2020

The Key Distribution Problem: Prior Advances and Future Challenges

Benjamin Lee

2020

The Key Distribution Problem: Prior Advances and Future Challenges

Benjamin Lee
Trinity College, Hartford, Connecticut

The Key Distribution Problem: Prior Advances and Future Challenges

Benjamin Lee

Introduction and Background

In order to ensure secure communications, known information must be shared among both communicating parties. This piece of information, called a cryptographic key, allows the sending party to successfully encrypt, and the receiving party to successfully decrypt the transmitted information. However, because both parties must have possession of this same key in order to successfully communicate, cryptographers throughout history have struggled with finding a method to transmit this key from the sending to the receiving party without it being intercepted along the way. For a time, until the advent of electronic cryptography, keys had to be distributed in person, via mail or by courier, as these were the only relatively secure methods for sending information from one person to another. Later in the computer age, however, computer scientists and cryptographers alike worked to devise schemes to successfully and securely transmit key material over insecure channels. As a result, there today exist solutions to the Key Distribution Problem for both symmetric, as well as public key cryptosystems.

On the symmetric side, the two most promising solutions are to either use a Key Derivation Function or a Key Wrapping scheme. Both of these methods provide a secure means for a symmetric cryptographic key to be transmitted from sender to receiver without the threat of outside intrusion. With public key systems, both Merkle's Puzzles and the Diffie-Hellman Key Exchange are viable, efficient solutions. While these methods differ by the type of cryptosystem in which they are employed, there also exists disparity in their respective methods of action. Both methods used in symmetric systems, Key Derivation Functions and Key Wrapping, are protocols which directly transmit the key material. In other words, it is the key itself (though successfully shielded) that is being sent over the insecure channel. Conversely, both protocols used in public key systems are key agreement protocols. With these schemes, trivial information is sent over the insecure channel, and this information is used by both parties to construct the same shared key governed by a predetermined set of rules. With all four of these protocols, cryptographic keys are able to be successfully and securely transmitted from one party to another. Trouble lies ahead, though, when considering the future of computing and its effect on the cryptographic landscape.

Quantum computing is the future. In fact, today, multiple companies such as Google and IBM already have working quantum computers which are able to complete tasks at speeds exponentially faster than their classical computer counterparts. When quantum computing becomes a reality, all cryptographic key distribution as well as encryption/decryption protocols will be rendered null and void. This is because quantum computers are able to withstand an exponentially larger computational load than common classical computers. Because of this, they would be easily able to break all current encryption/decryption and key distribution. Due to this conundrum, many "quantum safe" cryptographic algorithms have been developed, in which the computational effort required to break these ciphers is too great even for a quantum computer to realistically solve. The Key Distribution Problem carries over to the quantum space as well, and as such various "quantum safe" key distribution protocols have been developed as well, such as the well-known BB84 protocol. The work being done now to create standards of cryptography that can remain in a quantum future will ensure that people's data and privacy remain safeguarded for the years to come.

Key Distribution in Symmetric Cryptosystems

For the majority of the history of cryptography, symmetric ciphers were the gold standard for secure communication. Until the advent of more advanced computing technology beginning in the late 1930s, manual, symmetric ciphers were for the most part, the only type of encryption system that had been developed and was the only one which was widely used. Looking at these ciphers through a lens of the Key Distribution Problem, just as the ciphers themselves were manual, so were key distribution solutions. These included physically transporting the key, either over mail, by in person meeting or via courier, to the desired recipient of the future message. It was not until computerized encryption algorithms were created (such as DES) that solutions for distributing keys over networks, instead of over physical distances were needed. As such, multiple protocols for key distribution in symmetric cryptosystems were devised.

Key Derivation Functions

The first solution to the Key Distribution Problem for symmetric cryptosystems is the use of a Key Derivation Function. A Key Derivation Function is a mathematical function which takes a fixed input and produces an output which is used by both the sender and receiver as the encryption/decryption key (Sönmez, 2009). The information that is transmitted over the insecure channel is called the Key Derivation Key and is consists of a random string of bits. The sender will transmit this Key Derivation Key to the receiver. Both parties will then use the Key Derivation Key as the input to the agreed upon Key Derivation Function, and the result will be used as the encryption/decryption key.

