On Using One-Time Pad (OTP) Encryption with Hardware Network Encryptors to Provide For-All-Time (FAT) security

Jonathan Sanderson
jjs29356@gmail.com
2024/06/25

Purpose

The purpose of this white paper is the public disclosure of a novel idea (invention) for using One-Time Pad (OTP) encryption with Hardware Network Encryptors (HNEs) to provide a For-All-Time (FAT) secure connection between two end points that is immune to harvest now decrypt later attacks. This invention is being released as public domain knowledge so to disallow any person, persons, groups, cooperations, governments, or any other entity from claiming sole or partial exclusive rights to implement, patent, sell, extend, or any other way use the invention and variations disclosed herein.

Terminology

- **FAT Secure** - For-All-Time Secure. A term used in this paper to denote immunity to harvest now decrypt later attacks.
- **Symmetric Key** - A key used for encryption that both paries must have identical copies of.
- **Asymmetric Key** - A key used for encryption that both parties have partial knowledge of.
- **OTP** - One-Time Pad. An extremely long encryption key.
- **HNE** - Hardware Network Encryptor. A device used to encrypt network traffic and secure communications.
- **OKG** - One-time pad Key Generator. A device with dedicated hardware for efficiently and securely creating large One-Time Pads.
- **Packet/Frame** - A unit of data that is sent/received in a network system.
- **MAC** - Message Authentication Code. A piece of metadata sent alongside a message that is used to verify the authenticity of a message.
- **Chunk** - A term used here to describe the minimum amount of data encrypted/decrypted by OTP operation. The size of a chunk is arbitrary.
- **Block** - The minimum size of data that a storage device can read or write. The size is dependent on the storage media used.

1. Introduction

The goal of cryptography is allowing two or more entities the ability to communicate with each other while disallowing all other entities from deciphering/understanding the communication. This is accomplished via a shared secret or part of a shared secret. There are two ways of encrypting something, using either a shared key (symmetric key cryptography) or part of a key is shared (asymmetric key). Asymmetric key cryptography is commonly used to negotiate a commonly shared key without the worry of a "middle man" being able to recover the secret. Standards for this, like RSA, Diffie-Hellman, Elliptic key cryptography, and similar standards are commonly used for transactions on the internet. Until recently, these standards were thought to adequately secure. However, with the growing progress of quantum computers, it is generally understood that these standards will be able to be broken by quantum computers in the future [1]. The day when quantum computers are able to break common asymmetric algorithms is referred to as Q-day. Regardless of when Q-day happens, the threat vector applies to today, due to harvest now decrypt later attacks, where, communications over the internet today can be stored till a future date when the encryption can be cracked. While symmetric encryption schemes like AES are known to be quantum resistant today, it is not possible to know whether vulnerabilities will be discovered in the future by quantum computers or some technology not yet discovered.

There is one encryption scheme that if used correctly can guarantee an encrypted message is fundamentally unbreakable. This encryption scheme is known as One-Time-Pad (OTP). The scheme does have several caveats, mainly that it is symmetric key based, and the key must be at least as long as the message being sent. These pose special challenges for securely managing and using OTP. As with all encryption techniques, if a system implementing them is not sufficiently secure, such as a software bug, zero day software vulnerabilities or user error, the security of the algorithm used becomes a moot. In order to provide the highest level of security using OTP, For-All-Time (FAT) security, specialized hardware is needed to minimize the scope of possible attack vectors.

This paper seeks to provide concepts and details related to creating a dedicated Hardware Network Encryptor (HNE) that provides FAT security to network traffic between two secured points connected via an unsecured network (internet). The rest of this paper is broken into a background around OTP and HNEs, threat scenario and proposed mitigations, a summary of the conceptual invention and finally a section detailing how a possible system(s) could be implemented.

