International Journal of Enhanced Research in Management & Computer Applications
ISSN: 2319-7471, Vol. 5 Issue 6, June-2016, Impact Factor: 1.544

Mail-Server Unleashed

Bharat Bhushan[1], Dr. Yashpal Singh[2]

[1]Research Scholar, Department of Computer Science & Engineering, Ganga Institute of Technology and Management, Kabalana, Jhajjar, Haryana, INDIA
[2]Associate Professor, Department of Computer Science & Engineering, Ganga Institute of Technology and Management, Kabalana, Jhajjar, Haryana, INDIA

ABSTRACT

Hodiernal Cryptovirology is rooted on cryptographic based algorithms and mathematical functions. Ccryptographic algorithms are contemplated around computational hardness assumptions, making such algorithms hard to break in practice by any adversary. It is theoretically possible to break such a system, but it is not feasible to do so by any known practical means. These schemes are, therefore, termed computationally secure, theoretical advances, e.g., improvements in integer factorization algorithms, and faster computing technology require these solutions to be continuously adopted. There exist information-theoretically secure schemes that probably cannot be broken with the computing power available as on date, an example is the one-time pad but these schemes are more difficult to implement than the best theoretically breakable but computationally secure mechanisms.

Keywords: Cryptography, Crypto virus, Algorithms, Jacobi, Synchronous, Asynchronous.

I. INTRODUCTION

Benefits of Crypto virology from General Advances

The original Macintosh cryptovirus was a proof-of-concept that had substantial room for improvement. e.g. instead of using the outputs of truerand directly, it could have applied Neumann's algorithm or a more advanced entropy extractor to the output of truerand. Ideally, a cryptovirus would also utilize such sources as hard disk turbulence and the RNG that is provided by the underlying operating system. Although the simple extortion attack did not require a large number of random bits, it should have nonetheless applied an entropy extractor to several different sources of entropy.

II. STRONG CRYPTO YIELDS STRONG CRYPTOVIRUSES

The RSA cipher is secure enough to mount the cryptovirus extortion attack. However, improved public key cryptosystems exist. A known drawback to RSA is that each RSA ciphertext c = me mod n leaks the Jacobi symbol of m with respect to n. The Jacobi symbol of m with respect to n is either 1 or −1. Since e is odd, it is not hard to see that, J(m/n) = J(m/n)e = J(me/n) = J(c/n)

n \ m	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	1																
3	0	1	-1														
5	0	1	-1	-1	1												
7	0	1	1	-1	1	-1	-1										
9	0	1	1	0	1	1	0	1	1								
11	0	1	-1	1	1	1	-1	-1	-1	1	-1						
13	0	1	-1	1	1	-1	-1	-1	-1	1	1	-1	1				
15	0	1	1	0	1	0	0	-1	1	0	0	-1	0	-1	-1		
17	0	1	1	-1	1	-1	-1	-1	1	1	-1	-1	-1	1	-1	1	1

Jacobi symbol (m/n) for various *m* (along top) and *n* (along left side). Only $0 \le m < n$ are shown, since due to rule (2) below any other *m* can be reduced modulo *n*. Quadratic residues are highlighted in yellow — note that no entry with a Jacobi symbol of -1 is a quadratic residue, and if *m* is a quadratic residue modulo a coprime *n*, then $(m/n) = 1$, but not all entries with a Jacobi symbol of 1 (see the $n = 9$ row) are quadratic residues. Notice also that when either *n* or *m* is a square, all values are nonnegative.