Mathematical operations called pseudorandom functions are used to generate the key material (Sönmez, 2009). These functions are designed to behave as close to random as possible, since it is surprisingly difficult to successfully mathematically model random behavior. The random behavior of these functions is critical in a scheme like this, as it ensures that the creation of the key is one-way. That is, it would be extremely difficult for an adversary to construct the correct key following recovery of the Key Derivation Key. This is due to the relative random behavior of strong Key Derivation Functions. The more random the function behaves, the more difficult it is for an adversary to reconstruct the key from intercepted material.

When considering the security of a Key Derivation Function, it is also important to look at the length of the Key Derivation Key being used. This depends on the pseudorandom function being used in the specific key derivation. The two main pseudorandom functions used in most key derivations, Hash Message Authenticated Code (HMAC) and Cipher Message Authenticated Code (CMAC), require different key lengths to ensure a secure key derivation (Sönmez, 2009). When using an HMAC function, the Key Derivation Key is allowed to be any length (Sönmez, 2009). On the other hand, when using CMAC as the pseudorandom function, the length of the Key Derivation Key is determined by the length of the blocks being used in the CMAC, which is itself a form of block cipher (Sönmez, 2009).

The security of a Key Derivation Function also depends on the strength of the pseudorandom function being used. In this case, the strength of the pseudorandom function is defined as the amount of effort it would take an adversary to correctly map a random input string of bits to the correct output string, in other words, the key (Chen, 2008). Therefore, the greater the amount of work it would take an adversary to recover correct output material, the greater

strength the pseudorandom function has. The strength, and by proxy the required effort, are most likely dependent on how random the function being used behaves. The more a function trends towards truly random behavior, the more work it would take someone to reverse the process, as it is much more difficult to detect and exploit patters when the behavior of the data becomes more and more random.

A final defining feature of Key Derivation Functions is their ability to be used to create a key hierarchy. This is where key material derived through the use of a Key Derivation function is fed back into the same function, resulting in more keys (Sönmez, 2009). This process can be repeated as many times as desired. This feature of Key Derivation Functions is important when considering the efficiency of a symmetric cryptosystem. Given that each communication needs to be encrypted/decrypted using a unique shared key, being able to create many keys from one original would be much more efficient than using redoing the original process over multiple times. When considering a system in which many unique keys are used, such as modules encrypted using DES, having many keys available at once as opposed to creating a new key each time one is needed is certainly more efficient.

Key Wrapping

Key Wrapping is another successful, secure scheme for distributing symmetric cryptographic keys. Key Wrapping is the practice of encrypting key material itself, and then transmitting the encrypted key over the insecure channel to the desired recipient (Gennaro, 2009). Key Wrapping differs from using a Key Derivation Function in the sense that it is the key itself, though encrypted, that is being sent over the insecure channel, whereas with a Key Derivation Function it is trivial information that is sent to later construct the key.

A secure key wrapping scheme is executed following a method called Hash-then-Encrypt (Gennaro, 2009). This is when key material is first fed into a hash function, and the output is symmetrically encrypted (Gennaro, 2009).[i] Encryption of the key itself is commonly done using the AES protocol. In a scenario like this, the original key will be hashed into blocks of 64 bits, and each block of 64 bits will be encrypted according to the AES protocol (Sönmez, 2009).

Given that key wrapping is typically a process involving AES, it is important to also consider how the properties of AES itself will affect the security and efficacy of the key wrap. The first property to consider is the size of the key itself that is getting wrapped. Generally, it is believed that the longer the key is, the more secure the key wrap will be overall (Sönmez, 2009). The reasoning behind this is simple; the longer the message being encrypted is (the message being encrypted in this case *is* the key) the more difficult it is for an adversary to decrypt it. This, once again, becomes a problem of effort. Similar to Key Derivation Functions, where the inherent security was the amount of effort it would take to solve the problem, much of the security of a key wrap lies on the same principle. A key wrap is simply a symmetric cipher whose contents happen to hold critical significance. As such, its security is no different than the security of any other symmetric cipher and relies heavily on the length and complexity of the message being encrypted.