2. Background

2.1. One-Time Pad

One-Time-Pads (OTPs) saw some use in the Cold War, where absolute security was paramount. One-Time-Pad shares commonalities with a Caesar cipher and a Vigenère cipher as they are a substitution cipher. However, instead of using a constant or changing substitution key, with OTPs the substitution is completely random. More specifically, the key is a large pad of pre-generated values. In order for OTP to be FAT secure, it must meet four criteria [2]:

1. **The key must be at least as long as the plain text.**
2. **The key must be truly random.**
3. **The key must never be reused in whole or in part (key is destroyed once used).**
4. **The key must be kept completely secret.**

Thus, a device providing FAT security must facilitate these for four rules and minimize human error as much as possible.

2.2. Hardware Network Encryptors

Modern Operating Systems (OS) have numerous software subsystems, and are very complex. This complexity allows for many different attack vectors, such a memory leaks, kernel bugs, privilege escalations, and may others. These technical issues are compounded by social engineering attacks that can result in tricking users to install malware, compromising

the whole system. A simple solution to minimizing these attach vectors between two computers is by simply using a piece of dedicated hardware (HNE) to ingest network traffic, encrypt it, and transmit the encrypted network traffic to be eventually received and decrypted by another HNE. By encrypting all traffic passing through the HNE mitigates several vulnerabilities related to malformed packets, improper routing configuration, user error, etc. HNEs can operate on various levels of the Open Systems Interconnection (OSI) model. Typically, they operate on Layers 2, 3, or 4, shown in Figure 1. When operating on layers 2 and 3 HNEs essentially create a virtual private network. Layer 4 is sometimes used due to less overhead of being able to directly encrypt packets without having to convert the data to a higher layer level to be properly routed to their destination.

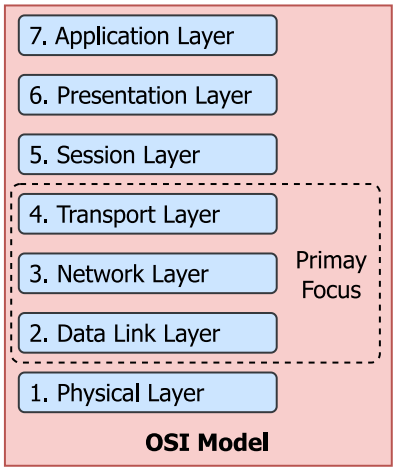

Figure 1: OSI Model

2.3. Related Patents

- The substitution used in OTP can be computed extremely quickly by computers via the binary XOR operation. This concept was patented by Gilbert Vernam in 1919 [3].
- In 2022 a patent [4] was granted for using OTP with a software defined network where smart switches linked by a controller to encrypt network traffic dynamically. Dynamically being that the network packets would only be encrypted if they contained a flag in the packet header
- At the time of writing, there is currently a patent [5] pending for embedding OTP in Internet-of-Things (IoT) devices. The patent details a design where the OTP would last the lifetime of the IoT device and could not be changed after manufacturing.
- In 2017 a patent [6] was filed but is now abandoned that describes a way in which OTP could be used to encrypt network traffic.

3. Threat Scenario & Mitigations

3.1. Threat

1. Attacker monitoring network traffic and employing harvest now decrypt later attacks.
2. Attacker gaining access to OTP key during exchange (before OTP is used).
3. Attacker gains access to OTP key after it has been used.
4. Attacker gaining physical access to devices carrying out encryption of network traffic.

3.2. Mitigation

1. Anything sent over a public network is considered high risk, subject to monitoring, interception, storage and tampering. Whenever sending data, an HNE encrypts it with OTP and traditional methods of encryption, thus make decryption impossible without obtaining the OTP key. When receiving data, the authenticity should always be verified before attempting to decrypt the data (encrypt then MAC).
2. The OTP keys are exchanged via the physical transport of storage devices. This makes it much more difficult for an attacker to have physical access to the storage device than

intercepting network traffic. However, as a measure added security, devices containing OTPs are encrypted using traditional storage methods.

