Information Security Assignment 02

Yanan Xiao
abraham.x91@gmail.com

October 26, 2013

1 Public Key Crypto

1.1 Que. 6

For this question I used an open source code by [1].

- For part a, run *crypt(19,33,3)* and we will get $C = 28$. For decryption, run *crypt(28,33,7)* and we will get the decrypted message $M = 19$. This is applicable since both encryption and decryption are exponential modulus.

- For digital signature, it is done by sign with one's private key, and verify with public key (PK). So run *crypt(25, 7, 3)* and we will get $S = 31$. The verification is done by *crypt(31, 33, 3)*, after this, we obtain the authentic message, i.e. $M = 25$.

1.2 Que. 8

For now, researchers have discovered two kinds of cube root attack. They are [2]:

- Type 1: If the plaintext M satisfies $M < N^{\frac{1}{3}}$, then $C = M^e = M^3$. In this situation, the mod N has no effect.

- Type 2: If the same message M is encrypted with three (or more) different users' public keys, then the Chinese Remainder Theorem can be used to recover the message.

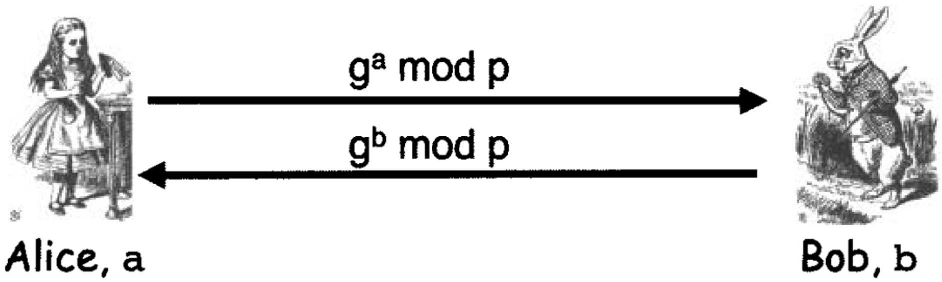


Figure 4.1: Diffie-Hellman Key Exchange

Figure 1: A Popular Key Exchange System

As a real educational example, if we use $(N, e) = (33, 3)$ and $d = 7$, when the $M = 3$, we compute *crypt(3, 33, 3)* and get encrypted message 27. Since public key pairs are "public", when we get the (N, e) pair, we have no trouble doing a recomputation and recover M. When $M = 4$, encrypted message is 31, greater than $N^{\frac{1}{3}}$. So if Trudy wants to recover, he has to do factoring.

1.3 Que. 11

In Diffie-Hellman (DH) Key Exchange algorithm, it is important for both sides of communication, i.e. Alice and Bob, to keep a and b as private as possible. On the other hand, the prime p and generator g are public.

1.4 Que. 12

In textbook [2], the author mentioned 3 methods to prevent man-in-the-middle attack on DH, I will illustrate encrypting with public keys. For this enhanced way, both sides will do the following stuff, assumption is that an effective network communication has or will have been set.

- Alice calculates $g^a\ mod\ p$ and encrypts this with Bob's one public key available.
- Alice sends the message to Bob.
- Bob calculates $g^b\ mod\ p$ and encrypts this with Alice's one public key available.

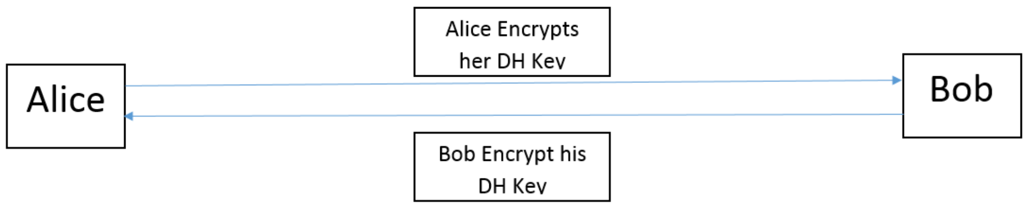


Figure 2: Encrypt the DH exchange with public keys

- Bob sends the message to Alice.
- Both sides decrypt the other's message with his/her own respective private key.