Figure 1 Jacobi Symbol

Since the Jacobi symbol must be one of two different values, a single bit of information about m is leaked in every ciphertext. Ideally, no poly-time computable function of the plaintext should be efficiently computable given the ciphertext. In this example the computable function has a certain semantic meaning, namely the Jacobi function. But other efficiently computable functions with semantic meaning may exist. Another drawback to RSA is that it is deterministic. Every time that a particular message m1 is encrypted in RSA the exact same ciphertext c1 will result. This means that it is possible to guess the plaintext in a given ciphertext and then verify the guess. For example, suppose c1 = me 1 mod n is known. If m2 is suspected as being the plaintext it can be encrypted to obtain c2 = me 2 mod n. If c2 = c1 then m2 = m1. When a user encrypts short messages there is the risk that someone else can guess the plaintext and verify the guess. A heuristic way to prevent adversaries from guessing the plaintext and then verifying is to shrink the message space and include a random bit string in each encryption. This makes each plaintext message map to more than one ciphertext message. However, a probably secure approach is more desirable. Generally, it should not be possible to select two messages and distinguish between their encryptions.

These observations lead to two notions of security: semantic security and message in distinguish ability. As it turns out, when the adversary is allowed to be any probabilistic poly-time algorithm, these two definitions of security are equivalent. To avoid such vulnerabilities a semantically secure cryptosystem can be used. An encryption algorithm is semantically secure against plaintext attacks if for all probability distributions over the message space, anything that a passive adversary can compute efficiently about the plaintext given the cipher text can also be efficiently computed without the cipher text given any historical information about the plaintext. An efficient public key cryptosystem called Optimal Asymmetric Encryption Padding (OAEP) has been proposed that can be implemented given any trapdoor one-way function such as RSA. Its security was proven within the random oracle model.

When a cryptosystem is used that is semantically secure against plaintext attacks to encrypt a symmetric key, no partial information about the symmetric key is revealed in the resulting cipher text. Such a cryptosystem therefore provides more security than using RSA as originally defined. In computer security, the standard approach is to formalize the capabilities of the adversary in a threat model, develop an algorithm to deal with the possible presence of the adversary, and then prove that the algorithm is secure within that threat model. The bottom line is that a cryptosystem is only guaranteed to be secure within the threat model that it is designed to handle and may well succumb to more powerful adversaries should they exist. Semantic security against plaintext attacks guarantees that an adversary that tries to determine some additional partial information concerning a plaintext given the corresponding cipher text along with existing partial information cannot do so.

However, there exists an even more powerful type of adversary. For example, in the case of the crypto virus extortion attack what has not been considered is a group of victims that is prepared to pay multiple ransoms, that use the virus writer as a decryption oracle, and that choose the public key cipher texts based on previously decrypted cipher texts. This amounts to what is called an adaptive chosen-cipher text attack. This illustrates some of the subtleties in designing secure systems. A cryptosystem that is semantically secure against plaintext attacks is not necessarily semantically secure against adaptive chosen-cipher text attacks. However, it turns out that OAEP is secure against adaptive chosen-cipher text attacks. This reasoning implies that crypto viruses benefit directly from general advances in cryptology. Even though the original crypto virus did not use multiple entropy sources and an entropy extractor, and even though it did not use OAEP, it is straight forward to utilize these more advanced techniques to design secure crypto viruses.

III. MIX NETWORKS AND CRYPTO VIRUS EXTORTION

Arguably, the weakest aspect of the extortion attack from the perspective of the virus writer is obtaining the ransom without getting caught. To this end, it may be best to avoid demanding cash entirely. If the virus writer seeks information alone, then a more attractive alternative is possible. Methods exist that enable two mutually distrusting parties to communicate securely over a network in an anonymous fashion. The basic vehicle for doing so is called a mix network. A mix network forms the basis for anonymous re-mailing systems and is a fundamental building block for many cryptographic protocols. In a nutshell a mix network is a service that lets users send messages anonymously to other users, and that makes the correlation of output messages with input messages nearly impossible.

Catagories of Mix networks:
There are two major categories:

- ✓ Synchronous mixes
- ✓ Asynchronous mixes.

