

Information theoretically secure, enhanced Johnson noise based key distribution over the smart grid with switched filters

Elias Gonzalez, Laszlo B. Kish*, Robert Balog, Prasad Enjeti

Department of Electrical and Computer Engineering, Texas A&M University, College Station, TX 77843-3128, USA

*E-mail: Laszlo.Kish@ece.tamu.edu

Abstract

We introduce a protocol with a reconfigurable filter system to create non-overlapping single loops in the smart power grid for the realization of the Kirchhoff-Law-Johnson-(like)-Noise secure key distribution system. The protocol is valid for one-dimensional grids (chain-like power line). The speed of the protocol (the number of steps needed) versus grid size is analyzed. When fully developed such system has the potential to achieve unconditionally secure key distribution over the smart power grid of arbitrary dimensions.

Introduction

1.1 KLJN, the information theoretically secure wire-based key exchange scheme

A smart grid [1,2] is an electrical power distribution network that uses information and communications technology to improve the security [3,4], reliability, efficiency, and sustainability of the production and distribution of electricity.

Private key based secure communications require a shared secret key between Alice and Bob who may communicate over remote distances. In today's secure communications, sharing such a key also utilizes electronic communications because courier and mail services are slow. However the software based key distribution methods offer only limited security levels that are only *computationally-conditional* thus they are *not future-proof*. By having a sufficiently enhanced computing power, Eve can crack the key and all the communications that are using that key. Therefore, unconditional security (indicating that the security holds even for infinite computational power), which is the popular wording of the term "information theoretic security" [5], requires more than a software solution. It needs the utilization of the proper laws of physics.

The oldest scheme that claims information theoretic security based on the laws of physics is quantum key distribution (QKD). It is an interesting episode that currently the security of available QKD schemes has been compromised [6-20], though this situation is probably temporary because they have the potential to reach a satisfactory security level in the future. However, QKD devices are prohibitively expensive and have other practical weaknesses, such as

sensitive to vibrations, they are bulky, limited in range, and require a special “dark optical fiber” cable and sophisticated infrastructure.

On the other hand, the smart grid offers a unique way of secure key exchange because each household (host) in the grid is electrically connected. To utilize a wire connection for secure key exchange, different set of the laws of physics (not the laws applied for QKD that works with optical fibers) must be utilized. Recently a classical statistical physical alternative to QKD, the Kirchhoff-Law-Johnson-(like)-Noise (KLJN) key exchange system has been proposed [21,22], which is a wire-based scheme that is free from several weaknesses of QKD. Similarly to QKD, KLJN is also an information theoretically secure key distribution [23]; however it is robust; not sensitive to vibrations; it has unlimited range [24]; it can be integrated on chips [25]; it can use existing wire infrastructure such as power lines [26]; and KLJN based networks can also be constructed [27].

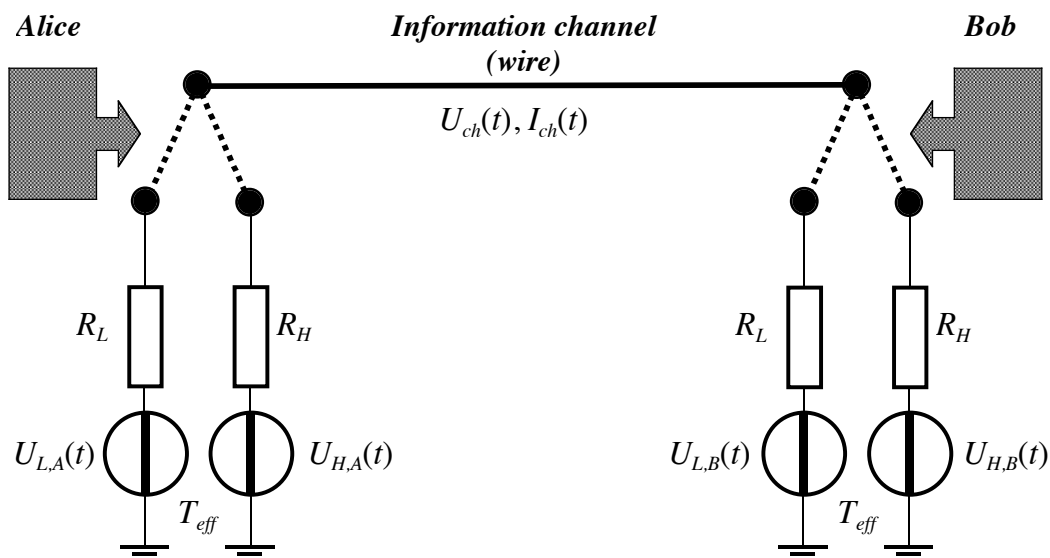


Figure 1. The core of the KLJN key scheme [21]. This simple system is secure only against passive attacks in the idealized case (mathematical limit). Security enhancements (including filters) to provide protection against invasive attacks [21,22] and other types of vulnerabilities are not shown. In practical applications electronic noise generators emulate an enhanced Johnson noise with a publicly agreed high effective temperature.