Due to the assumption that Trudy has no idea of Alice and Bob's private keys, this way works.

1.5 Que. 15

In this question, MAC stands for a less popular acronym Message Authentication Code rather than Media Access Control in the networking field. For MAC, it "uses a block cipher to ensure data integrity". As often the case with block cipher, which is a subset of Symmetric Key Cryptography, the repudiation always exists because the *authentication and verification* use the same key. So it is fairly easy for either side of the communication to repudiate some messages.

However, as often the case in asymmetric key crptosystem, we make such assumption that the sender digitally signs his/her message, with the private key that is *taken good care of.* In other words, only one side of the communication knows (owns) the private key, which is then used in digital signature.

1.6 Que. 16

With this question, I will have a discussion on the "entangled" relationship in everyday use of symmetric and public cryptography system.

For a hybrid system using Diffie-Hellman as public key system and DES as symmetric cipher.

- Diffie-Hellman is no more than a key exchange system. In this senario, DH will be used to generate keys for DES, and transmit it in a DH way.

- It is obviously showed in 1. Alice sends Bob $g^a \ mod \ p$, and Bob sends Alice $g^b \ mod \ p$. The key they would use in DES then is $g^{ab} \ mod \ p$.

- After a shared, and to some extend secure key is constructed, the message then could easily, and more speedy (that's why we use then in reality) encrypted. After encryption, it's all about transmission, which we will not discuss for now.

For a hybrid crypto system using RSA as the public key system and AES as symmetric cipher.

- It is fairly easy to conclude that once the RSA is not used to "encrypt" something, then it's task would be to digitally sign something. A digital signature is like a handwritten signature—only more so. Bob is the only one who can digitally sign as Bob, since he is the only one with access to his private key.

- As described, Bob generates a symmetric key for AES encryption, and digitally sign his AES key wit his RSA private key. After doing these, Bob sends Alice the encrypted message, as well as a key signed by himself.

- Alice uses Bob's public key, which she may already have, or obtain from a public key infrastructure. If and only if Alice finds the key is truly signed by Bob, she will star decrypting messages using the (signed) key.

1.7 Que. 22

It's a very textbook question. As with the public key $(18, 30, 7, 26)$, I wrote a MATLAB program. The answer is as follows.

- Run *SimpleKnapsack* and we will get private key, $(14, 39, 42, 15)$.

- In the output, another hint we find is ciphertext 74 as in decimal.

1.8 Que. 26

The discussion is given below.

- When $g = 1$, it would become meaningless. For one thing, no matter what value a takes, $g^a \equiv 1$. Moreover, it is known to all that $1 \; mod \; p \equiv 1$ no matter what value p takes. In short, the key Alice and Bob exchange would always be 1.

- If g takes the value $p-1$, since p and $p-1$ are relative prime, according to Euler's Theorem, we can easily find that no matter what value a takes, $(p-1)^a \; mod \; p \equiv 1$. Therefore, it's unsafe as with the senario discussed above. It's allowable on condition that we do not consider security as an isssue.

2 Hash Functions++

In this section, we solve some hash problems.

2.1 Que. 3

Here is a short description of how brute force attack may be conducted.