Key Distribution in Public Key Cryptosystems

Public Key Cryptography holds special significance in the context of the Key Distribution Problem. Public Key Cryptography is, itself, a solution to the Key Distribution

Problem found in symmetric cryptosystems. Public Key Cryptography is based off the principle of there being two keys used in the scheme: the public key, used by the sender to encrypt the message and the private key, used by the recipient to decrypt the message. However, issues with key distribution still do exist. For example, for someone wishing to send someone else an encrypted message through a public key system, the sender needs to have the public key of the recipient in order to encrypt the message. While it will not corrupt the integrity of the system to have the public key sent over the insecure channel, it may be of good conscious for the sender to securely distribute his public key anyway. For this there are solutions.

Merkle's Puzzles

Ralph Merkle is a computer scientist who is credited as one of the inventors of public key cryptography. He, along with fellow computer scientists Whitfield Diffie and Martin Hellman invented the field in 1976, after months of pondering over how to create a cryptosystem in which key exchange became unnecessary (Singh, 1999). In doing so, they realized that a successful public key system is reliant on a strong method of secure key exchange. In preliminary work on the matter, Merkle devised a scheme in which two parties could exchange a key by means of a set of cryptographic puzzles, which were "cryptogram[s] which [are] meant to be broken" (Merkle, 1978, p. 296). These later became known as Merkle's Puzzles and formed the basis for public key cryptography.

To execute a key exchange through the use of Merkle's Puzzles, the sender would first encrypt a set of short plaintext hashes via any type of encryption scheme he desires. Most times, this is a simple symmetric cipher. The number of puzzles encrypted is agreed upon by the sender and recipient (Merkle, 1978). Each puzzle is made up of two pieces. The first is a number unique to each puzzle that is called the ID number and is used to identify each puzzle in the set. The second is the puzzle key, and is the key associated with the chosen puzzle (Merkle, 1978). Having received the entire set of puzzles, the recipient will choose one at random and decrypt it. He will then receive the ID number from the decrypted puzzle and send it to the sender over the insecure channel. With the sender having the ID number, he can recover the puzzle from which it is from. The puzzle key from the chosen puzzle is used as the encryption key for further secure communication.

Security in Merkle's Puzzles appears in two places. The first is with the size of the key used to encrypt the puzzles initially. As with many cryptographic schemes, security is dependent on the effort it would take someone to break the scheme. This is the same for Merkle's Puzzles. The puzzles can be encrypted with a key of any length, and as such a longer key would require a greater amount of effort to solve the cipher by brute force (Merkle, 1978). Therefore, security of the puzzles is firstly defined by the length of the key used to encrypt them. The second aspect of security appears in the number of puzzles chosen to be used in the scheme. Once again, this is a problem of effort. The more puzzles used in the set, the longer it would take an adversary to decrypt enough of them until reaching the correct puzzle and key. Therefore, the greater the number of puzzles used in the set, the more secure the scheme becomes as a whole.

Merkle's Puzzles are an early, and relatively simplified example of one of the key facets of public key cryptography: one-way functions. Computer scientist Subhash Kak describes Merkle's Puzzles as being "equivalent to the dictionary method of one-way function generation" (Kak, 1984, p. 106). Put simply, a one-way function is a function which can be computed easily one way but is difficult to solve in the other direction. These form the basis of public key

cryptography. Kak is referencing a scenario in which party A creates two dictionaries of languages X and Y (Kak, 1984). Party A then transmits dictionary X to party B, who selects an entry at random, comprising the chosen word in both language X and language Y (Kak, 1984). Party B then sends the element in language Y from the chosen entry back to Party A, who finds that word in the dictionary for language Y, using the corresponding word in language X as the encryption/decryption key (Kak, 1984).

This scheme invented by Merkle opened the door for research and development into public key cryptography, some even close to home. Two of Merkle's colleagues, Whitfield Diffie and Martin Hellman expanded upon the principles of Merkle's Puzzles and created a second, more comprehensive form of key agreement.