3. To mitigate this, and maintain FAT security, HNEs use a “scorched earth policy”, as soon an OTP (or part of an OTP is used), it is destroyed shorty after.
4. Not all attack vectors can be mitigated when an attacker has physical access to an HNE device. To reduce the possibility of recovering sensitive data, Trust Platform Modules (TPMs) are used to store sensitive information.

4. Summary Of Invention

The invention is as follows:
Creating custom HNE to use OTP to encrypt all network data passing through it. Data is can be sent between two HNEs that have been paired. The two HNEs do not need to be synchronized or controlled by a central system. Two HNEs are designed to work as a peer-to-peer or a single client and single server configuration. HNEs are not to route network traffic, only encrypt it. Multiple methods of encryption are performed in addition to OTP, using symmetric and/or asymmetric keys. HNEs use commercially available digital storage media to hold a very large pre-generated OTP keys (in the GB to TB range). The network traffic can be ingested and encrypted at any level of the OSI model. The encrypted traffic encrypted by the HNE may or may not resemble the type of network traffic ingested and may be the result of a transformation to a different OSI layer. Two HNEs have an identical OTP keys stored on physical storage media, such as a HDD, SD card, SSD or any media that does not employ (wear-leveling), and that can be incrementally overwritten securely during operation (scorched earth policy). HNEs may employ one or multiple storage media to increase network throughput. This storage media uses encryption to secure the OTP on storage medium. As the OTP is used incrementally to encrypt network traffic the bytes used, are overwritten in real-time. Thus seconds after use the being used, the OTP is destroyed. HNEs can employ a counter to keep track of the location on the physical media where to read the OTP data. In the case of two HNEs de-syncing, due to packet drop or any other failure, the HNE with the OTP that should have been erased can be “fast-forwarded” and delete the OTP, once re-establishing a connection. HNEs can employ traditional verification methods such as a Message Authentication Code (MACs), to ensure data integrity and authenticity of packets received.

Generating OTPs in the GBs to TBs range takes a significant amount of time. To mitigate this, a additional device is proposed: OTP Key Generator (OKG). The OKG can be used to generate OTPs before they are needed. These storage devices containing OTPs are encrypted using traditional encryption methods. To securely synchronize symmetric keys between the OKG and HNEs, the devices are physically connected together in a secure environment. This need only be performed once in the lifetime of the devices.

5. Detailed Description

There are many factors that have to be considered in implementing a system utilizing OTP in order to maintain reliability and security. This section gives details of how such a system could be implemented in order to prove the proposed invention’s feasibility.

5.1. OTP HNE Device

In order to make the HNEs intuitive to use, reduce possible user error, the device enclosure should clearly indicate which ports should be connected to an untrusted network and and what ports are to be connected to a trusted network/device. It is recommended these ports be on opposite sides of the physical enclosure. An HNE can have one or multiple ports for trusted networks and one or multiple ports for untrusted networks.

To operate securely HNE should contain multiple hardware security components for secure operation. Each HNE should be equipped with a Central Processing Unit (CPU) for handling general operations including encryption and decryption of network traffic. HNE should have a fast interface for reading and writing to the storage containing the OTP. HNEs can contain a internal True Random Number Generator (TRNG) chip, that can be used for generating public private key pairs (for details on where this is used, see Section 5.6). To securely store all keys other than the OTP, a HNE should contain a Trusted Platform Module (TPM) chip. As an extra layer of security protection, HNEs should have a simple interface where a user can enter a session password to be used in the encryption/decryption process (see Section 5.4 and Section 5.5). An example of this system is shown in Figure 2.

