




Research Article

On Man-in-the-Middle Attack Risks of the VPN Gate Relay System

Yunxiao Sun ^{1,2}, Bailing Wang ^{1,2}, Chao Wang ^{1,2} and Yuliang Wei^{1,2}

¹School of Computer Science and Technology, Harbin Institute of Technology, Weihai 264209, China

²Research Institute of Cyberspace Security, Harbin Institute of Technology, Harbin 150001, China

Correspondence should be addressed to Yunxiao Sun; syx@hitwh.edu.cn

Received 28 June 2021; Accepted 8 October 2021; Published 21 October 2021

Academic Editor: Zhe-Li Liu

Copyright © 2021 Yunxiao Sun et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

With the development of the Internet, more and more people use virtual private network (VPN) to circumvent censorship or hide themselves for privacy purposes. However, VPN itself faces some security and privacy risks. Widely used all over the world, the VPN Gate is a volunteer-organized public VPN relay system launched in 2013. By analyzing the security of the system, we have found that there is a man-in-the-middle attack risk because an attacker may hijack a VPN session and decrypt the traffic. According to our study, the reason of the security issues is the misuse of the SSL certificate. To mitigate the security risks, we offered a series of recommendations.

1. Introduction

A virtual private network (VPN) establishes a secure and encrypted tunnel to transfer data. Users use this to access services or resources in the remote networks. Either to circumvent censorship [1] or for privacy purpose [2], many users use VPN to obtain network services. However, VPN still faces some privacy and security risks [3]. The VPN Gate [4] has been an academic experimental platform started by the University of Tsukuba since 2013. Used as a platform for running the VPN Gate plugins, the SoftEther VPN is open-source software to support multiprotocols, including L2TP/IPSec, OpenVPN, MS-SSTP, and “SSL-VPN” [5]. According to the website of the VPN Gate project, the motivations of the project are (1) to bypass censorship firewalls, (2) to hide IP address, and (3) to antipacket sniffing [6]. The VPN Gate has grown into a large distributed system (up to 14,000 volunteer nodes during peak hours). With numerous system nodes and a wide range of users, the security of the system is worthy of further study.

The analysis of security protocols can be divided into two technical routes, as we call design-oriented approach and implementation-oriented approach. Design-oriented security analysis generally establishes a formal model of the protocol and verifies whether the protocol has specific

security properties through logical reasoning. Tamarin [7] is a commonly used tool for formal analysis of security protocols. Implementation-oriented security analysis focuses on the discovery of the errors in implementing an encryption algorithm or misuse of a security protocol in the software or systems. Literatures [8, 9] analyzed the security vulnerabilities in the implementation of a security protocol. Literature [10] showed the security and privacy risks of commercial VPN services.

This paper explores security risks when a user connects to a public VPN server offered by the VPN Gate relay system using “SSL-VPN.” We have found two security issues of the VPN Gate system: (1) SSL certificate is shared with all the volunteers, even an untrusted volunteer. An attacker can pretend to be a normal volunteer node, redirect user traffic to the controlled malicious node via IP hijacking, and then sniffing the user’s traffic; (2) Because the VPN Gate lacks SSL certificate verification at client side, the attacker can launch a classical SSL man-in-the-middle attack with any SSL certificate (even a self-signed certificate) to obtain the VPN encryption key and then decrypt the VPN traffic. The above issues break the authentication mechanism of the SSL communication so that the confidentiality and integrity of information are hard to be guaranteed.

Our main contributions are discovering the security risks of the certificate sharing mechanism currently adopted by the VPN Gate system and evaluating the feasibility and the complexity of the exploitation. To illustrate the impact of the security issues, we constructed two attack scenarios. The attacks we constructed do not need the exploitation of software vulnerabilities, nor a forged certificate issued by Certificate Authority, and may be easily launched by government censorship firewalls, ISPs, or a skilled person. This means that when using the VPN Gate in an environment with strict Internet censorship mechanisms, the user's privacy may be leaked.

We will present some background knowledge on the VPN Gate relay system and the SSL-VPN protocol in Section 2, discuss the threats of the security issues and the mitigate method in Section 3, and in Section 4, we introduce the attack scenarios and do the experiments. To facilitate verifying or reproducing the experiments in this paper, we have published the Proof-of-Concept on GitHub at [11]. We hope that this paper could be a starting point and attract more researchers to analyze the system in-depth and mitigate the security risks.

