

Secure distributed session keys for new endpoints

Distributed key system for distributing more keys securely

Albert Meyburgh
British Columbia Institute of Technology
Practicum
ameyburgh@gmail.com

Abstract - Traditionally distributed key systems required that a key be delivered through courier or in person to each person that you wish to establish a secure link with. The approach herein has eliminated this encumbrance. At any time, you can start communicating to someone else that uses this framework without having to wait for a distributed key to be delivered.

A distributed key is a key that has been pre-distributed by some manual means to the parties involved. This is the most secure method of ensuring key privacy. However, this is a problem when new dynamic sessions need to be established between a party with a pre-distributed key and others who do not have pre-shared key information.

This configuration uses the distributed key, not as a key for a point-to-point link, as would traditionally be done, but instead that key is used to distribute encrypted “session” keys to be used for the original intention of establishing secure links of communication. Distributed keys by their nature, not only allow for the encryption of traffic, but also the authentication of the other party.

Keywords - Whitenoise; GateKeeper, Key Vault, one time pad; encryption; authentication

I. INTRODUCTION

This secure tunnel system is composed of two applications called the GateKeeper and the KeyVault. They work together to create a dynamic distributed key environment for TCP/UDP tunneling. The GateKeeper creates and encrypts tunnels based on simple standard netfilter rules while the KeyVault facilitates the retrieval of point-to-point keys as required by GateKeepers as they talk to each other. The system facilitates near-transparent, dynamic, encrypted point-to-point communication between networks on a network.

This application uses the Whitenoise Superkey Encryption Algorithm (WSEA) [1] [2], a new generation symmetric stream cipher. Stream ciphers convert plaintext to ciphertext one bit at a time. WSEA, completely random and non-repetitive, can encrypt never-ending streams of communications traffic.

Whitenoise encryption provides the highest level of security with the greatest speed with virtually no overhead and with no latency in telecommunications of all kinds. Whitenoise can be deployed either in software or on silicon.

The KeyVault and GateKeeper systems work together to create a layer on any IP based network like the Internet that allows communications to remain secure and confidential. The innovative component is the implementation of a dynamic distributed key system.

II. DEPLOYING THE SYSTEM

The GateKeeper and KeyVault servers can be used in any tier of network architectures traveling from IP to IP. It can be used from computer to computer, network to network, or computer to network. It can be used in any IP to IP context: wired-to-wired, wireless-to-wired, and wireless-to-wireless. These applications can work separately and independently but they obviously compliment each other.

The system is able to plug anywhere into a network easily because it relies on the data link layer between systems. Using the data link layer instead, allows immediate integration with every IP based application with no delay. The applications don't even know that the tunnel is there at all.

Some other encryption systems rely on the application level (SSH is an example of this). When the application level is used, the secure tunnel is application specific and needs to be re-integrated with each application that wishes to utilize it.

There is a simple, hybrid TLS-WN-DIVA extension for openssl and LDAP/CAS (Microsoft) [3] which is an example of an approach that overcomes many of application level limitations by incorporating a Transport Layer Security extension.

The GateKeeper tunneling system could be used on its own to only facilitate the traditional notion of static point-

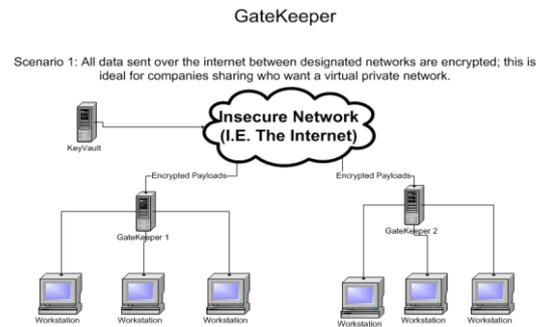
to-point tunnels that would be useful for ISPs, governments, embassies, or corporations.

The KeyVault architecture to distribute session keys based on a distributed key allows point-to-point dynamic connections that can be applied on other areas of the network apart from the tunnel.