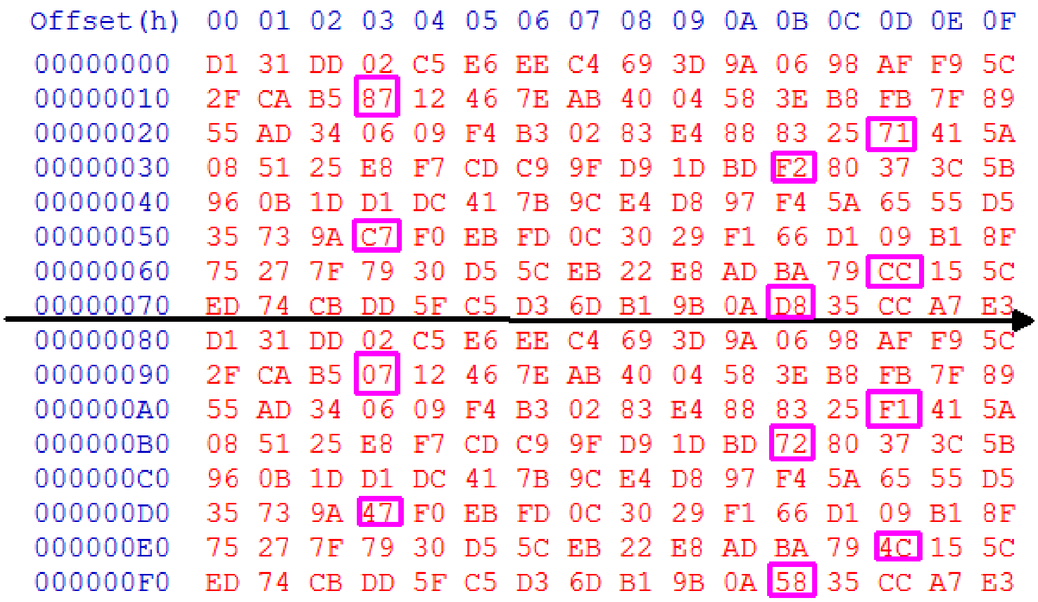
- Case 1: Suppose Trudy does not have a database which stores pre-computed entries of each and every hash value of this hash function.

- Trudy has to hash each and every possible values, and compare with the target hash value, let's say T_{hash}. The total computational complexity would be $O(hashing) + O(comparing)$. (which varies according to hashing complexity, therefore non-linear.)

- Case 2: Suppose Trudy already has a hashing value database, then all he has to do is to compare n times. The computational complexity is $O(n)$.

2.2 Que. 4

The essence of hashing is mapping [3]. When keeping this in mind, the explanation would come easily.



Offset(h)	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
00000000	D1	31	DD	02	C5	E6	EE	C4	69	3D	9A	06	98	AF	F9	5C
00000010	2F	CA	B5	87	12	46	7E	AB	40	04	58	3E	B8	FB	7F	89
00000020	55	AD	34	06	09	F4	B3	02	83	E4	88	83	25	71	41	5A
00000030	08	51	25	E8	F7	CD	C9	9F	D9	1D	BD	F2	80	37	3C	5B
00000040	96	0B	1D	D1	DC	41	7B	9C	E4	D8	97	F4	5A	65	55	D5
00000050	35	73	9A	C7	F0	EB	FD	0C	30	29	F1	66	D1	09	B1	8F
00000060	75	27	7F	79	30	D5	5C	EB	22	E8	AD	BA	79	CC	15	5C
00000070	ED	74	CB	DD	5F	C5	D3	6D	B1	9B	0A	D8	35	CC	A7	E3
00000080	D1	31	DD	02	C5	E6	EE	C4	69	3D	9A	06	98	AF	F9	5C
00000090	2F	CA	B5	07	12	46	7E	AB	40	04	58	3E	B8	FB	7F	89
000000A0	55	AD	34	06	09	F4	B3	02	83	E4	88	83	25	F1	41	5A
000000B0	08	51	25	E8	F7	CD	C9	9F	D9	1D	BD	72	80	37	3C	5B
000000C0	96	0B	1D	D1	DC	41	7B	9C	E4	D8	97	F4	5A	65	55	D5
000000D0	35	73	9A	47	F0	EB	FD	0C	30	29	F1	66	D1	09	B1	8F
000000E0	75	27	7F	79	30	D5	5C	EB	22	E8	AD	BA	79	4C	15	5C
000000F0	ED	74	CB	DD	5F	C5	D3	6D	B1	9B	0A	58	35	CC	A7	E3

Figure 3: Tagged Differences Between Two Hex Files

- Since the output would be 12-bit, it can hold as many as 2^{12}, namely 4096 completely different messages. So we do not expect collision.
- The formula would be: $Number\ of\ collision = 2^{m-n}$.