2. Background Knowledge

In this section, we introduce the background knowledge of the VPN Gate relay system and the SSL-VPN protocol used by the system. The SoftEther VPN includes two software programs named SoftEther-client and SoftEther-server. If a user chooses to enable the VPN Gate function on the SoftEther VPN server [5], his or her computer will become a volunteer node in the VPN Gate system, providing VPN services for other people.

To create an encrypted VPN tunnel, the VPN Gate uses a particular SSL-VPN, a protocol with two modes, TCP mode and UDP mode. However, the VPN Gate recommends users to choose TCP mode, and thus the research work in this paper only involves TCP mode. As it is noticed that when working in TCP mode, the client will first establish a TLS session, and then a UDP session. By analyzing the source code of the SoftEther, we have learned that the UDP session is used for data transmission acceleration and the VPN Gate enables UDP acceleration by default. Since the SoftEther does not have an official protocol specification to SSL-VPN and lacks unified terms to describe the protocol interaction process, for convenience of presentation, we call the TLS session control-channel and the UDP session UDPAcc-channel, respectively, based on the understanding of the protocol. The knowledge of the SoftEther protocol mainly comes from the source code of the project at [12].

The establishment of a VPN tunnel can be divided into two stages. In the first stage, the client first initiates the control-channel, and then the client and server will exchange information on software version, operating system, IP address, port, and encryption keys through the encrypted TLS channel. In the second stage, the encrypted UDPAcc-channel will be established based on the ports and encryption keys from the message exchanged in the first stage.

2.1. The Control-Channel Protocol. The control-channel in the SoftEther protocol is an encrypted channel over TLS. The initiator sends a random 128 byte `udp_acceleration_client_key_v2` to the responder via the HTTP request "POST/vpnsvc/vpn.cgi," then the responder sends `udp_acceleration_server_key_v2` back to initiator via the HTTP response with the same length as the initiator does. The notation of keys, taken from the source code of SoftEther software, means that the keys are used by UDP acceleration, and we call them `UDPACC_KEY_Ver2` for short. Both the initiator and the responder will take the first 32 bytes of the `UDPACC_KEY_Ver2` as encryption keys used by the UDPAcc-channel. The message sequence chart of the control-channel protocol is shown in Figure 1. The description of the protocol here mainly focuses on the cryptography-related parts, omitting some protocol implementation details.

2.2. The UDPAcc-Channel Protocol. The UDPAcc-channel is an encrypted UDP tunnel used for data transmission acceleration. The encryption algorithm of the UDPAcc-channel is Chacha20-Poly1305, and the encryption key is generated from `UDPACC_KEY_Ver2`, which has been exchanged via the control-channel in the process of Figure 1 as described in Section 2.1. The packet structure of the UDPAcc-channel is shown as Figure 2, where the notation IV is the initialization vector of the encryption algorithm, and the TAG entry is the MAC of the encrypted message. The data entry is an Ethernet frame received by VPN virtual network adapter, and the data length entry is the length of the Ethernet frame. The Flag entry is a compressed symbol, which is disabled by default, meaning that the data have not been compressed.

3. Discussion of the Security Issues

According to the description of the SoftEther protocol in Section 2, we can learn that the security of encrypted traffic depends on the confidentiality of the symmetric encryption key used in UDPAcc-channel, which is exchanged by the client and the server through the TLS-based control-channel. Therefore, the security of the TLS session is the key factor in ensuring the confidentiality of transmitted information. Through a detailed study on the VPN Gate software, combined with analyzing the captured packets, we found the following two security issues.

3.1. Issue 1: Sharing SSL Certificate with Untrusted Volunteers. The VPN Gate system relies on volunteers to provide VPN services, but it is hard to distinguish whether a volunteer has malicious intentions or not. The attacker can also set up a honeypot and pretend to be a normal node in the system. All the volunteer nodes in the VPN Gate system use the same server certificate, which has been issued to the domain "`*.opengw.net`" by Sectigo RSA Domain Validation Secure CA. This means that the malicious node and the normal node have the same identity authentication materials, and

