A brief analysis for Pretty Good Privacy

ABSTRACT

The goal of this paper is to demonstrate how the Pretty Good Privacy works from theoretical aspects and to investigate the reliability of Pretty Good Privacy in practical ways. Some important algorithms will be discussed, including Hash function, DES, RSA, MD5. Then, attacks aiming Pretty Good Privacy implementing will be displayed, in order to proof whether it has durable reliability.

1. INTRODUCTION

As the Internet technology and applications are becoming much more accessible than ever before, the population using the Internet has been growing exponentially these years. Because of none set-up costs for Email using and Email's asynchrony (in other words, people connecting each other do not bother to exchange messages in the arranged time), Email has become one of the most quick, convenient, and economical communication styles. Meanwhile, the security issues about Email are getting obvious. In fact, the delivery process of Email messages is repeated on the network replication procedure. Consequently, it is easy for unauthorized people to theft, tamper, or even damage Email messages, in terms of the uncertainty of network transmissions.

Therefore, there is an urgent need for public users to encrypt messages and ensure the security of transmission through the

Internet.

Under this circumstance, PGP (Pretty Good Privacy)[1] has emerged. Pretty Good Privacy is a program that uses encryption to protect the privacy of your electronic mail and the files that you store on your computer (Garfinkel, 1995)[2]. It was firstly invented and published to the Internet by Philip Zimmermann in 1991. PGP itself is not an encryption algorithm, but a completed security program package integrating some of the encryption algorithm, for instance, RSA, IDEA, AES, etc (Harold F. Tipton, 2008). Philip Zimmermann has completed the following main jobs: 1. choosing some excellent algorithms as basic components for the encryption algorithm, then put them together into an application program; 2. making a program package including important files, later turning it to open-sources; 3. co-operating with enterprises in order to occupy the markets. (Mollin, 2006)

This paper is constructed as follows. In section two we indicate how dose PGP works. In section three we briefly survey the reliability from practical angle.

2. THE PRINCIPLE OF IMPLEMENTING PGP

As we have known, PGP is not only the name of program, but also the name of a network standard (RFC 2440: Open PGP Message Format (J. Callas, 1998)). In this section, we mainly discuss the PGP as RFC standard.

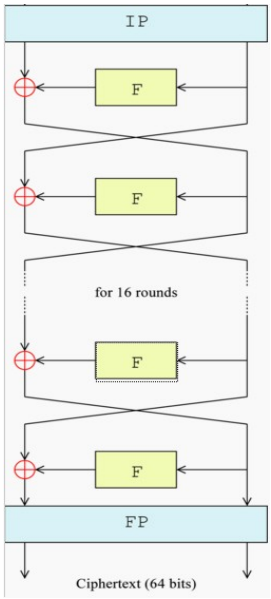
---

[1] PGP and Pretty Good Privacy are trademarks of Philip Zimmermann.
[2] Notice that this definition is not the only one, but a common idea among the academic realms.

PGP is a kind of system to protect the privacy of electronic documents based on 4 sections, namely, data compression, hashing, symmetric-cryptography, and public-key cryptography, each of which includes their own cryptographic algorithms, i.e. Hash function, DES (Data Encryption Standard), RSA, MD5. We will focus on how these algorithm works.

2.1 HASH FUNCTION

A hash function H is a computation which puts a variable-size input into the function and returns a fixed-size output, which is called the hash value (Thomas H. Cormen, 2001). The value is usually called message digest or digest (i.e. hashing a 3M file, outputting a 128 bit digest). Message digest represents the characteristics of the original data. When the original data changes, the re-generated message digest will also change, even if the change is only a very small part of the original data. Hence, hash function can be sensitive to detect whether the original data has been tampered. Combined with other algorithms, hash function can be used to protect the integrity of original data. Classic hash functions include: MD5, SHA-1, HMAC, GOST, etc.



What makes a good one-way[3] hash function?

1. The unfeasibility of reverse computing. Take arbitrary M and H,

[3] "One-way" in the name refers to the property of such functions: they are easy to compute, but their reverse functions are very difficult to compute.

h=H(m) can be computed easily, but it is not the case vice versa.

2. Weak collision property. Take arbitrary M, to find another M' satisfying H(M')=H(M) is unfeasible.

3. Strong collision property. To find a pair (M, M') satisfying H(M')=H(M) is unfeasible (Thomas H. Cormen, 2001).

2.2 DES (Data Encryption Standard)

DES is a kind of symmetric cryptography, which means the same key will be used in both encryption and decryption process. Its blocks and keys have a size 64-bits and 56-bits Figure 1 respectively. DES was developed by IBM, and adopted by National Bureau of Standard and American National Standard Institute in 1976. The overall Feistel structure of DES was displayed by Figure 1 (WIKIPEDIA, 2009).