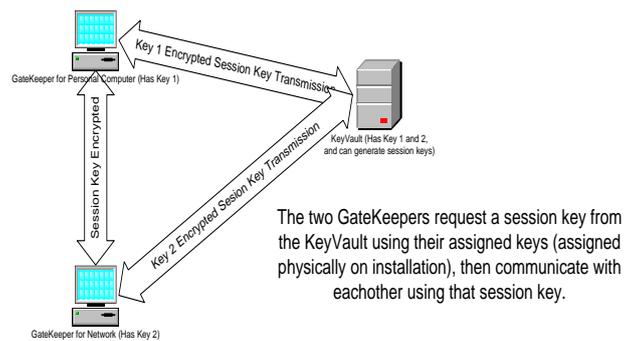
No one GateKeeper can decrypt arbitrary data. When encrypted data needs to be decrypted, only the destination computer can decrypt it since only the two computers involved in the transmission can obtain the session keys from the KeyVault. The session keys are encrypted by a unique key pairing with the key vault. The GateKeeper client creates and encrypts the request for the session key with the other GateKeeper with its private distributed key. Only the KeyVault that holds the session key has a copy of that private key. Only the two GateKeepers involved in the session could request the session key since their private keys authenticate their requests with the KeyVault.

The sequences of events that drive a secure link start with the GateKeeper on the initiating side, move on to the KeyVault, and finally end at the receiving side. This can be seen in the subsequent diagram. The GateKeeper and the KeyVault work together to form the distributed key system in establishing secure point-to-point communication. The GateKeeper communicates through tunnels to other GateKeepers using existing cached keys, and retrieves any needed session keys from the KeyVault as needed. The KeyVault simply receives and respond to key requests. The KeyVault application has one main loop that listens for incoming key requests, and fulfills the requests with key

responses.

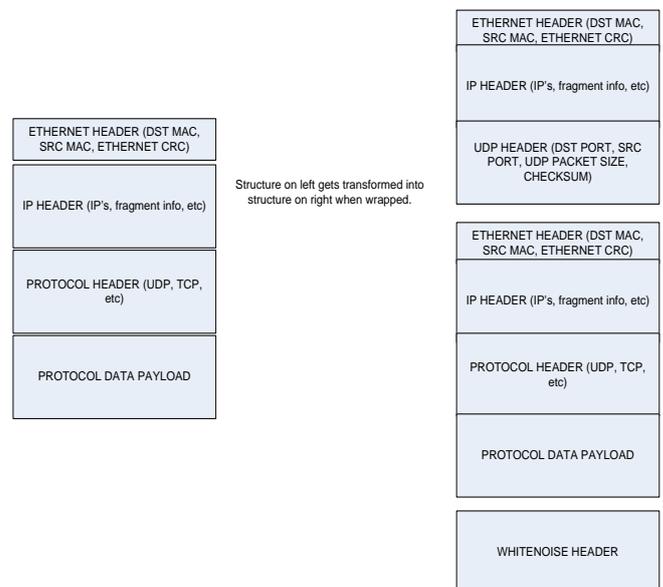


Distributed Key Paradigm: Each GateKeeper will have a unique key-pairing with it's KeyVault



The actual composition of the encapsulated packet is as follows:

Unwrapped VS Wrapped Packets



Once the packet has been encapsulated into the new packet with the Whitenoise header, the embedded packet can be encrypted with the appropriate session key.

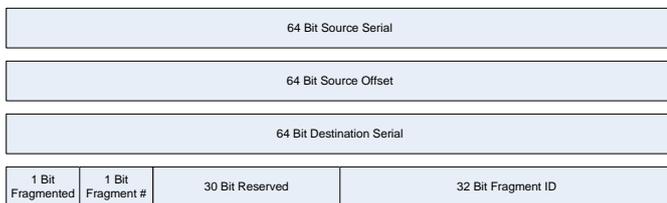
The reasons UDP packets were chosen to encapsulate the encrypted traffic are twofold:

UDP is the only common protocol that includes the data size in the protocol, thereby allowing additional headers to be appended

Since this is a tunnel protocol, if any re-transmission of data is required, the clients can request it. It is not needed for the tunnel to keep track of lost data.