2.3 Que. 24

I have enlisted my detailed results in *HashResults.txt*. Please refer to that file for more information. Below I take some screen shots

2.4 Que. 42

Mom told me a picture worths a thousand words. Thereafter, the pictures below will do.

2.5 Que. 48

In this explanation we discuss the usage of random numbers in cryptography.

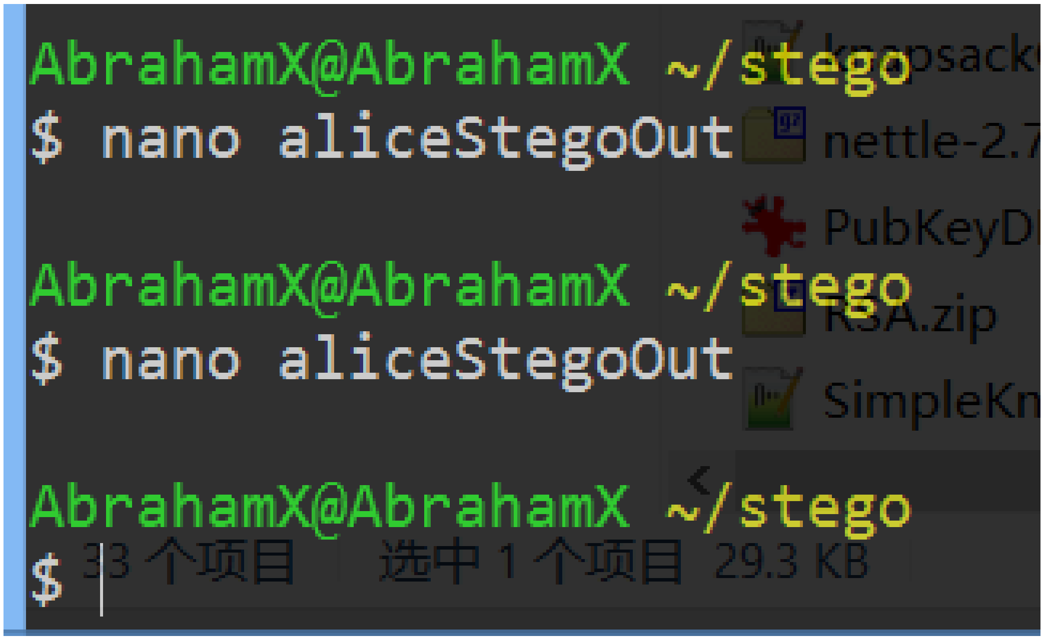


Figure 4: Use Nano Command to Read File without Suffix

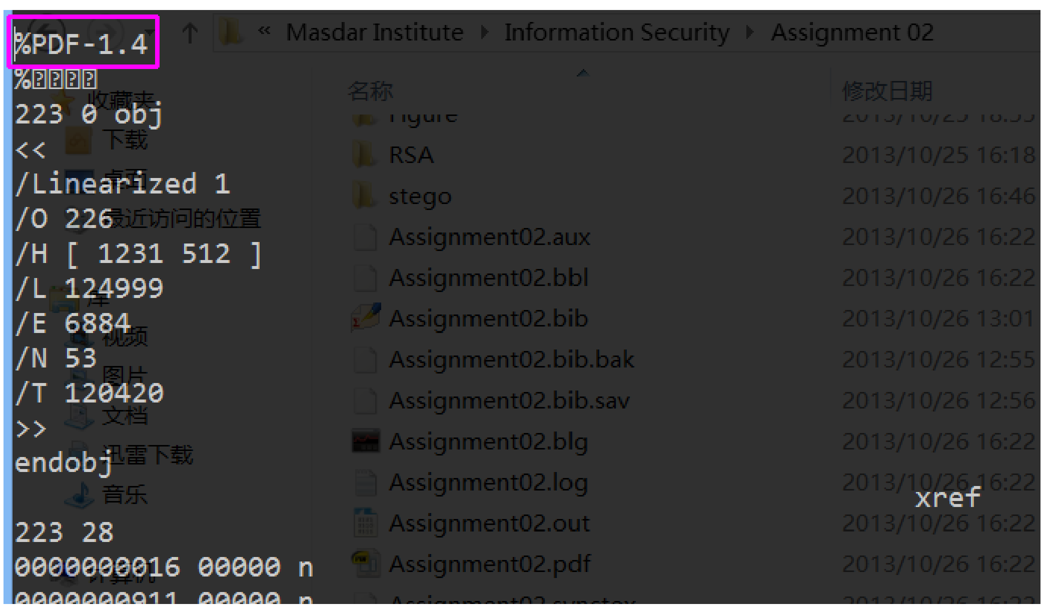


Figure 5: PDF File Header of AliceStegoOut

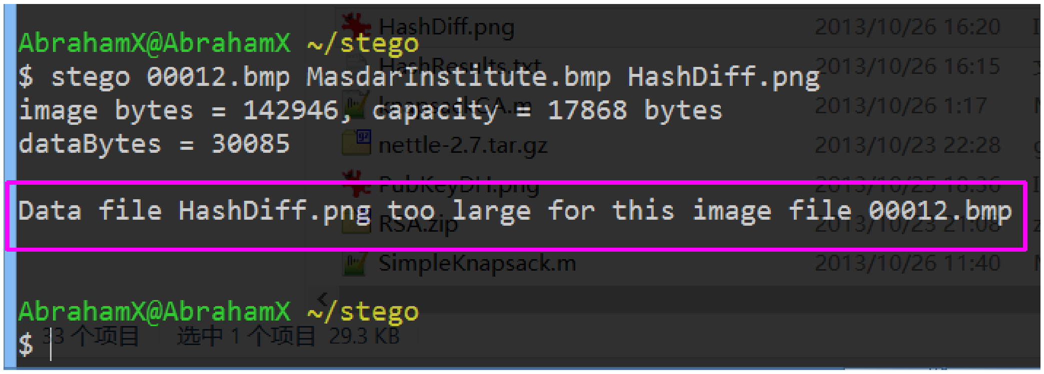


Figure 6: File Too Large to Be Written into Image Empty Space

