Encrypted transmission of a PGP Public Key to destinations

Peiman Ghasemi
(ORCiD: 0000-0003-0579-8966)
An honorary advisor to the US President; An asylum seeker (refugee)

August 13, 2015

Abstract

To protect your private information, you may use a data encryption and decryption computer program like PGP. But for an espionage agency even the PGP Public Key is not completely unbreakable. So you may prefer to encipher the Public Key before you send it to the destination, then it would become probably an impossible goal for the Internet fraud operatives to decipher the contents.

1. Introduction Transmission of highly sensitive and confidential information and data over the Internet always was a major problem for those who want to protect their private information. After more than 12 years of cooperation with the Central Intelligence Agency,[1] the necessity of writing a brief article, for secure transmission of data, was sensible to me at the moment.

It won't take a long time for an espionage agency to break the code of your ciphered texts. Inside the public documents (and not inside our recent documents) you might see that it could take a few weeks for the NSA (the National Security Agency) to break the code of the texts which are encrypted by a 32 bit or 64 bit PGP Key[2] but what if you encrypt the PGP Public Key

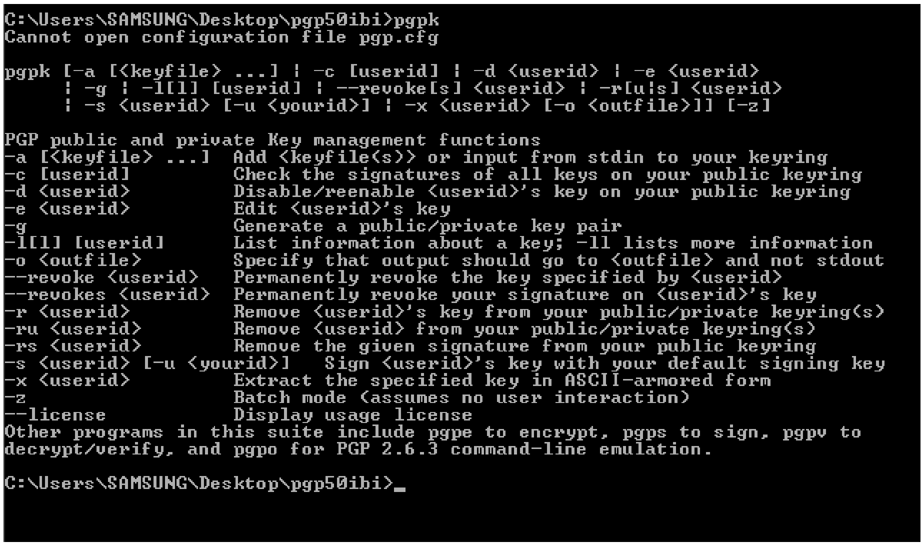
[*] ***Peiman Ghasemi: http://defensetech.military.com/profile/member-profile.html?member_id=33530522***
Email address: peiman.ghasemi@aol.com

[1] and following to the contents of my books about online communications security, privacy, and ciphering; and also since I wrote some articles on different websites: www.ezinearticles.com/?expert=Peiman_Ghasemi
The book "Kings' dirty operation: Concise memos of my cooperation with the CIA and Illuminati's hell" that shall come to markets in last months of 2015, about my cooperation with the US President, the Central Intelligence Agency, etc. (www.amazon.com/author/peiman) and while every days and every weeks I had the opportunity to talk with the Mr. President via some methods of communication (communication technologies relating to the Apollo Soyuz, etc.), and while I was under an extremely high amount of mental stress during my cooperation with the international governmental adversaries

[2] www.tomsguide.com/us/encryption-nsa-edward-snowden-rsa-ssl,news-17503.html

itself,[3] and/or even once again after encryption by PGP (or before encryption by PGP) you encrypt the body of your message!

2. Pretty Good Privacy If you send your Pretty Good Privacy (PGP)[4][5] Private Key with your text message then at the moment the espionage agency may decipher the text, but the PGP has a solution for this problem. The PGP[1][2] can make a Public Key for the destination (and a Private Key for the owner of the password and key) and then you may encrypt the message by the Public Key of the destination. When they receive your cipher encrypted message then they decrypt the cipher by their own Private Key and password. If you encrypt a Public Key of the Pretty Good Privacy (PGP) itself, before you send it (before you send the Public Key) to the target that would encrypt his/her words with your key, then the deciphering process would need spending a long time for the espionage agency, or even it would be impossible to decipher the contents.