Diffie-Hellman Key Exchange

Whitfield Diffie and Martin Hellman are computer scientists and cryptographers who worked with Ralph Merkle during the infancy of public key cryptography. In 1976, following months of work on the matter, Diffie and Hellman created a key distribution protocol that expanded on the principles laid out in Merkle's Puzzles, specifically the use of one-way functions (Holden, 2017). The Diffie-Hellman Key Exchange is grounded in what has aptly been named the Diffie-Hellman Problem, a concept similar in nature to the discrete logarithm problem. Its security derives from this key question: Given two values a$Xi \bmod q$ and a$Xj \bmod q$, compute a$XiXj \bmod q$. This problem, while seemingly simple at first glance, is actually extremely difficult and nearly impossible for a human to realistically solve, especially when values of p are extremely large.

The protocol is conducted as follows. First, both parties participating in the scheme (i & j) must decide on a large prime number q (Diffie & Hellman, 1976). Today, it is recommended for this prime to be at least 600 digits long in order to ensure proper security (Holden, 2017). This is because, as in the problem described above, the larger the q value is, the more difficult the problem will be due to the added effort needed to deal with a number that large. Second, both parties need to find a second number a, which is a primitive root of q (Diffie & Hellman, 1976). This number becomes important during the final computation. It is important to note that both these numbers can be sent over an insecure channel with no consequence, as they themselves hold no relation or resemblance to the key. Next, each party chooses a random number $Xi$and Xj respectively, where $Xi \in (1, q-1)$ and $Xj \in (1, q-1)$ (Diffie & Hellman, 1976). It is critical to the integrity of the scheme that both Xi and Xj remain secret. Using these values, each party then constructs a new number Y according to the following rule: $Yi = aXi \bmod q$ and $Yj = aXj \bmod q$ (Diffie & Hellman, 1976). It is these two values which will be transmitted over the insecure channel, with Yi sent to party j and Yj sent to party i. Now, each party takes the value Y received from the other and raises that value to the power of their respective X value. Doing so yields the following result for party i: $(Yi)Xj = (aXi)Xj = Kij$. And for party j: $(Yj)Xi = (aXj)Xi = Kij$ (Diffie & Hellman, 1976). If done correctly, they produce the same result which is used as the key.

In creating their protocol, Diffie and Hellman exploited some of the key principles used in Merkle's Puzzles. Firstly, both protocols only send information trivial in relation to the key itself over the insecure channel. In the case of this scheme, that information is the original prime q, the primitive root a, and the composed values Y. None of these four values contain the encryption/decryption key and as such all of them are safe to be transmitted over the insecure channel. This is what makes Diffie-Hellman (and Merkle's Puzzles for that matter) unique when

considering the scope of key distribution. Instead of sending the actual key in some secure fashion, what is being sent are components, or building blocks of the key. What makes these components themselves secure is the Diffie-Hellman Problem to which they adhere. This, on top of the fact that the information being transmitted simply is *not* the key is what makes Diffie-Hellman secure, and is what started the migration to public key cryptography.

Quantum Key Distribution

Today in 2020, recent developments in computing technology have cemented the fact that quantum computing is a reality and is going to become commonplace sooner rather than later. The caveat with this fact, though, is that once quantum computing becomes normal, all cryptographic algorithms created to date will be rendered obsolete due to the sheer computational power of these new machines. Therefore, various new "quantum proof" cryptographic and key distribution protocols have been developed to date. These protocols remain unsolvable by even a quantum computer.

Quantum Basics

Quantum mechanics is defined by the Encyclopedia Britannica as "the science dealing with the behavior of matter and light on the atomic and subatomic scale" (Squires, 2020). Most commonly, this study is focused on the investigation of the behavior of photons--elementary light particles-- and electrons. Much of the quantum mechanics employed in Quantum Key Distribution protocols revolves around the quantum state of a particle.

Quantum states are an integral aspect to quantum mechanics as a whole and exist due to one of the foundational laws of quantum mechanics—superposition. Superposition states that a quantum particle can exist as a linear combination of states before being measured (Holton, 2020). At measurement, the particle will collapse into a single one of these states bounded by the probability of measuring each state given its initial superposition (Holton, 2020). Therefore, the quantum state of a particle is the set of the probabilities of measuring each possible outcome. The quantum state of a particle can also be represented by a wave function. This is a function which represents the full quantum state of a particle at any given point in time (Tamvakis, 2019). It's the combination of superposition and quantum states that is used in Quantum Key Distribution protocols.