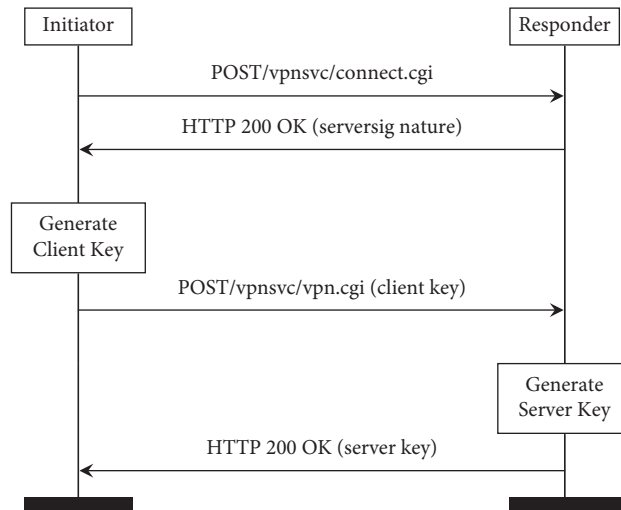


FIGURE 1: The message sequence of the control-channel.

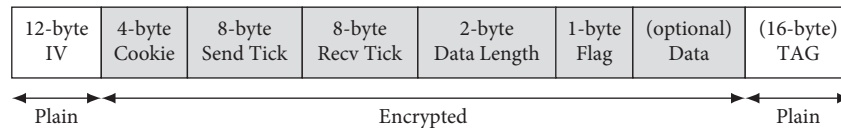


FIGURE 2: The packet structure of the UDPACC-channel.

the client cannot tell if a node is malicious node or not by checking the server certificate.

TLS runs on top of the TCP/IP protocol, and the TCP/IP protocol is considered as an unreliable channel because the data transmitted through the IP protocol lack integrity checks. When an attacker deploys a malicious middle-box in the wire, he or she has the ability to modify any IP packet. The attacker can modify the destination address of an IP packet, recalculate the IP header checksum, and send the modified IP packet to the wire; thus, the VPN session has been hijacked and redirected to a malicious node. The attack is of low cost and easy to implement because the attacker does not need to implement a complete TLS protocol stack or VPN protocol stack and only need to modify the IP header. The malicious node has the same certificate as a normal volunteer node does, even a skilled user cannot figure out the attack only by certificate checking.

The root cause of this security issue is how VPN Gate organizes the distributed volunteers. It is dangerous to share server certificate with all the volunteers because we cannot identify a malicious volunteer node. We think this issue is a problem in protocol design, which is worthy of further study. The way to mitigate the threat is to verify Server Name Identification (SNI) in the TLS handshake protocol at the server side. When the server finds that the SNI field in the ClientHello message is inconsistent with its own IP, it should immediately end the TLS handshake process. To demonstrate the impacts posed by this security issue and prove the feasibility of the attack, we constructed an attack scenario in Section 4.1.

3.2. Issue 2: Lacking SSL Certificate Verification at Client Side.

It is well known that the SSL certificate is used for identity authentication in TLS protocol. The client should verify the server certificate to protect against man-in-the-middle attack. However, in the VPN Gate system, we find that the client side does not verify the server certificate because the option is disabled by default. We tried to enable the certificate verification option on the VPN Gate client, but the software prompts that the user has no permission to change the configuration. This can be attributed to the fact that the current configuration of the VPN Gate cannot resist man-in-the-middle attacks. When there is a malicious intermediary node in the communication link, security cannot be guaranteed.

A man-in-the-middle attacker can implement IP hijacking by modifying the IP destination address and then implement TLS man-in-the-middle attack. Due to the lack of certificate verification, an attacker even does not need a forged server certificate issued by a trusted CA and only needs to issue a self-signed certificate to implement a TLS man-in-the-middle attack. Once the attack is successfully implemented, the attacker can parse out the symmetric encryption key of the UDPAcc-channel and then decrypt the packets transmitted in the UDPAcc-channel. The encryption key in UDPAcc-channel protocol is exchanged over the control-channel, which does not achieve perfect forward secrecy, and this means that the attacker can save the UDPAcc-channel traffic as pcap file and then decrypt it offline.

The basic reason of this security issue is that certificate verification has been disabled by default by the VPNGate, which is a misuse of the TLS protocol. The way to solve this

problem is to modify the configuration of the client to enable SSL certificate verification. To demonstrate the impacts posed by this security issue and prove the feasibility of the attack, we constructed an attack scenario in Section 4.2.