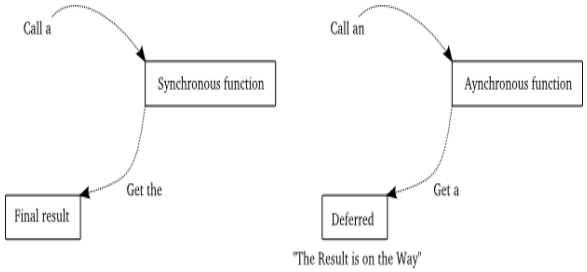

Figure 2 Synchronous and Asynchronous Mixes

In practice asynchronous mixes are ideal for anonymizing e-mail traffic, whereas synchronous mixes are best for randomizing message traffic in batches. It is typically easier to produce formal proofs of security for synchronous mixes than for asynchronous mixes, and as a result synchronous mixes tend to be employed in protocols such as electronic voting. This allows ballots to be cast in a probably anonymous fashion. Asynchronous mixes are arguably easier to deploy on a large scale and rely on a number of heuristic defenses against particular attacks. A mix network consists of a collection of N mix net nodes. The basic idea behind an asynchronous mix is to take an incoming message, send it from node to node along a randomly chosen path, and then send it to its final destination.

It is necessary to use encryption to prevent correlations based on content as well as to fix the length of each message so that correlations based on size are not possible. This implies that large messages need to be broken down into smaller pieces, and short messages need to be padded out to the requisite length. The fixed sized messages are encrypted using a probabilistic public key cryptosystem. Therefore, even if the same message is sent through the network on more than one occasion it will look different each time with overwhelming probability. Not only outsiders are a threat to mix networks, but insiders are a threat as well. A properly designed mix net is still secure even if a fixed fraction of the nodes are operated by malicious persons that collude in order to track messages. Assuming that a sufficient number of messages go in and out of a given node at any given time it is important that the message go through multiple honest nodes to make the probability of tracing it Negligible. Onion routing is a common method for implementing asynchronous mix networks.

In an onion routing system, each of the N nodes has a key pair. A user selects a random traversal among the N nodes and successively encrypts in reverse-order the message using the public keys corresponding to the nodes that the message will traverse. In choosing the traversal, the same node can be chosen multiple times and therefore loops are possible in the path that the message takes. The layers of the resulting ciphertext are peeled away by performing decryption as the message travels through the network. By padding with random bytes, it is possible to accomplish this in such a way that the length of the transmitted message is always the same. Over time a given mix net node may receive 1,000 incoming messages. These are decrypted, the order of them is mixed, and the resulting messages are sent out. The original packet headers are discarded and the messages are sent in newly constructed packets.

This prevents trivial matching based on packet headers. If a message is decrypted entirely then it is sent to the final recipient. This way, a given input message to the node may end up being any of 1,000 different outgoing messages. It is also possible for a mix node to decrypt a message, determine the next intended recipient, and then re-encrypt the message using the public key of the recipient. When this re-encryption is probabilistic, it makes it more difficult for the original sender to identify his or her own message as it moves through the mix network. An asynchronous mix network must have a sufficiently large message volume at all times. If only a handful of messages are traveling through the network then correlation is trivial. There are numerous attacks on mix networks that have been described in the literature. For example, an active adversary may try to determine the partial path of an unknown message by sandwiching it between messages of the adversaries' own choosing.

The object is to fill the queue of messages in the mix with custom-made messages, all except for the one that needs to be traced. The adversary then watches the messages as they leave the node to determine where the sandwiched message goes. The adversary takes note of how many messages each intended recipient receives. This is enough to determine where the sandwiched message went. A novel solution to this problem has been proposed. The idea is to regard the mix net nodes as probabilistic algorithms and let them affect the paths that messages take. The way this is accomplished is by flipping coins, and with a certain probability sending a given message on a short, randomly chosen inter-mix detour. When a detour occurs, it has the effect of adding a few new layers back onto the message in question. This mechanism has the novel property that even the sender does not know for sure what path his or her message will take within the mix network. The probability that a given message is sent on a detour must be low enough to keep the message volume from growing out of control. Methods have been devised to not only allow anonymous messages to be sent, but also to allow anonymous replies. In the

cryptovirus attack, the virus can instruct the victim to place the victim's email address on a public bulletin board. To avoid embarrassment, the virus can tell the victim to first encrypt the e-mail address using the public key in the virus.