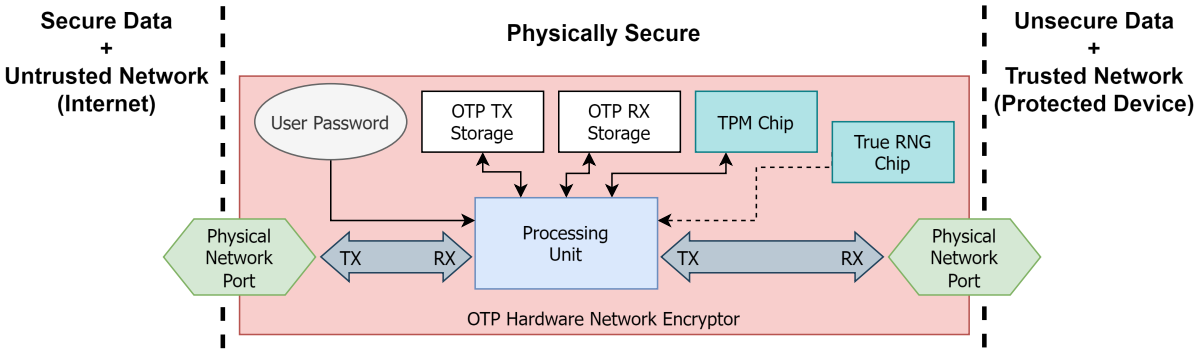

Figure 2: Example OTP HNE Device

5.2. OTP Key Generator Device

The OKG device is very similar to HNEs, but differences in a couple of aspects. The OKG is not used for encrypting/decrypting network traffic, but is solely for generating OTP pads thus it need not have network ports. The OKG should be able to accept four or more storage devices, two for each HNE device. The OKG **MUST** contain a TRNG to be used for generating OTPs. The OKG need not have a method of receiving a password from a user, as this is only needed when encrypting and decrypting network traffic.

5.3. Secured Packet Structure

As two HNEs will be sending and receiving packet over an untrusted network, they must have an agreed upon method of structuring data in packets. An example of such a structure is depicted in Figure 3.

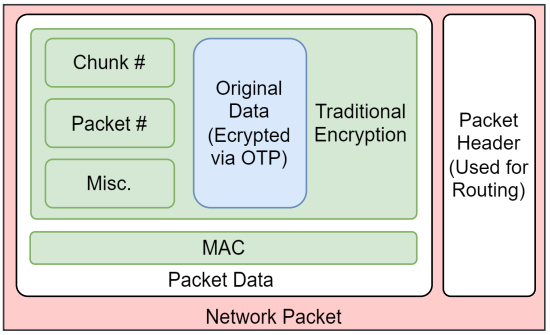

Figure 3: HNE example packet structure. The box in blue shows data encrypted using OTP. The items in green represent data encrypted using traditional encryption methods.

The data is doubly encrypted, once by OTP and once by traditional methods. This is needed as some metadata required for decrypting the OTP encrypted data, cannot by itself be encrypted via OTP. See section Section 5.5 for more details on how this metadata is used.

5.4. HNE TX Operation

The term TX is used to describe the direction when a network packet/frame is received from a trusted network, secured and transmitted to an untrusted network. A detailed diagram of the process for securing the communication can be seen in Figure 4.