4. Attack Scenarios

In order to make the VPN Gate users have an intuitive understanding of the security issues, we adopted a simple and direct attack method in the attack scenario. Our intention is not to invent new attack methods but prefer to use a simple and old trick to break the encrypted VPN tunnel because a simple trick means low cost and low complexity. The attack is dependent on the redirection by a malicious middle-box, which has been a well-known security threats. In the experiment, we used a malicious router to act as a government censorship firewall. The malicious router is just used to explain the mechanism of man-in-the-middle attacks. It does not mean that government censorship firewalls or ISPs will actually use a malicious router to decrypt the traffic. They may use high performance middle-boxes to carry out a large-scale attack.

4.1. Scenario 1: VPN Session Hijacking. This scenario is used to illustrate how an attacker hijacks the VPN session with security issue mentioned in Section 3.1. The network topology of the experimental environment is shown in Figure 3. The attack can be launched by the following steps: (1) installing SoftEther server software on the malicious server, (2) on the user interface of the SoftEther server software, enabling the VPN Gate option, so that the malicious server now becomes a volunteer node of the VPN Gate relay system, and (3) adding DNAT rules in the malicious OpenWrt router to redirect all VPN traffics to the malicious server. To add the DNAT rules, we use the following Linux Shell commands: `iptables -t nat -A PREROUTING -d 219.100.37.X --dport 443 -p tcp -m tcp -j DNAT --to-destination 47.242.230.X:443`

In this experimental environment, the client intends to establish a VPN connection with 219.100.37.X, but the traffic has been purposely redirected to 47.242.230.X by the attacker. The malicious node runs the same software as the normal volunteer node does and can establish an encrypted VPN tunnel with the client. The VPN server acts as a proxy server, and so that when querying the IP address via some website such as “whatismyip.com,” the VPN Gate user should get the IP address of the VPN server to which the computer has connected. However, if this attack occurs, when the client sees on the user interface of SoftEther-client, the IP address of the VPN server connected is 219.100.37.X, but the IP address queried through `https://www.whatismyip.com` is 47.242.230.X shown in Figure 4. The certificate of the SoftEther server is shown in Figure 5, which was issued to the domain “*.opengw.net” by Sectigo RSA Domain Validation Secure CA. This certificate has been used for all the volunteer nodes. From the experimental results, the client’s traffic has been hijacked and is redirected to the malicious server. A simple way to detect a TLS man-in-the-middle

attack is to check whether the server uses a certificate issued by a trusted CA, but in this attack scenario, the malicious node has the same certificate as a normal volunteer node does, even a skilled user cannot find the attack only by certificate checking.

In this scenario, the traffic of the attacked computer has been hijacked and redirected to a malicious server. The attacker may extract the user’s private data such as passwords, files, and access contents by analyzing the traffics and may also guide users to visit malicious websites through DNS hijacking. The malicious server needs to forward all the traffic of the attacked computer. When deployed on a large scale, an attacker will consume a large amount of bandwidth.

4.2. Scenario 2: Control-Channel MITM Attack. The scenario described in this section illustrates how an attacker hijacks the VPN session with security issue mentioned in Section 3.2. The network topology of the experimental environment is shown in Figure 6.

In order to complete the control-channel hijacking and the UDPAcc-channel decryption experiments, we developed two software tools, `tlsproxy` and `udpdecrypt`. The `tlsproxy` is designed to execute on the malicious server with the Linux operating system. The `tlsproxy` is to establish a TLS connection with the client and the server, respectively, and is responsible for forwarding the control-channel messages between client and server. In the process of forwarding, the `tlsproxy` will parse the protocol message and get the `UDPAcc_KEY_Ver2` used by the UDPAcc-channel. Once the decryption key is obtained, the attacker can use the software `udpdecrypt` to decrypt the saved traffic of the UDPAcc-channel offline.

The attack can be launched by following steps. Step 1 is executing the `tlsproxy` on the malicious server. Step 2 is adding DNAT rules in the malicious OpenWrt router to redirect the control-channel traffics to the malicious server. To do this, we use the following Linux Shell command: `iptables -t nat -A PREROUTING -d 219.100.37.X --dport 443 -p tcp -m tcp -j DNAT --to-destination 47.242.230.X:443`

Step 3 is logging the UDPAcc-channel traffic with `tcpdump` on the malicious OpenWrt router. Step 4 is parsing the log file of `tlsproxy` to get the decryption keys. And finally, Step 5 is using the decryption key to decrypt the logged traffic with `udpdecrypt`.