The e-mail address can be chosen specifically for dealing with the virus writer and therefore not reveal the identity of the victim. The virus writer periodically scans the board for such ciphertexts, and decrypts.them when found. The virus writer then sends the demand anonymously to the victim. Provided that the ransom is information, the victim can include the ransom within the anonymous reply. As long as the ransom itself does not reveal the identity of the victim, the attack preserves the victim's anonymity. A mix network is therefore a powerful building block for carrying out cryptovirus attacks.

To apprehend the virus writer, law enforcement bodies may seek to subpoena, the administrators of each node in the mix network. Such a subpoena might call for the current private key and all previous private keys of each of the administrators. If all of the needed private keys and message traffic were obtained, this would allow law enforcement to trace any given message. However, if each mix net node adhered to a de facto mix net protocol standard, generated new key pairs every so often, and deleted all previous private keys and coin tosses, then the subpoena would likely not help law enforcement. Also, if the nodes spanned multiple countries then the tracing effort would be hampered even more due to legal complications. The fact that mix net nodes traditionally decrypt incoming messages and then re-encrypt them when they are sent out, implies that the individual private keys of each mix net administrator can be used to trace message traffic. A recent method known as universal re-encryption has been proposed as a basis for a provably secure mix network.

By using a cryptosystem such as ElGamal that allows re-encryption without first decrypting, it is possible to have the mix net nodes randomize the incoming messages in an oblivious fashion. With respect to the virus attack, this implies that there are no administrator private keys for law enforcement to subpoena. If a re-encryption mix net node does not store the random permutation that it used in a mix operation, then the permutation is effectively lost forever. This property makes re-encryption mixes very attractive to criminals that need to communicate anonymously. Various indirect methods exist to achieve financial gain through extortion. For example, a determined attacker may premeditate an extortion attempt by purchasing several shares of a small public company, provided that a substantial number of shares are up for sale. Once the attack is carried out the victim can be forced to purchase a high volume of shares from the small company. This has a tendency to drive the share price up, at which point the attacker can cash out. The obvious drawback to this method is that all of the outstanding shareholders may be regarded as suspects.

IV. RELATED WORK

4.1 MAIL SERVER UNLEASHED

We have tried several ways to secure a server or check in loop holes into the mail servers by applying the crypto virology. Below we tried less harmless crypto virus onto the different mail server after completion of specific research work we remove them from the server though server can't get affected by them. In following process we will found that some mail servers are stronger enough to face such type of virus on other side some are not. Below are the experiments done during the specific period of time:

(a) In Figure 3 we upload the specific file containing the malicious crypto code onto the rediffmail, and here we realize that no protection is available out there on to those servers to help users to save themselves from the crypto viruses.

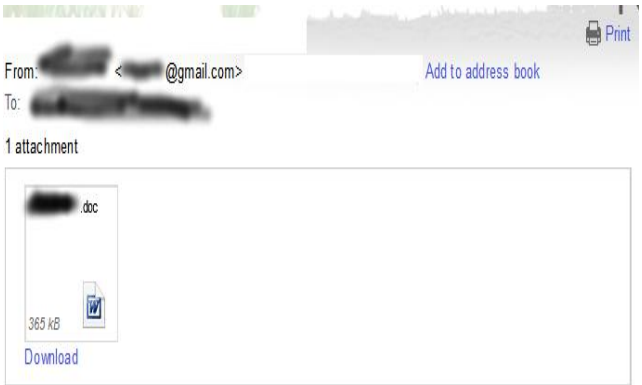

Figure 3 In-secured Rediffmail server

(b) In Figure 4. The encrypted file with the extension successfully uploaded to rediffmail but gmail is much secured and it detects the malicious code into the file.