The encryption steps are given as follows:

- Input is a plaintext block of the size 2w bits;
- The block is divided into two parts L and R;
- Two parts going through n rounds of processing;
- At every round, a function F (round function) is applied to the right half using a (sub)key, the result is XOR'cd with the left half of the data;
- At every round a new (sub) key may be used; all (sub) keys are generated from the same secret key (Lisitsa, 2008).

2.3 RSA

In 1978, Ron Rivest, Adi Shamir, and Leonard Adleman who were

at MIT issued the RSA algorithm to the public. The word “RSA” is the combination of their family name’ initials (Robinson, 2003). RSA is an asymmetric algorithm, which generally means two kinds of different keys are used in the processes of encryption and decryption, namely, public key and private key. Taking a metaphor, public keys, private keys and money are similar to a person’s bank account numbers, PINs, messages respectively. The account number can be known by everyone hence other people could be able to transmit the money to his/her account. However, no one consider it normal to tell the PIN to somebody else, or else his/her money in the bank will not be safe anymore. Furthermore, RSA algorithm is the one that can combine encryption and digital signature. Because of its durability, safety, simplicity, it has been accepted by the public as one of the most excellent public key encryption algorithms.

The generation of RSA keys

- The algorithm picks up two random and large primes p and q, then computes: $n = p * q$ and $\varphi(n) = (p-1) * (q-1)$.
- Uses the random encryption key e satisfying $1 < e < n$ and $\gcd(e, \varphi(n)) = 1$.
- Computes the unique $d \in N$, $ed \equiv 1 \pmod{\varphi(n)}$.
- Issues (n, e) as public keys, while keeps d, p, q, and$\varphi(n)$ as private keys (Mollin, RSA and public-key cryptography, 2003).

2.4 MD5

As an evolutionary version of MD4, MD5 (Message Digest algorithm 5) was published by Ron Rivest in early 90s. Actually, MD5 also

belongs to the hash algorithm, as what we have discussed above. Therefore, we will not bother to explain the principle of this algorithm. But one thing still need attention: although reports about decryption of MD5, HAVAL-18, MD4, and RIPEMD have been made by Xiaoyun Wang at Shandong University, in Crypto'2004, it is too haste to draw a conclusion that those algorithms all have been decrypted. In fact, Wang's decryption method just raised the decrypting efficiency to SHA-1 by 2000 times. It is not a big threaten to SHA-1, but till, it puts the SHA-1 to the surviving edge. No need to worry, we still have SHA-256 and SHA-512 to use. At the same time, experts are working hard on developing new hash functions.

3. THE RELIABILITY OF PGP

Using protection program does not guarantee a 100% security. Even if someone put the most secure lock on the door, the thief can still come into his/her house by the window. That is to say, a computer equipped with PGP can still be attack easily. There are a lot of attacks aiming to PGP, here we will only study some of these attacks.

Forms of attacks:

1. Attacking public and private Keys by Brute-Force is the most direct attack to PGP. In this case, the security of PGP depends on that of RSA and IDEA using by PGP. Hackers try to decrypt the key of PGP used for encryption.
2. The security of private keys of PGP is based on the following two points: accessing the private key and the degree of knowing about passwords of every private key. To use the private key, two points we've mentioned above is needed, accordingly, so dose attacks to private keys.

3. Since public keys depended by PGP play a crucial rule in the whole process, a lot of hackers will focus on attacking public keys.

When it comes to private keys, vulnerabilities of PGP's private keys can be investigated through the following steps. "In a chosen cipher-text attack, a hacker creates a message and then sends it to a targeted user with the expectation that this user will send the message to other users. When the targeted user distributes the message in an encrypted form, the hacker listens to the messages and computes the key from the newly created cipher-text[4]" (Michael Cross, 2002).

Then, our discussion will focus on the attacks of the public key.
According to the procedure of implementing PGP, let us consider a simple case first. Hacker C intercepts public keys of A and B when they are exchanged by their owner. Next, C uses his/her own public key to replace that of both A and B. In other words, both A and B will consider C's public key as each other's public key. Finally, C can listen the messages between A and by using his/her private key.

The above procedure is based on A and B passing public keys to each other and C intercepting the public keys. However, it is clear that if A and B use the certificated signature when they exchange their public keys, they can avoid the above-mentioned attacks effectively, for the reason that C can not pose his/her public key to A or B since they will ask their common authorized user D for each other's public key, thence increasing the difficulty of attacking. Unfortunately, Hackers still have some ways to attack.