The client will see the address of the real VPN server when querying the IP address after the attack is completed. To check the server’s certificate in the client software interface, you will see that the server uses a self-signed certificate as shown in Figure 7. The UDPAcc-channel traffic decryption result is shown in Figure 8. Experimental results have proved that when an attacker implements a man-in-the-middle attack, the encryption keys of the UDPAcc-channel will be leaked, and the attacker can decrypt the communication traffic.

In this scenario, the attacker only needs to forward the control-channel traffic and does not need to forward the

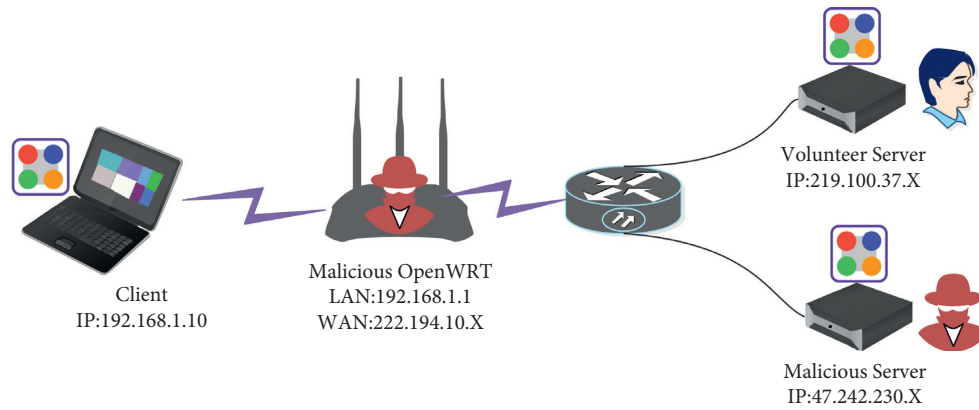


FIGURE 3: The network topology of attack scenario 1.



FIGURE 4: The result of IP address querying.



The following information is available for this certificate.



FIGURE 5: The certificate of the malicious server.

UDPAcc-channel traffic, which consumes less bandwidth resources than the previous scenario and may be deployed on a large scale.

4.3. Discussion of the Attacks. We introduced two attack scenarios in this article and introduced the experimental environment. The differences between the two scenarios are as follows:

- (1) *Different Complexity.* The MitM attack of scenario 1 occurred in IP layer, but the MitM attack of scenario 2 occurred in TLS layer. In scenario 1, the attacker does not need to forge a certificate or issue a self-signed certificate. In scenario 2, the attacker needs to rely on a self-signed certificate to implement a TLS

man-in-the-middle attack. Scenario 1 has the lower complexity.

- (2) *Different Bandwidth Cost.* In scenario 1, the attacker need to forward all the traffic of VPN user, but in scenario 2, the attacker only needs to forward the control-channel traffic, without needing to forward the UDPAcc-channel. This means that attack scenario 2 needs to consume a very small amount of bandwidth and does not need to forward the data transmitted by the user. Scenario 2 is more likely to be implemented by government censorship firewall or ISPs for its lower cost.
- (3) *Different Mitigate Method.* As far as we are concerned, the VPN Gate system needs to upgrade the server software to prevent attacks in Scenario 1 and