Quantum Key Distribution harnesses the superposition property of quantum particles to transmit key information over an insecure channel. It's important to note that these protocols have been distinguished into three categories: discrete-variable, continuous-variable, and distributed-phase-reference (Scarani, 2009). What makes these categories each unique is how the quantum particles are measured. In discrete-variable and continuous-variable protocols, measurement of the particles (which in this case are photons) occurs following the transmission of the particles. Distributed-phase-reference protocols employ a technique called homodyne detection, which measures the particles via the frequency of their wave function (Scarani, 2009). Though the different Quantum Key Distribution protocols do vary in their methods of action, they attempt to complete the same task. As an example of how these protocols work to ensure security, I will describe the first Quantum Key Distribution protocol, BB84, invented by Charles Bennett and Gilles Brassard.

The BB84 Protocol

The BB84 Protocol relies on the use of photons and their quantum properties to encode and send key information across an insecure channel. The scheme begins when the sender party chooses a random string of bits which will be used as the key. The sender then encodes each bit onto a photon. This is done via the photon's polarization, or the direction it spins. For each bit, the sender will choose one of two methods for encoding the bit at random. The first is called the rectilinear basis, where a bit of zero is encoded as $|0\rangle$,meaning a horizontal polarization, and a bit of one is encoded as $|1\rangle$,a vertical polarization (Chong, 2009).[ii] The second is the diagonal basis, where zero is encoded as $|+\rangle$, a diagonal spin in one direction, and one is encoded as $|-\rangle$, a diagonal spin in the opposite direction, orthogonal to the other photon (Chong, 2009).

Just as each photon had to be encoded according to either the rectilinear or diagonal basis, the same principle exists at measurement. Measurement is done using a photon filter, either vertical/horizontal or diagonal, as determined by the receiver. If the encoding basis and the decoding basis for a given bit are the same, the measurement will be successful. This is because it is possible to distinguish between two orthogonal states, which in this case are the photon itself and the filter it's being measured with (Van Assche, 2006). However, if the encoding and decoding bases differ, the measurement will fail, and the photon will collapse into either a zero or a one (in whatever base the photon was encoded in) (Van Assche, 2006).

Following the transmission and measurement of all the encoded photons, the sender will tell the recipient the base he used to encode each bit and corresponding photon. Bits that were encoded and decoded using different bases will be discarded, and only bits encoded and decoded with identical bases will be kept (Van Assche, 2006). This process, known as sifting, is used to determine which bits were correctly measured without revealing the actual bits themselves. The remaining bits available following the sifting process are then used as the encryption/decryption key for further communication.

A defining feature of the BB84 Protocol is the ability of the communicating parties to detect intrusion into the scheme by a third party. Were there to be an intruder attempting to intercept transmitted key material, they would have to measure the photons just as the receiver did. As such, the intruder will also have to choose a base to measure the photon on at random. Given this forced guesswork, it is safe to assume that the intruder will guess the correct base around 50% of the time, not jeopardizing the measurement (Haitjema, 2007). However, the other 50% of the time he will use the incorrect base. As such, the photon will collapse into either zero or one at random, and the correct encoded information is lost. This is guaranteed by Heisenberg's Uncertainty Principle (Haitjema, 2007).[iii] Following the transmission, the sender and receiver will compare actual measured bit values of bits measured with the same base. If multiple errors are found, the presence of an intruder is very likely.

Looking to the Future

The Key Distribution Problem is an issue that has been worked on continuously for many years. Unlike many issues faced in science, it's one which requires different solutions depending on the context of the problem as a whole. It was evhen directly responsible for the creation of an entirely new field of study in cryptography--public key cryptography--which opened the doors for safe and secure communication over long distances.

It's also important to acknowledge that the problem is not yet over. While we have

created contingency plans for solving this problem in the future, i.e., the Quantum Key Distribution protocols, people today do not yet know how prevalent quantum computing technology will become. As such, the technology could easily grow, change, or evolve to a point that even our current "quantum-safe" protocols will be of no use. But today, we have the problem solved.