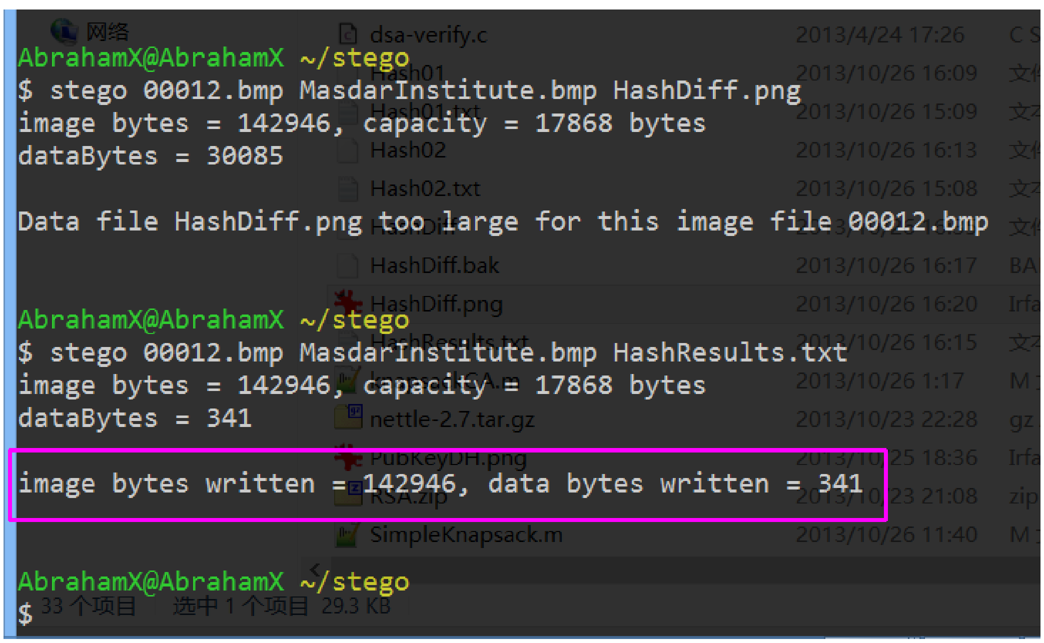


Figure 7: A File Successfully Written into Image Empty Space

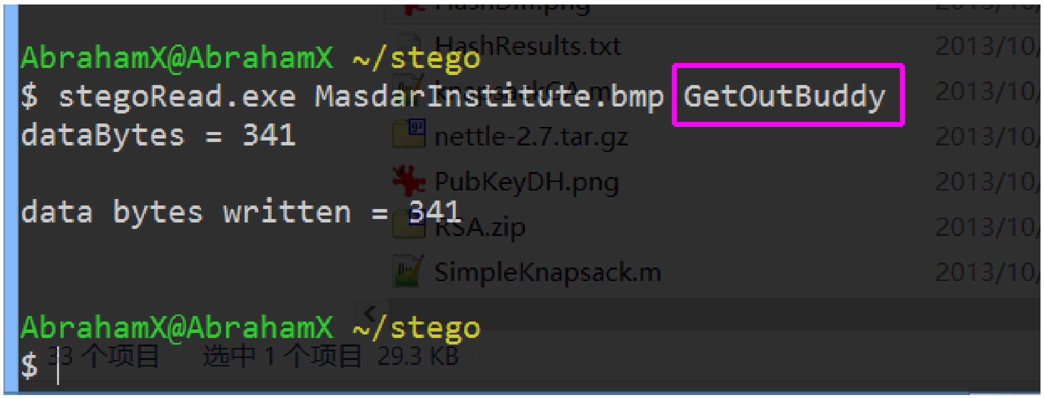


Figure 8: A File Successfully Read Out as A Validation

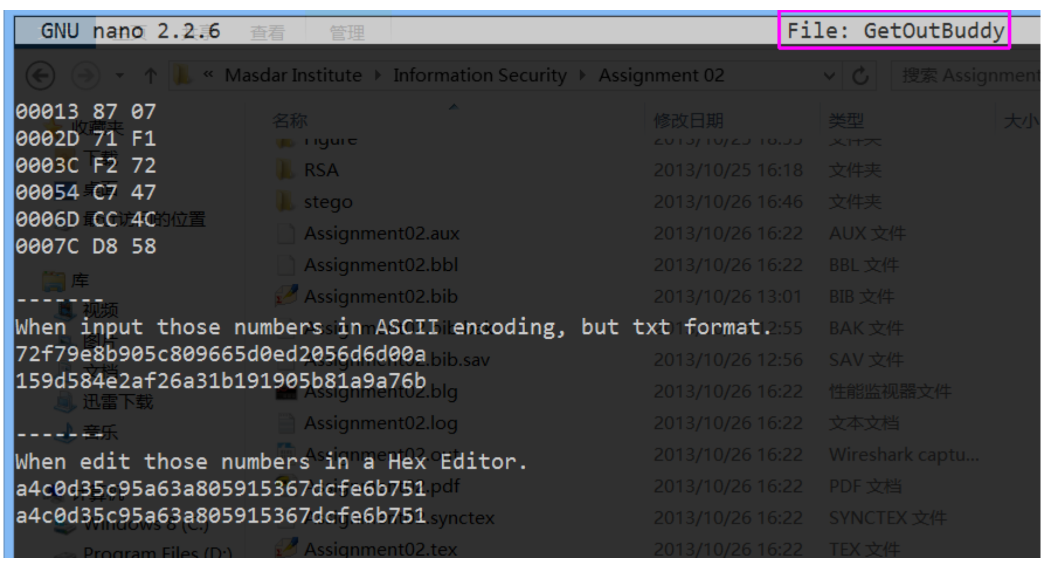
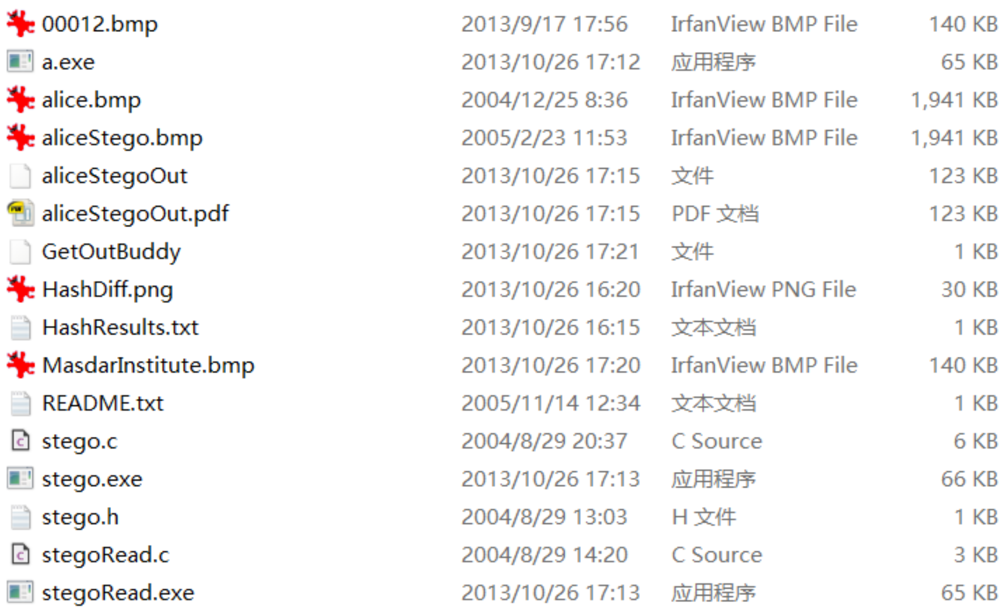