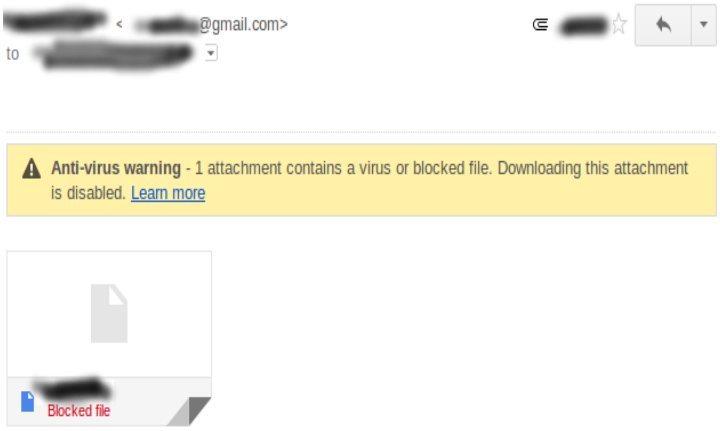

Figure 4. Gmail server

Now the question arises that whether there is some other way available where we can hide the specific code even from the gmail antivirus scanners. So we encrypt the file and divide the code of the file into the small pieces by using compression software, the division of code bring the specific code untraceable and our file is available for download as shown in Figure 5.

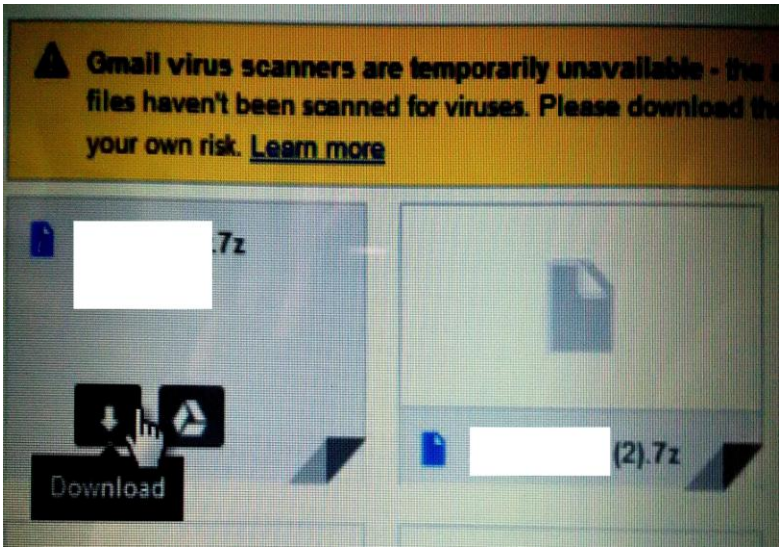

Figure 5. Gmail Scanners

The cryptoviruses encrypts files with the following extensions:

- *.odt, *.ods, *.odp, *.odm, *.odc, *.odb *.doc, *.docx, *.docm, *.wps, *.xls, *.xlsx, *.xlsm, *.xlsb, *.xlk, *.ppt, *.pptx, *.pptm, *.mdb, *.accdb, *.pst, *.dwg, *.dxf, *.dxg, *.wpd, *.rtf, *.wb2, *.mdf, *.dbf, *.psd, *.pdd, *.eps, *.indd, *.cdr, *.dng, *.3fr, *.arw, *.srf, *.sr2, *.bay, *.crw, *.cr2, *.dcr, *.kdc, *.erf, *.mef, *.mrw *.nef, *.nrw, *.orf, *.raf, *.raw, *.rwl, *.rw2, *.r3d, *.ptx, *.pef, *.srw, *.x3f, *.der, *.cer, *.crt, *.pem, *.pfx, *.p12, *.p7b, *.p7c

The following Figure 4.4 shows the files affected by the Teslacrypt virus.