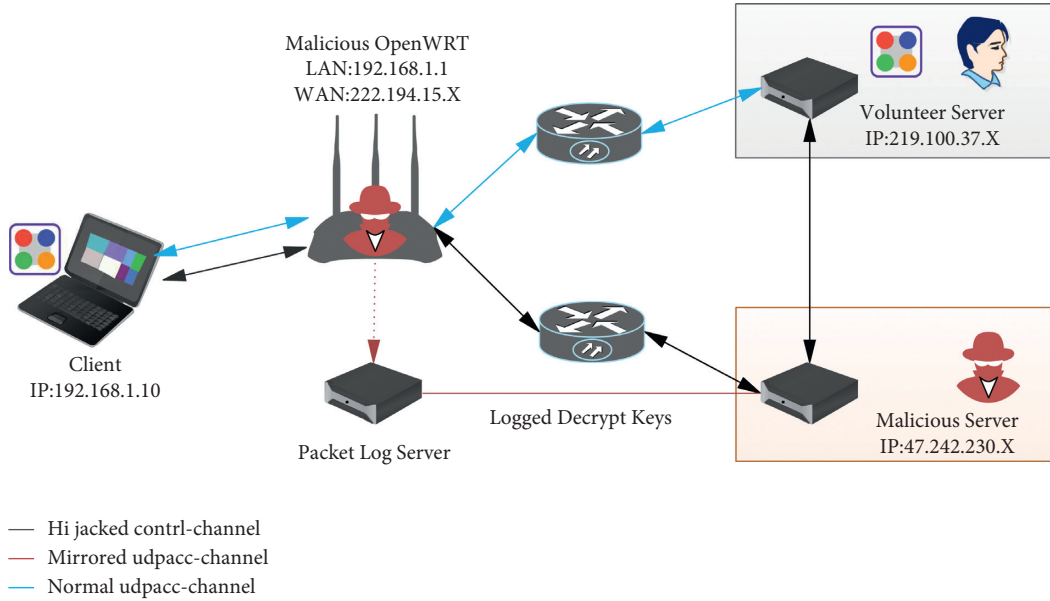


FIGURE 6: The network topology of attack scenario 2.

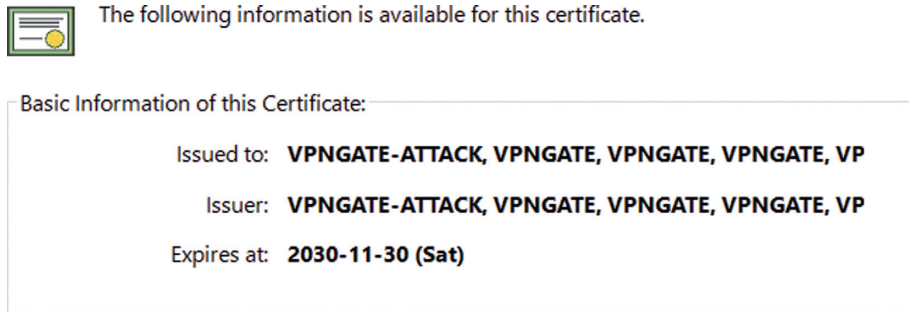


FIGURE 7: The untrusted certificate used by the attacker.

```

root@hitwh:~/udpdecrypt# ./udpdecrypt
0000 - e2 95 6c d7 00 00 00 00-16 e8 45 8d 00 00 00 04 ..l.....E....
0010 - 28 a0 ac 43 00 b3 00 01-00 5e 7f ff fa 5e d6 56 (.C.....^...^..V
0020 - d5 05 fd 08 00 45 00 00-a5 85 bd 00 00 04 11 74 .....E.....t
0030 - 9c 0a f1 c1 03 ef ff ff-fa e9 67 07 6c 00 91 5f .....g.l.._
0040 - eb 4d 2d 53 45 41 52 43-48 20 2a 20 48 54 54 50 ..M-SEARCH * HTTP
0050 - 2f 31 2e 31 0d 0a 48 6f-73 74 3a 20 32 33 39 2e /1.1..Host: 239.
0060 - 32 35 35 2e 32 35 35 2e-32 35 30 3a 31 39 30 30 255.255.250:1900
0070 - 0d 0a 53 54 3a 20 75 72-6e 3a 73 63 68 65 6d 61 ..ST: urn:schema
0080 - 73 2d 75 70 6e 70 2d 6f-72 67 3a 64 65 76 69 63 s-upnp-org:devic
0090 - 65 3a 49 6e 74 65 72 6e-65 74 47 61 74 65 77 61 e:InternetGatewa
00a0 - 79 44 65 76 69 63 65 3a-31 0d 0a 4d 61 6e 3a 20 yDevice:1..Man:
00b0 - 22 73 73 64 70 3a 64 69-73 63 6f 76 65 72 22 0d "ssdp:discover".
00c0 - 0a 4d 58 3a 20 33 0d 0a-0d 0a 00 00 00 00 00 00 .MX: 3.....
00d0 - 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 .....
00e0 - 00
Tag Verify Successful!

```

FIGURE 8: The decrypted Ethernet frame encapsulated in UDPAcc-channel.

upgrade the client software to prevent attacks in Scenario 2.

The technology used in our attack is not a new idea or invention. We believe that although cryptography experts have designed a sufficiently secure communication protocol based on reliable theories, network security in the

real world ultimately depends on the implementation of software systems. We just try to demonstrate how a simple and traditional trick can break a VPN system through the attack scenarios so as to attract more researchers to jointly improve the security of the software system in the real world.