The Whitenoise header consists of information to use the encryption, and some information regarding fragmentation for when the tunnel needs to fragment the data packets due to the MTU being exceeded.

Whitenoise GateKeeper Tunnel Header



The first serial is the serial of the originating system, the second serial is the destination system serial, and the offset is the offset into the Whitenoise stream that was used to encrypt this particular packet.

The fragmented bit indicates if this is a fragmented tunnel packet, the 1 bit fragment number indicates if it's the first or second fragment, 30 bits have been reserved for an authentication pad and 32 bits are used for the fragment id used to distinguish these fragments to other fragments. There is a 1 in 2^32 chance that fragments may have overlapping fragment ids and this would corrupt the re-assembly.

This header consists of 256 bits plus the additional Ethernet, IP, and protocol headers in the encapsulated packet. This makes up the overhead in the tunnel system. This overhead is per packet, so if many small packets are sent out,

then the percentage overhead is relatively large. However, if large packets from file transfers are used then the overhead is very low.

III. IMPLEMENTATION IMPLICATIONS

There are some implications in implementing a secure tunneling system combined with the KeyVault system. Not only does the system create a secure point-to-point communications layer but it also provides a way for dynamically adding new GateKeepers to the system without having to copy the key manually to every other client before communication can commence. Additionally it is satisfying the authentication requirement.

The problem with SSH (an alternative secure tunnel system) for example is that it is vulnerable to man-in-the-middle attacks. Man-in-the-middle attacks, MiM, include any type of network attack where your information could pass through the hands of a hacker without you even realizing it.

Distributed keys, by their very nature destroy the possibility of a MITM attack; since, an unencrypted key exchange never occurs there is never a chance for a hacker to intercept or spoof the keys.

Secure communication channels may be mandated at an organization or perhaps authentication is the goal. Imagine a scenario where a company only accepts email from other companies that have their email servers connected to a central KeyVault authority. Anyone that sends and receives email through this system is guaranteed knowledge of the e-mail's server of origin. This enables trusted authentication of email which is difficult at the moment.

Since the system relies on Berkeley packet filter type expressions to determine the types of packets read, this system could be easily integrated with firewall features.

IV. TRADITIONAL ATTACKS

There are many traditional data link based attacks used to steal data in transit on a local area network. These

attacks are often oriented around intercepting data and viewing communications that wasn't meant for them. Some attacks are performance hit oriented like DDOS.

The two categories of techniques for stealing data usually involve either pure pass-through interception of the traffic, or alternatively some sort of authentication simulation. In an SSH handshake for example, the man-in-the-middle (attacker) would shake hands with both parties making them believe the MiM is actually the other party.

In an ARP poisoning attack on the other hand, there is no handshaking necessary. The attacker simply redirects the traffic to his machine and is able to look at it in a packet sniffer.

Disabling non-encrypted traffic is of course an option in the GateKeeper system. However, this is not practical for most environments since people need to send email outside of the company and surf the web. In some situations like in hospitals and military, perhaps even corporate research facilities, the need for security may be great enough that the GateKeeper would drop all non-encrypted traffic.

V. CONCLUSION

There is a need for rapidly scalable distributed platforms where there is only partial exposure of credentials. Distributed key systems previously required that a key be pre-authenticated and pre-distributed. The approach discussed enables the establishment of secure point-to-point communications with a party or endpoint that does not have a pre-distributed key.

The GateKeeper and KeyVault is a scalable symmetric key authenticated encryption system where no single trusted third party knows the final key.

This distributed key architecture advances trusted cyber and trusted computing. It can be used to create a hybrid

system without direct integration with asymmetric public key infrastructures (PKI).

A dynamic distributed key extension eliminates system vulnerability to man-in-the-middle attacks. It has no exposed public key and mitigates against known attacks particularly quantum computing attacks because of the use of Whitenoise keys.

