b43 a125 83e2 96ac f1f0 d921
57bd ae1b 208d 370a e484 dea6 08dc 7c7d
9edb ed99 4a5d 45
```

### Task 1.3

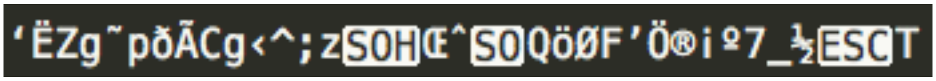
```
The world doesn't live off jam and fancy perfumes - it lives off bread
and meat and potatoes. Nothing changes. All the big fancy stuff is
sloppy stuff that crashes. I don't need dancing baloney - I need stuff
that works. That's not as pretty, and just as hard.

   -Theo de Raadt
```

### Question 2

For symmetric key encryption, Sodium opts for the xSalsa20 algorithm. xSalsa20 is a stream cipher instead of the typical block cipher. Since stream ciphers encrypt each value individually, it is less likely that a failure will cause the loss of significant amounts of information; whereas, block ciphers encrypt entire blocks of data at a time, so a failure can cause the loss of an entire block. xSalsa20 also has the advantage of being faster and requiring less memory than a block cipher like AES.

**Task 3.3**

Public:



```
‘ËZg˜pðÃCg‹^;zSOHŒˆSOQöØF’Ö®iº7_½ESCT
```

Ciphertext:

```
8267 ebe8 dcfd 026b 70fb 8c8b b8b8 c4ef
1f21 bd5a faa5 dbb2 703b 39a6 7fbc 1bc0
8b7e 480f d738 96bf c500 e25e 2b61 e4a5
a929 c353 4f71 3614 fc8d 4df6 0350 2121
79da fc62 408c 1fff 01fc d2ab d504 2b4e
1e64 8aa7 f0c9 d0f3 f29b 3d3e 19e4 d83c
2325 1540 7a01 a0cf 65a4 bff7 0d88 f311
1ddd fa09 532c a856 69fa b7c7 9384 1147
2c14 a289 2595 a3b1 eac7 f3cd 92fc 7e54
1b43 5ee4 66cb b389 36b6 2a66 51e2 8a1d
9b1b bdc8 33bf 9ee3 8ae1 a756 5960 f012
ebc2 473f f2aa d65d e617 39df 97da 3c1d
498d 4cf2 30cd f33b 78a1 b39a 5fc1 af00
187c ab13 5bdd 6eb9 f9fd bac3 610e 84d8
d2ba 5f11 5b7a 5a80 31cc 9b2c 213b 3f66
08cc 5d2e b7c5 31f4 70c3 609c 2acf 66bf
3847 25b9 05cb 96dd eab8 ac2d e1fb b13d
d184 f50e 3342 cdb1 5205 87de c148 c4db
549c 4011 ab5b ae45 6a44 6089 c046 fe47
e8a4 a772 f5ea 7b2d 0acb f39f be91 3b63
64a0 d845 e2ac 408d 5267 4615 8cff 33d8
6774 61a5 817a 39dc f6cc f357 10fd 26e5
0dd8 8002 675f 590a c648 25db da42 2c96
fbe6 ffa6 2417 d32f 9e86 c9c2 ab86 bd7d
b7df e1be 7114 0c23 0ee2 f335 fe65 39f2
d827 2e36 178e 2082 38f1 c497 22d4 b02c
c1e1 b286 4705 d7eb ba2d edf8 33e9 fea3
ebca d5ec b9f3 dcc4 dfa5 5208 564d 39f4
7915 265d d105 0238 710e df44 f235 d66b
2f1a e6a7 e2eb ac
```

**Task 3.4**

```
d738 e0c8 c640 f104 abeb 39a9 2ec4 87fc
cb41 c65a 933d f4fb 9d1b f2ad acce ad3b
2b9f 4904 b163 97c2 5b17 2301 d830 4c18
8769 9327 3c4e 796e 10cb 85bb a85f 7f78
ba17 395b 3846 d564 869a 39d6 a26c 1ab3
9699 cd4c 67d8 f8e1 65e1 8b72 7676 31c9
6d9b 0537 2c27 ed31 a18c 8593 613d 3d51
3a57 5e20 5d82 2500 6f07 6ffb eee8 50d0
9cee 5f45 2267 9727 c28b 2e87 88c3 81e6
ee1a 7bfd 9c86 cc81 8e57 82b7 e05f b539
7d8a 25f3 dfac 1235 03e0 2e61 cb47 0d74
bacb 7be2 afc4 b731 de96 9877 010a be16
9c13 75e7 039f ee87 d2c5 95f2 1a1a 8487
6023 5e64 2632 dca1 a57a 9291 7b6e b71b
5af5 c205 c773 9a88 c9c9 95da 1d91 f6b5
7918 0b1d 9fd6 abb0 9fdb a739 bbca 4114
386b 3a7f e262 2393 dfa1 b40f df04 d437
6b1e 61f3 2023 cf4b 5ade a4a8 9e7d f4f9
4968 97a2 889d 1d21 f37a b276 94f6 af08
b9c1 9537 62b0 5308 b69b fc47 7f0b 170e
45de 750d c929 51b7 5117 a1a9 77d0 263a
b5bf 6b46 2c70 1fe9 8985 42e5 89a6 de41
5c1c 5fdc 095d c1d4 b45c 1263 4c49 de4d
dced 1986 9761 a4f0 6ac0 0173 0e49 8bb7
0bc6 5dfb 10b0 9612 a734 1f1c 2a37 6749
26e9 6a2a 2679 fd7e 9cb6 b12d 0612 5cb5
4593 02c1 45c0 ce76 c89e 6242 98e8 e000
dc3e c9ab 94d7 d754 0ecb bcaa 34b0 13e3
cc9c 3ac2 a767 9610 2e32 8c0b 493b fb49
fd91 d37a e368 34
```

## Task 3.5

…

## Question 4

The Sodium API uses Curve 25519 to handle the key exchanges. This is better than RSA because RSA requires larger keys in order to maintain security since the keys are based off of prime numbers, and prime numbers become increasingly easier to factor. Larger keys ultimately lead to slower performance.