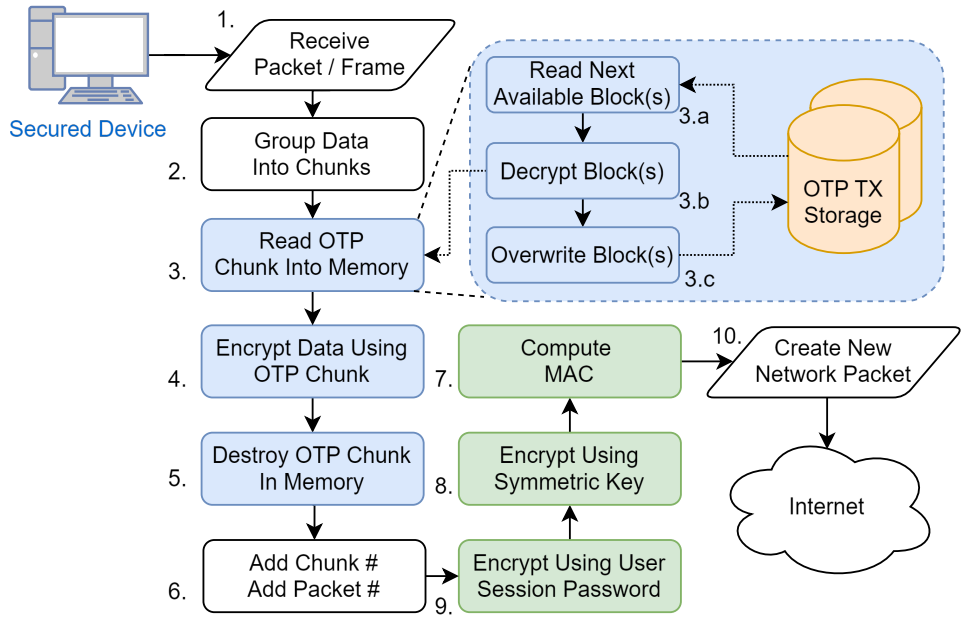

Figure 4: Example TX operation. The item in blue show operations dealing with OTP. The items in green represent operation using traditional security methods. The solid arrows represent the operation flow and the dashed arrows represent data flow.

The order of operations are:

1. Receiving of packets or frames.
2. Grouping of received data into chunks in order to optimize processing efficiency.
3. Reading part of the OTP stored into memory. The size of the OTP read into memory need only be the same as size as the chunk being encrypted. The reading from the storage requires several sub steps.
    a) Reading the appropriate number of blocks from the storage device.
    b) Decrypting these blocks using a key(s) stored in the TPM chip, to ascertain the OTP chunk.
    c) Once the OTP chunk is loaded into memory, a background process starts securely overwriting the blocks just read, destroying the OTP chunk kept on disk. This prevents possible recovery of data from the storage.
4. Encrypt the data chunk with the OTP chunk.
5. Destroy the OTP chunk from memory by overwriting it.
6. Add metadata needed for reconstructing chunks, such as the packet order and location on the storage device to read OTP data for decryption.
7. Encrypt chunk using user session password.
8. Encrypt using HNE symmetric key.
9. Compute validation MAC.
10. Package encrypted data and MAC into a network packet and send it to an untrusted network to be routed to the other HNE.

When transmitting data the HNE can use a protocol such as TCP to know the percent of packets that arrive at the other HNE. If two many packets are being dropped the HNE transmitting data can elect to send empty packets so the TX OTP is not waisted when reestablishing the connection.

5.5. HNE RX Operation

The operation of receiving and decrypting packets from an untrusted network is the reverse of TX operation but with extra steps for dealing with packet loss and validation must be accounted for to maintain a secure synchronized connection. A diagram of this can be seen in Figure 5.