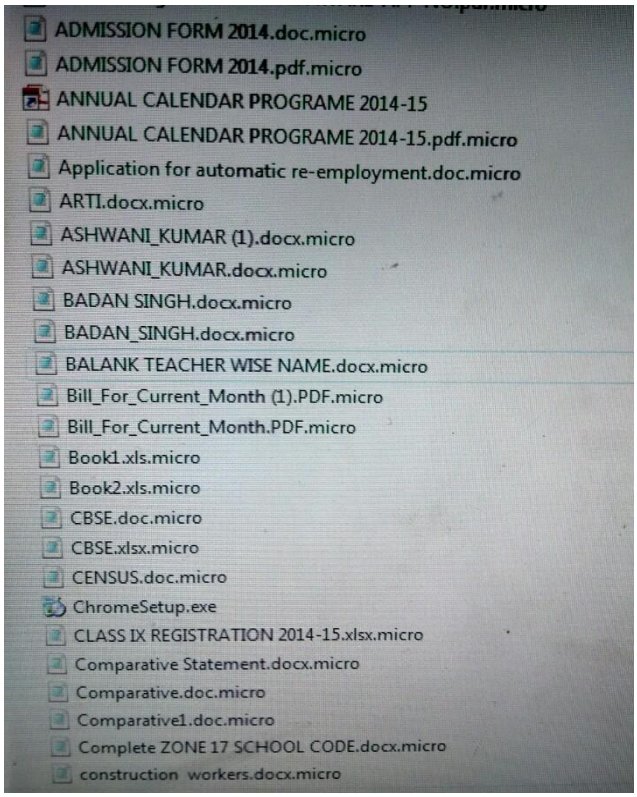

Figure 4.4 Files affected by the Crypto viruses

VI. FUTURE SCOPE

It is the need of hour to realize that if we build more tools regarding the toughness of cryptography, it will help the crypto viruses also and we are likely to face more fast viruses in near the future. Therefore, the only way to protect our data/information from the crypto virus, is to develop habit to have scheduled backup on daily basis.

REFERENCES

[1]. http://www.welivesecurity.com/2013/12/12/11-things-you-can-do-to-protect-against-ransomware-including-cryptolocker/
[2]. Adam Young and Moti Yung, Cryptovirology: Extortion-Based Security Threat and Countermeasures, Proceedings of the 1996 IEEE Symposium on Security and Privacy.
[3]. https://heimdalsecurity.com/blog/ctb-locker-ransomware/
[4]. Shivale Saurabh Anandrao, "Cryptovirology: Virus Approach" International Journal of Network Security & Its Applications (IJNSA), Vol.3, No.4, July 2011
[5]. https://www.opendns.com/enterprise-security/threat-enforcement/features/cryptolocker-containment-is-the-new-prevention/
[6]. Adam Young and Moti Yung, Deniable password snatching: On the possibility of Evasive Electronic Espionage, the 1997 IEEE Symposium on Security and Privacy.
[7]. "Malicious Cryptography Exposing Cryptovirology" by Adam Young Moti Yung , Wiley Publishing, Inc.
[8]. http://www.pandasecurity.com/mediacenter/malware/cryptolocker/
[9]. Weaver, N., Potential Strategies for High Speed Active Worms: A Worst Case Analysis, http://www.cs.berkeley.edu/~nweaver/worms.pdf, last accessed on December 4, 2002.
[10]. https://www.symantec.com/security_response/writeup.jsp?docid=2013-091122-3112-99&tabid=2
[11]. CERT Coordination Center, CERT® Advisory CA-2002-27 Apache/mod_ssl Worm, http://www.cert.org/advisories/CA-2002-27.html, last accessed on December 4, 2002
[12]. https://www.linkedin.com/pulse/20141028135604-2269330-computer-crypto-virus-on-the-rise
[13]. Netcraft, Netcraft Web Server Survey, http://www.netcraft.com/survey, last accessed on December 4, 2002
[14]. https://nordic-backup.com/guides/how-to-protect-yourself-from-cryptovirus-attacks/
[15]. http://esupport.trendmicro.com/solution/en-US/1099423.aspx
[16]. https://coolchilli.com/crypto-virus-information-advice
[17]. http://www.abc.net.au/news/2014-10-07/fake-auspost-emails-used-in-crypto-ransomware-attack/5795734
[18]. https://www.hilltopconsultants.com/blog/new-cryptolocker-virus-february-20