5. Conclusion and Future Work

This paper has conducted a preliminary research on the SoftEther protocol from the perspective of cryptography and has discovered the risks of man-in-the-middle attacks in the VPN Gate system. Two attack scenarios are constructed to confirm the feasibility of the attack. The experiment results show that an attacker may decrypt the traffic easily. People who use public VPN service provided by the VPN Gate relay system may suffer from a high risk of privacy leakage.

According to the system defects found in this paper, we give the following suggestions for the VPN Gate users. We recommend that after one successfully connects to a VPN server, one should check the server's certificate manually with the information shown on the software interface. When an untrusted certificate is found, the connection should be immediately disconnected and the VPN service provided by this server should no longer be used forever. Meanwhile, users should check whether the IP address is in consistence with that of the target VPN server or not and stop using it when you find that the IP addresses are inconsistent. When using the VPN Gate, transmission of sensitive information should be avoided. We also hope that the SoftEther developers can publish protocol specification documents and encourage security researchers to write protocol specifications for the SoftEther protocol based on the SoftEther project source code so that subsequent researchers can quickly understand the details of the protocol.

The analysis was current as of November 22, 2020, and may not reflect recent system changes. We have not fully understood all the details of the SoftEther protocol in UDP mode nor have we built a symbolic model for the SoftEther protocol. These will be covered in our future research work.

Data Availability

The data used and/or analyzed in this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported in part by the "National Key R&D Program of China" (2020YFB2009502).

References

- [1] Y. Wang, P. Ji, B. Ye, P. Wang, R. Luo, and H. Yang, "Gohop: personal vpn to defend from censorship," in *Proceedings of the 16th International Conference on Advanced Communication Technology*, pp. 27–33, IEEE, Pyeongchang, Korea (South), 16 February 2014.
- [2] P. Zhang, M. Durresi, and A. Durresi, "Mobile privacy protection enhanced with multi-access edge computing," in *Proceedings of the 2018 IEEE 32nd International Conference on Advanced Information Networking and Applications (AINA)*, pp. 724–731, IEEE, Krakow, Poland, 16 May 2018.
- [3] M. Ikram, N. Vallina-Rodriguez, S. Seneviratne, M. A. Kaafar, and V. Paxson, "An analysis of the privacy and security risks of android vpn permission-enabled apps," in *Proceedings of the 2016 Internet Measurement Conference, IMC '16*, 14 November 2016.
- [4] D. Nobori and Y. Shinjo, "VPN gate: a volunteer-organized public VPN relay system with blocking resistance for bypassing government censorship firewalls," in *Proceedings of the 2014 11th USENIX Symposium on Networked Systems Design and Implementation (NSDI 14)*, pp. 229–241, USENIX Association, Seattle, WA, April 2014.
- [5] "Softether vpn project," 2021, <https://www.softether.org/>.
- [6] "Vpn gate project," 2021, <https://www.vpngate.net/>.
- [7] S. Meier, B. Schmidt, C. Cremers, and D. Basin, "The tamarin prover for the symbolic analysis of security protocols," in *Computer Aided Verification*, N. Sharygina and H. Veith, Eds., Springer, Berlin, Heidelberg, pp. 696–701, 2013.
- [8] D. Adrian, K. Bhargavan, Z. Durumeric, P. Gaudry, and M. Green, J. Alex Halderman, N. Heninger, D. Pringall et al., "Imperfect forward secrecy: how diffie-hellman fails in practice," in *Proceedings of the 2018 22nd ACM SIGSAC Conference on Computer and Communications Security, CCS '15*, Association for Computing Machinery, New York, NY, USA, December 2018.
- [9] D. Felsch, M. Grothe, Joerg Schwenk, C. Adam, and M. Szymanek, "The dangers of key reuse: practical attacks on ipsec IKE," in *Proceedings of the 27th USENIX Security Symposium (USENIX Security 18)*, pp. 567–583, USENIX Association, Baltimore, MD, USA, August 2018.
- [10] M. T. Khan, J. DeBlasio, G. M. Voelker, A. C. Snoeren, C. Kanich, and N. Vallina-Rodriguez, "An empirical analysis of the commercial vpn ecosystem," in *Proceedings of the Internet Measurement Conference 2018, IMC '18*, October 2018.
- [11] "Vpn gate attack poc," 2021, <https://github.com/HITWH-INET/vpngate-mitm>.
- [12] "Softether vpn source code," 2021, <https://github.com/SoftEtherVPN/SoftEtherVPN>.