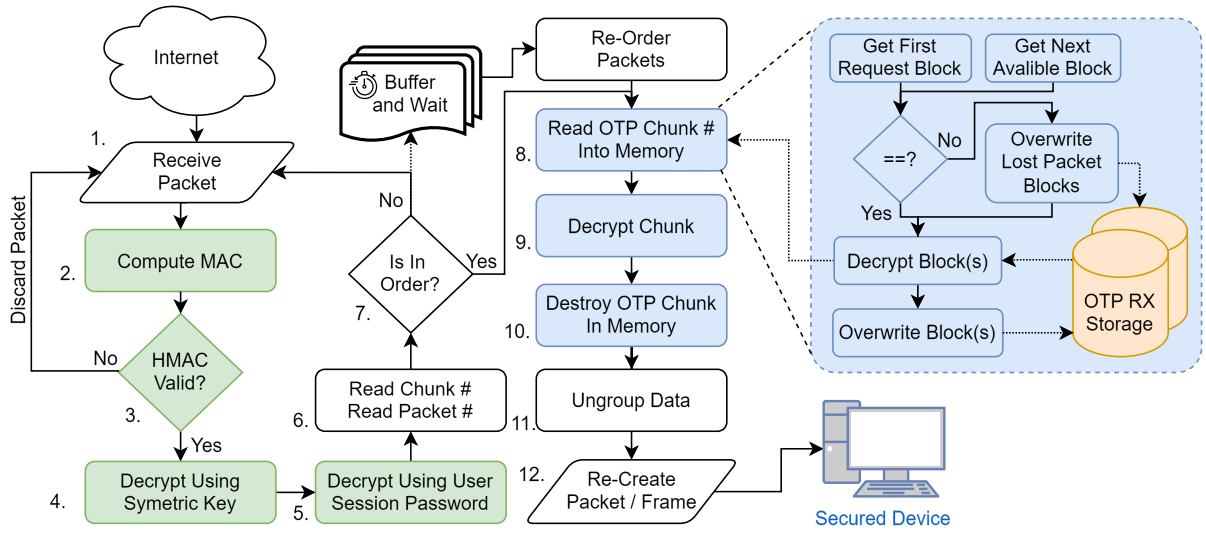

Figure 5: RX Operation. The item in blue show operations dealing with OTP. The items in green represent operation using traditional security methods. The solid arrows represent the operation flow and the dashed arrows represent the data flow.

The flow of operations is as follows:

1. A packet is received from an untrusted network.
2. The MAC of the data is computed.
3. The MAC is checked to be valid. If it is found to be invalid the packet is discarded (ignored).
4. The data is decrypted using the same symmetric key it was encrypted with.
5. The data is decrypted using the users' session password.
6. Metadata on packet order and chunk number is read.
7. Out of order data are buffered so they may later be re-ordered. If this is a result of a packet lost in transit, the data will eventually be discarded.
8. Once the data has been correctly ordered into a chunk, the storage is queried for an OTP chunk. This process is very similar to the one described in Section 5.4, however if packets are lost in transit, several blocks of the drive must be overwritten (destroyed) to "fast-forward" to the correct OTP chunk being read.
9. The data is decrypted via the OTP chunk.
10. The OTP in memory is destroyed.
11. The data is ungrouped to remble the data originally send by the trusted device.
12. The data is packaged into a packet or frame that is identical to the one received during the first step of the TX operation, and is sent to the trusted network / device.

5.6. Key Exchange

A major aspect to the security of this system is the exchange of symmetric keys between the OKG and HNEs. The key synchronization should be with all devices physically connected to each other over a cable and air gapped. The key exchange only needs to be performed once, as once completed, multiple OTPs can be generated and encrypted with the same keys (assuming all the devices are kept physically secured). A diagram of a potential way to do the transfer is shown in Figure 6.