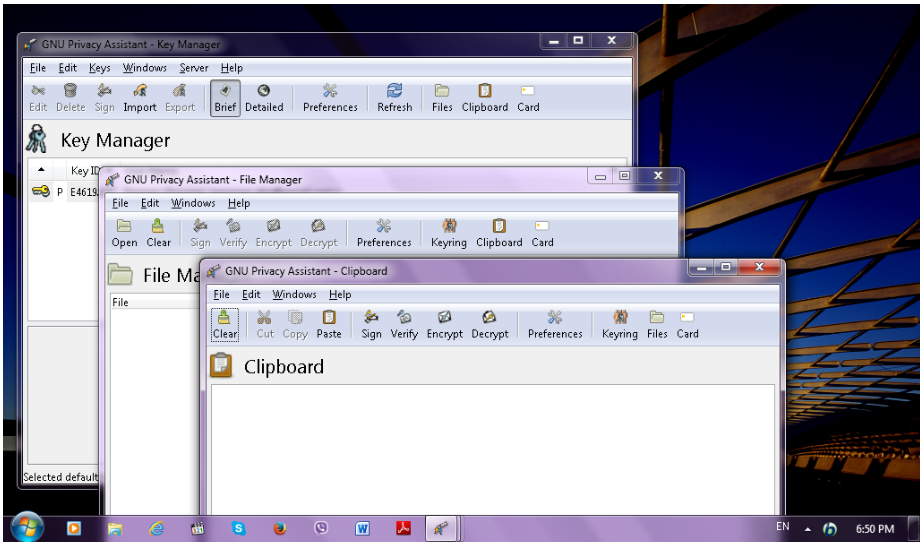
```
C:\Users\SAMSUNG\Desktop\pgp50ibi>pgpk
Cannot open configuration file pgp.cfg

pgpk [-a [<keyfile> ...] | -c [userid] | -d <userid> | -e <userid>
     | -g | -l[l] [userid] | --revoke[s] <userid> | -r[u|s] <userid>
     | -s <userid> [-u <yourid>] | -x <userid> [-o <outfile>]] [-z]

PGP public and private Key management functions
-a [<keyfile> ...]  Add <keyfile(s)> or input from stdin to your keyring
-c [userid]         Check the signatures of all keys on your public keyring
-d <userid>         Disable/reenable <userid>'s key on your public keyring
-e <userid>         Edit <userid>'s key
-g                  Generate a public/private key pair
-l[l] [userid]      List information about a key; -ll lists more information
-o <outfile>        Specify that output should go to <outfile> and not stdout
--revoke <userid>   Permanently revoke the key specified by <userid>
--revokes <userid>  Permanently revoke your signature on <userid>'s key
-r <userid>         Remove <userid>'s key from your public/private keyring(s)
-ru <userid>        Remove <userid> from your public/private keyring(s)
-rs <userid>        Remove the given signature from your public keyring
-s <userid> [-u <yourid>]   Sign <userid>'s key with your default signing key
-x <userid>         Extract the specified key in ASCII-armored form
-z                  Batch mode (assumes no user interaction)
--license           Display usage license
Other programs in this suite include pgpe to encrypt, pgps to sign, pgpv to
decrypt/verify, and pgpo for PGP 2.6.3 command-line emulation.

C:\Users\SAMSUNG\Desktop\pgp50ibi>_
```

GnuPG/PGPi, can work over Command Prompt

[3] "The security of encrypted data is entirely dependent on two things: the strength of the cryptographic algorithm and the secrecy of the key." An Introduction to Cryptography, Page 12

[4] You can download the PGP, here: www.pgpi.org

[5] You can download the GPG4Win, a different edition of PGP, which is especially made for Microsoft Windows, here: www.gpg4win.org


GNU Privacy Assistant - Key Manager
File Edit Keys Windows Server Help
Edit Delete Sign Import Export Brief Detailed Preferences Refresh Files Clipboard Card
Key Manager
Key ID
P E4619
Selected default
GNU Privacy Assistant - File Manager
File Edit Windows Help
Open Clear Sign Verify Encrypt Decrypt Preferences Keyring Clipboard Card
File Ma
File
GNU Privacy Assistant - Clipboard
File Edit Windows Help
Clear Cut Copy Paste Sign Verify Encrypt Decrypt Preferences Keyring Files Card
Clipboard
EN
6:50 PM