[4] This is cited from page 131 [Michael Cross, 2002].

1. The public key ring will be checked only after it has been changed. That is to say, if new keys or signatures come, PGP will check them and put a tag on them indicating they have been checked, then PGP will never check them again. Therefore, hackers can modify the signatures in a public key ring to an “already-checked” one. By this procedure, the system will not check these changed signatures and hackers can attack again.
2. Another attack to a PGP public ring is possible though the use of PGP. Every public key have a “valid bit” included in the prefix form. PGP will compute the length of new arrived signature’s valid bit, and then put it into the buffer of the public key ring. Hackers could modify this valid bit from public key ring, thence making users trust this invalid key. For example, user A will check whether the public key received is valid in order to set the value to it. Meanwhile, the hacker C could make this key valid by changing its valid bit, thereby letting A believe that what A has received is really belonged to B, although there is no signature to identify the validity of this public key, regardless of the fact that, in fact, that is C’s public key.
3. The inducer’s trust key exists in the buffer of public key ring. The trust key can define the amount/size of trust to it. Therefore, PGP will accept an invalid key as a valid key by using a key with certain trust value to give a signature to that invalid key. For instance, if A/B totally trust certificate PK_D coming from D, then both A and B have the public key of D. But if a hacker C changed PK_D to PK_c, then C could use PK_c to signage for other public keys, hence making A/B trust those changed public keys (Vlastimil Klíma, 2001).

4. CONCLUSION

This paper represents the achievable theory of PGP from the algorithms of hash, DES, RSA and MD5. Some secure issues, including attack forms, are also described in the essay.
From what we have discussed above, we can safely conclude that one of the most serious problems within the PGP procedure should be that public key ring is no protection procedure providing safety for buffers. Everyone could be able to modify any bit of the public key ring without being found though any binary-file tools who can understand PGP codes and who access public key ring. According to this, PGP program should protect public key ring carefully, and detect any malice tamper immediately.

The PGP algorithm has never been proofed as having 100% security. Even though the mathematics on which PGP are based are considered very safe, but it is feasible to attack PGP as long as hackers have found some bugs of it. In other words, nothing has exact safety. If there is enough time and sources, every encryption algorithm will be conquered. However, the question is whether it is worthwhile to decode the data protected by encryption algorithms by spending certain amount of time and sources. We should be aware the fact that, the cost of decryption has been cutting as the time passes, because computers are becoming more and more efficient, and the hardware getting more and more chip. Anyway, hackers still have a long way to go in order to exceed the cryptologists.

REFERENCES

1. Garfinkel, S. (1995). *PGP: Pretty Good Privacy.* O'Reilly Media, Inc.
2. Harold F. Tipton, M. K. (2008). *Information Security Management Handbook.* AUERBACH.
3. J. Callas, L. D. (1998, 11). *OpenPGP Message Format.* Retrieved 10 25, 2009, from The Internet Engineering Task Force : http://www.ietf.org/rfc/rfc2440.txt
4. LisitsaAlexei. (2008, 12, 12). accessing date: 10, 21, 2009 source: University of Liverpool: http://www.csc.liv.ac.uk/~alexei/COMP522/COMP522-SymmetricEnc-07.pdf
5. Michael Cross, N. L. (2002). *Security plus study guide and DVD training system* (1th edition ed.). Syngress.
6. Mollin, R. A. (2006). *An introduction to cryptography* (2th edition ed.). Chapman & Hall/CRC.
7. MollinA.Richard. (2003). RSA and public-key cryptography. CHAPMAN & HALL/CRC.
8. RobinsonSara. (2003). Still Guarding Secrets after Years of Attacks, RSA Earns Accolades for its Founders. SIAM News, 36.
9. Thomas H. Cormen, C. E. (2001). *Introduction to Algorithms* (2nd edition ed.). The MIT Press.
10. Vlastimil Klíma, a. T. (2001, Mar). Attack on Private Signature Keys of the OpenPGP format, PGPTM programs and other applications compatible with OpenPGP. *Citeseer* .
11. *WIKIPEDIA.* (2009, 10, 21). Retrieved 10 25, 2009, from http://en.wikipedia.org/wiki/Symmetric-key_cryptography
12. WIKIPEDIA. (2009, 10, 23). Retrieved 10, 25, 2009, from http://en.wikipedia.org/wiki/Data_Encryption_Standard#The_Feistal_.28F.29_function