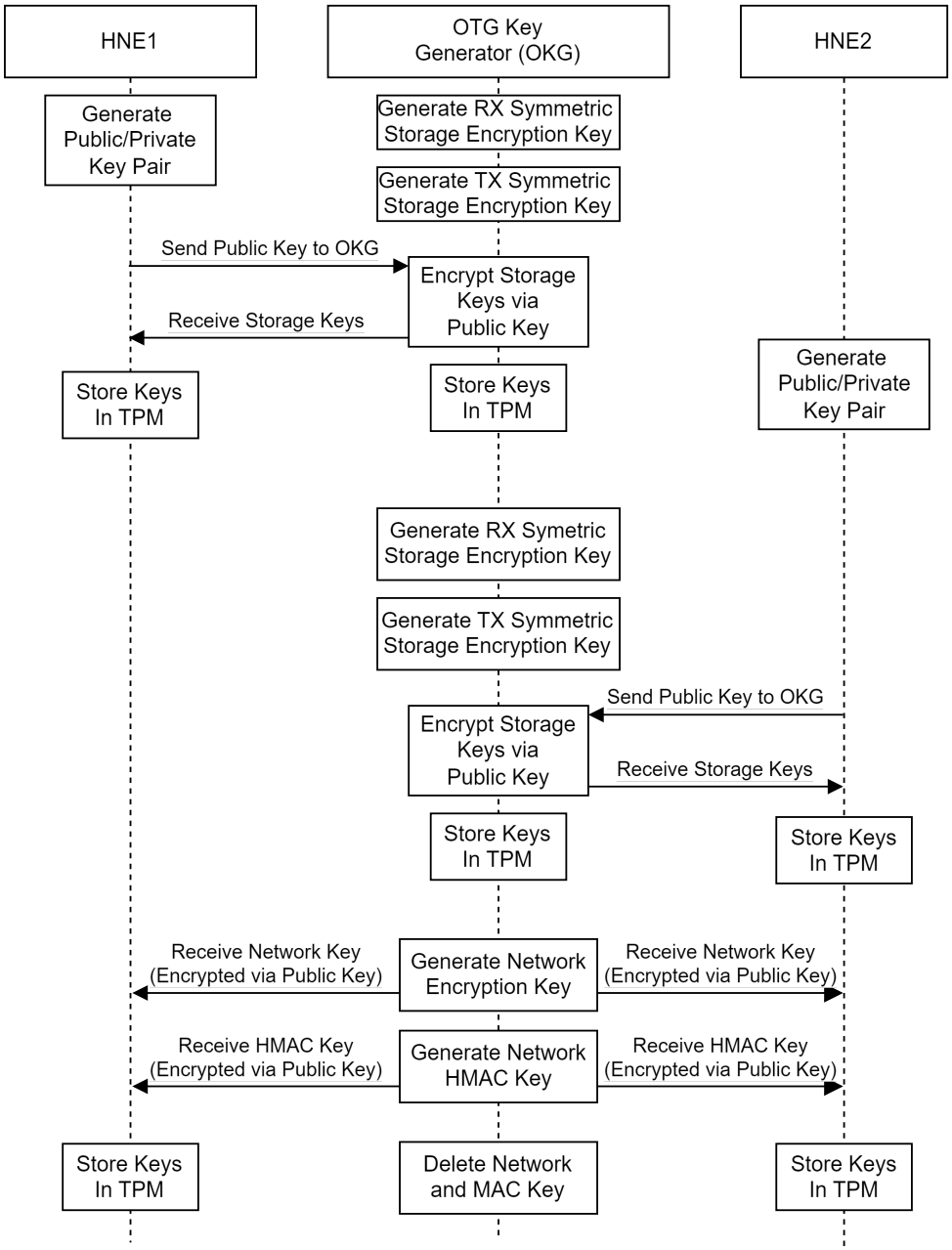

Figure 6: Proposed Sequence diagram for syncing keys between two HNEs and an OKG

Once the HNEs and OKG have synced keys, they can be transported to any secure physical location desired. The OKG can then be used to fill up storage devices with OTPs. The storage devices are encrypted with both the symmetric storage key and the HNEs' public keys, thus making it extremely difficult for an attacker to gain access to the OTP even with access to the physical storage device.

References

[1] CISA, [Online]. Available: https://www.cisa.gov/news-events/alerts/2024/06/18/cisa-and-partners-release-guidance-modern-approaches-network-access-security

[2] D. Rijmenants, [Online]. Available: https://ciphermachinesandcryptology.com/papers/one_time_pad.pdf

[3] G. Vernam, "Secret signaling system," no. US Patent 1,310,719. Jul. 1919.

[4] W. A. Sellers, J. M. Mengert, and others, "One-time pad encryption in a secure communication network," no. US Patent 11,388,153. Jul. 2022.

[5] P. G. Hunt and M. V. Bertolina, "One-time pad encryption for industrial wireless instruments," no. US Patent App. 17/420,652. Google Patents, Mar. 2022.

[6] T. A. Tomkow, "One-time pad communications network ," no. US Patent Application US20170180117A1. Mar. 2017.

[7] M. Blumenthal, "Encryption: Strengths and weaknesses of public-key cryptography." [Online]. Available: http://www.csc.villanova.edu/~mdamian/Past/csc3990fa08/csrs2007/01-pp1-7-MattBlumenthal.pdf