Gpg4win works over Windows

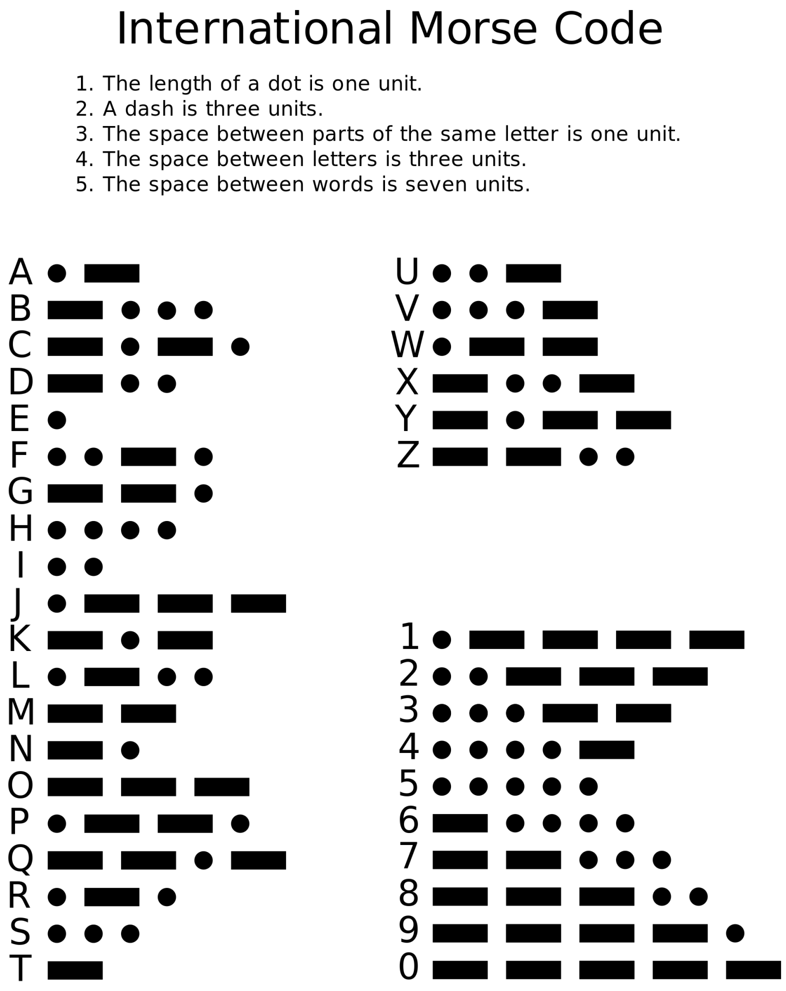
3. Morse code There are some methods by using Morse code[3] available for you to encrypt your texts. When you use the Morse code prosigns then you, freely, can replace the prosigns with your own favorite words, for example instead of “..” you can type “for” etc., instead of “.-” you can type “zero” etc. and so on. So instead of .. .- you would write: “for zero”. The words which you used for ciphering must be known for both of you (the person at the destination who wants to decrypt the ciphered texts later, and you). For example you may like to write to the destination “Instead of every “...” I typed the name of your grandfather’s favorite food” You may like to write this sentence a bit literary sophisticated (For example he has a tattoo on his hand with 3 big dots), you remind him a memo or an accident, you write for him “There is not only one shape on your hand, I saw the name of your grandfather’s favorite thing in the business of the numbers” you remind him the name of his grandfather’s favorite food. For example the name of his grandfather’s favorite food is lasagna, inside some part of your message he would see “for lasagna zero”, he will write “.. ... .-”, when he converts these Morse code prosigns it means “ISA”. If you don’t notify him to replace the entire “for”(s) with “..”(s), and the entire “zero”(s) with “.-”(s) then he may write “.... ... .” because you mentioned “business of the numbers” so since he has a tattoo on his hand with 3 big dots, he may try to replace the word “for” with four dots, etc. So you are responsible to give him firm and obvious notifications; abstractive notifications are useless. Since your data are not invisible to intelligence adversaries (Whether you use a normal protocol for connection to the server or not, for even Secure Socket Layer (SSL)[6] over HTTP, that we call HTTPS, because some advancements (the obvious and introductory

[6] www.digicert.com/ssl.htm

example is proxy server[7]) of the country of the user of the Internet network, and also since it must pass the international Internet backbone[8]), encryption would be an important issue to protect your private information. When you send your PGP Private Key with your text messages, an espionage agency may decipher your data. The PGP makes a Public Key for the destination, as a solution for this problem. When we make an intersection between the logic of the manual encryption/ciphering, in combination with an automatic encryption method through using a (at least) 128 bit key which is made by the PGP encryption algoritms, by this complex method, cipher decryption would be such an unachivable target.

[7] whatis.techtarget.com/definition/proxy-server

[8] global.britannica.com/EBchecked/topic/746032/Internet-service-provider-ISP



International Morse Code

1. The length of a dot is one unit.
2. A dash is three units.
3. The space between parts of the same letter is one unit.
4. The space between letters is three units.
5. The space between words is seven units.

A	● ▬	U	● ● ▬
B	▬ ● ● ●	V	● ● ● ▬
C	▬ ● ▬ ●	W	● ▬ ▬
D	▬ ● ●	X	▬ ● ● ▬
E	●	Y	▬ ● ▬ ▬
F	● ● ▬ ●	Z	▬ ▬ ● ●
G	▬ ▬ ●		
H	● ● ● ●		
I	● ●		
J	● ▬ ▬ ▬		
K	▬ ● ▬	1	● ▬ ▬ ▬ ▬
L	● ▬ ● ●	2	● ● ▬ ▬ ▬
M	▬ ▬	3	● ● ● ▬ ▬
N	▬ ●	4	● ● ● ● ▬
O	▬ ▬ ▬	5	● ● ● ● ●
P	● ▬ ▬ ●	6	▬ ● ● ● ●
Q	▬ ▬ ● ▬	7	▬ ▬ ● ● ●
R	● ▬ ●	8	▬ ▬ ▬ ● ●
S	● ● ●	9	▬ ▬ ▬ ▬ ●
T	▬	0	▬ ▬ ▬ ▬ ▬

International Morse code

References

[1] Philip R. Zimmermann, *The Official PGP User's Guide* (1996)

[2] Michael W. Lucas, *PGP & GPG: Email for the Practical Paranoid* (2006)

[3] Dave Finley, *Morse Code: Breaking the Barrier* (1998)