Figure 9: Use Nano Command to Verify the Integrity of File Read Out



00012.bmp	2013/9/17 17:56	IrfanView BMP File	140 KB
a.exe	2013/10/26 17:12	应用程序	65 KB
alice.bmp	2004/12/25 8:36	IrfanView BMP File	1,941 KB
aliceStego.bmp	2005/2/23 11:53	IrfanView BMP File	1,941 KB
aliceStegoOut	2013/10/26 17:15	文件	123 KB
aliceStegoOut.pdf	2013/10/26 17:15	PDF 文档	123 KB
GetOutBuddy	2013/10/26 17:21	文件	1 KB
HashDiff.png	2013/10/26 16:20	IrfanView PNG File	30 KB
HashResults.txt	2013/10/26 16:15	文本文档	1 KB
MasdarInstitute.bmp	2013/10/26 17:20	IrfanView BMP File	140 KB
README.txt	2005/11/14 12:34	文本文档	1 KB
stego.c	2004/8/29 20:37	C Source	6 KB
stego.exe	2013/10/26 17:13	应用程序	66 KB
stego.h	2004/8/29 13:03	H 文件	1 KB
stegoRead.c	2004/8/29 14:20	C Source	3 KB
stegoRead.exe	2013/10/26 17:13	应用程序	65 KB

Figure 10: Thank God It's Midterm!

- As testified by the textbook [2] and some related RFCs, in symmetric key crypto, random numbers are used to generate key pairs. Take DES as an example, pseudo random numbers are generated via S-box.

- In RSA, it is vital that the two prime numbers are as large and random as possible. So random number generator comes to rescue. Case is similar with Diffie-Hellman key exchange algorithm, where the secrete exponents, i.e. the $a$and$b$ in g^a $mod\ p$, g^b $mod\ p$ are generated pseudo randomly.

References

[1] Shaun Gomez. Implementation of rsa algorithm. Online: MATLAB Central, October 2012.

[2] Mark Stamp. *Information Security, Principles and Practice*. Wiley, second edition, 2011.

[3] Wikipedia. Hash function. Online: `https://en.wikipedia.org/wiki/Hash_function`, 2013.