This problem is also significant to the daily lives of any electronically connected human. Nowadays, seemingly everyone has some sort of personal data kept online, whether it be bank account information, medical history, or a myriad of other important personal data. Regardless, it is important that these critical forms of information remain secure while also being able to harness the full capacity of our computer networks and be sent from site to site as necessary. Therefore, much of the integrity of our personal data online relies on successful key distribution protocols, which ensure that our information is secure and uncompromisable when it is in transit.

Truly, it is not yet entirely known what role key distribution will play in the future of computing. But, using the past as any example, it is most likely a critical role. As such, it is important that this problem continue to be researched, worked on, and new solutions created, so that when the next leap in computing technology comes to light, data and privacy will not have to be compromised.

Bibliography

Chen, L. (2008). Recommendation for key derivation using pseudorandom functions. National Institute of Standards and Technology.

Chong, S.-K., & Hwang, T. (2010). Quantum key agreement protocol based on BB84. Optics Communications, 283(6), 1192–1195. https://doi.org/10.1016/j.optcom.2009.11.007.

Diffie, W., & Hellman, M. (1976). New directions in cryptography. *IEEE Transactions on Information Theory*, *22*(6), 644–654. https://doi.org/10.1109/tit.1976.1055638.

Gennaro, R., & Halevi, S. (2009). More on Key Wrapping. In M. J. Jacobson, V. Rijmen, & R. Safavi-Naini (Eds.), *Selected Areas in Cryptography* (pp. 53–70). Springer Berlin Heidelberg.

Haitjema, M. (n.d.). *A Survey of the Prominent Quantum Key Distribution Protocols* [Washington University in St. Louis]. Retrieved December 10, 2020, from https://www.cse.wustl.edu/~jain/cse571-07/ftp/quantum.pdf.

Hidary, J. D. (2019). Superposition, Entanglement and Reversibility. In J. D. Hidary (Ed.), *Quantum Computing: An Applied Approach* (pp. 3–9). Springer International Publishing. https://doi.org/10.1007/978-3-030-23922-0_1.

Holden, J. (2017). Public-Key Ciphers. In *The Mathematics of Secrets* (NED-New edition, pp. 201–240). Princeton University Press; JSTOR. https://doi.org/10.2307/j.ctvc775xv.11.

Holton, Wi. (2020). Quantum computer. In *Encyclopedia Britannica*. Encyclopedia Britannica. https://www.britannica.com/technology/quantum-computer.

Kak, S. C. (1984). On the method of puzzles for key distribution. *International Journal of Computer & Information Sciences*, *13*(2), 103–109. https://doi.org/10.1007/BF00978711.

Merkle, R. (1978). Secure communications over insecure channels. *Communications of the ACM*, *21*(4), 294–299. https://doi.org/10.1145/359460.359473.

Scarani, V., Bechmann-Pasquinucci, H., Cerf, N. J., Dušek, M., Lütkenhaus, N., & Peev, M. (2009). The security of practical quantum key distribution. *Reviews of Modern Physics*, *81*(3), 1301–1350. https://doi.org/10.1103/RevModPhys.81.1301.

Singh, S. (1999). *The Code Book: The Evolution of Secrecy from Mary, Queen of Scots, to Quantum Cryptography* (1st ed.). Doubleday.

Sönmez, O., & Paar, I. C. (2009). Symmetric Key Management: Key Derivation and Key Wrap. *Bochum, Germany, February 2009*.

Squires, G. (2020). Quantum mechanics. In *Encyclopedia Britannica*. Encyclopedia Britannica. https://www.britannica.com/science/quantum-mechanics-physics.

Tamvakis, K. (2019). Basic Principles of Quantum Mechanics. In K. Tamvakis (Ed.), *Basic Quantum Mechanics* (pp. 85–106). Springer International Publishing. https://doi.org/10.1007/978-3-030-22777-7_5.

Van Assche, G. (2006). *Quantum cryptography and secret-key distillation*. Cambridge: Cambridge University Press.

---

[i] A cryptographic hash function is a function which takes a data set of arbitrary size and maps the data to another set of another fixed size.

[ii] The notation used is called Dirac or bra-ket notation and is used in quantum mechanics to represent a quantum state.

[iii] Heisenberg's Uncertainty Principle states that "in a quantum system only one property of a pair of conjugate properties can be known with certainty" (Haitjema, 2007, p. 2).