VI. ACKNOWLEDGEMENTS

André Brisson and Stephen Lawrence Boren, founders of Whitenoise Laboratories Canada Inc., were my practicum advisors and guided this research while I was at the British Columbia Institute of Technology.

VII. REFERENCES

- [1] Albert Meyburgh, Distributed keys securely sharing session keys - http://www.wnlabs.com/downloads/Tunnel_Distributed_Keys_distributing_more_keys.pdf
- [2] André Brisson – Differences between PKI and DDKI handshakes - http://www.wnlabs.com/pdf/Comparison_of_handshakes.pdf
- [3] <http://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-57pt1r4.pdf>
- [4] Albert Meyburgh, Distributed keys securely sharing session keys - http://www.wnlabs.com/downloads/Tunnel_Distributed_Keys_distributing_more_keys.pdf
- [5] André Brisson Factorization of a prime number composite https://www.youtube.com/watch?v=GwkWgR_78dQ&feature=youtu.be
- [6] André Brisson “Rapid Factorization of Semi Primes,” http://www.wnlabs.com/pdf/Rapid_Factorization_of_semiprimes.pdf
- [7] D. Wagner, “A Security Evaluation of Whitenoise,” University of California, Berkeley, Oct. 2003. [Online]. Available: <https://eprint.iacr.org/2003/218.pdf>
- [8] I. Traore and M.Y. Liu, “Evaluation of Whitenoise Cryptosystem. Part 1: Encryption Algorithm,” Technical Report ECE03-3, University of Victoria, British Columbia, Canada, Feb. 2003. http://www.wnlabs.com/downloads/UVIC_Performance_Analysis.pdf
- [9] A.J. Brisson, “Cyber Belt presentation for NSA/NIST 2016 – Quantum computing attack resistant security and the cyber belt.” http://www.wnlabs.com/pdf/Cyber_Belt_Presentation.pdf
- [10] S.L. Boren and DW (CSE), Mathematical rebuttal refuting false break technique, <http://www.wnlabs.com/pdf/Response.pdf>
- [11] Laurie Perrin and A.J. Brisson, TLS Whitenoise DIVA extension, http://www.wnlabs.com/technology/WNL_TLS_extension.php
- [12] André Brisson, “Key distribution method to shield service providers from key management - <https://www.linkedin.com/pulse/key-distribution-paradigm-removes-management-from-carriers-brisson>
- [13] A.J. Brisson, “Factorization of a prime number composite” https://www.youtube.com/watch?v=GwkWgR_78dQ&feature=youtu.be
- [14] A.J. Brisson, “Rapid Factorization of Semi Primes,” http://www.wnlabs.com/pdf/Rapid_Factorization_of_semiprimes.pdf

- [15] A.J. Brisson, "The Whitenoise Algorithm – A Visual Look," Technical Report, Whitenoise Laboratories, Vancouver, British Columbia, Canada, Nov. 2011. [Online]. Available: <http://www.wnlabs.com/pdf/WhitenoiseAlgorithmVisualLook.pdf>
- [16] C. Coram-Mekkey, "Whitenoise Laboratories – An Overview," White Paper, Whitenoise Laboratories, Vancouver, British Columbia, Canada, June 2015. [Online]: <http://www.wnlabs.com/papers/WhitenoiseOverviewShort.pdf>
- [17] S.L. Boren and A.J. Brisson, "Dynamic Distributed Key System and Method for Identity Management, Authentication Servers, Data Security and Preventing Man-in-the-Middle Attacks," U.S. Patent Application 2009/0106551 A1, Apr. 2009.
- [18] A.J. Brisson, "Dynamic Identity Verification and Authentication, Dynamic Distributed Key Infrastructures, Dynamic Distributed Key Systems and Method for Identity Management, Authentication Servers, Data Security and Preventing Man-in-the-Middle Attacks, Side Channel Attacks, Botnet Attacks, and Credit Card and Financial Transaction Fraud, Mitigating Biometric False Positives and False Negatives, and Controlling Life of Accessible Data in the Cloud," U.S. Patent Application 2013/0227286 A1, Aug. 2013.