The KLJN channel is a wire [21]. At the beginning of each clock cycle, Alice and Bob, who have identical pairs of resistors R_L and R_H (representing the 0 and 1 bit situations) randomly select and connect one of the resistors, see Figure 1. In practical applications, voltage noise generators enhance the Johnson noise of the resistors so that all resistors in the system have the same, publicly known effective noise-temperature T_{eff} (where $T_{eff} \geq 10^9$ Kelvin). The enhanced Johnson noise voltages $\{U_{L,A}(t)$ or $U_{H,A}(t)$; and $U_{L,B}(t)$ or $U_{H,B}(t)\}$ of the resistor result in a channel noise voltage $U_{ch}(t)$ between the wire and the ground, and a channel noise current $I_{ch}(t)$ in the wire. Low-pass filters are used because the noise-bandwidth, which we also call

KLJN-band B_{kljn} (its value depends on the range), must be chosen so narrow that wave, reflection, and propagation/delay effects are negligible, otherwise the security is compromised [21]. Alice and Bob can measure the mean-square amplitudes $\langle U_{ch}^2(t) \rangle$ and/or $\langle I_{ch}^2(t) \rangle$ within the KLJN-band in the line. From any of these values, the loop resistance can be calculated [21] by using the Johnson noise formula with the noise-bandwidth T_{eff} :

$$\langle U_{ch}^2(t) \rangle = 4kT_{eff} R_{loop} B_{kljn} \quad \langle I_{ch}^2(t) \rangle = \frac{4kT_{eff} B_{kljn}}{R_{loop}} \quad (1)$$

Alice and Bob know their own choice resistor thus, from the loop resistance, they can deduce the resistance value and the actual bit status at the other end of the wire. In the ideal situation, the cases $R_L | R_H$ and $R_H | R_L$ represent a secure bit exchange event because they cannot be distinguished by the measured mean-square values. The eavesdropper (Eve) can do the very same measurements but she has no knowledge about any of the resistance choices thus she is unable to extract the key bits from the measured loop resistance.

1.2 Utilizing the smart power grid for information theoretic secure key exchange

The disadvantage of the KLJN key exchange protocol that it requires a wire connection. Investors are hesitant to cover the cost of new infrastructure for solely the purpose of security. On the other hand, virtually each building in the civilized world is connected to the electrical power grid. This fact is very motivating to explore the possibility of using the power grid as the infrastructure for the KLJN protocol. However, only the single loop shown in Figure 1 is secure. When Alice and Bob are two remote hosts in the smart grid, they should indeed experience a single loop connection like in Figure 1. Thus, for smart grid applications, proper filters must be installed and controlled for the KLJN frequency band where the key exchange operates. Though simple examples have been outlined to prove that a KLJN key exchange between two remote points in a chain-like power grids with filters [26] can be achieved, neither details about the structure of the filter units nor network protocols to connect every host on the grid with all other hosts have been deduced.

The present paper aims to make the first steps in this direction by presenting a working scheme with scaling analysis of the speed of key exchange versus network size. We limit our network to a one-dimensional linear chain network to utilize the smart power grid for KLJN secure key exchange. We show and analyze a protocol to efficiently supply every host with proper secure keys to separately communicate with all the other hosts.

Discussions and Results

Because the pattern of connections between KLJN units must be varied to provide the exchange of a separate secure key for each possible pair of host, the network of filters and their connections must be varied accordingly. The power line filter technology is already available [28,29] and we will show that the required results can be achieved by switching on/off proper filtering units, in a structured way in the smart grid. We will need filters to pass or reject the KLJN frequency band B_{kljn} and/or the power frequency f_p (50 or 60 Hz). When both B_{kljn} and f_p are passed, it is a short; and when both of them are rejected, it is a break. We will call these filters "switched filters".

2.1. Switched Filters

We call the functional units connected to the smart power grid *hosts*. A host is able to execute KLJN key exchange toward left and right in a simultaneous way. That means each host has two independent KLJN units. The filter system must satisfy the following requirements:

- i) Hosts that currently do not execute KLJN key exchange should not interfere with those processes even if the KLJN signals pass through their connections.
- ii) Moreover, each host should be able to extract electrical power from the grid without disturbing the KLJN key exchanges.

We define the size of a network as being of size N when that network has $N + 1$ hosts, see Figures 2 and 3.

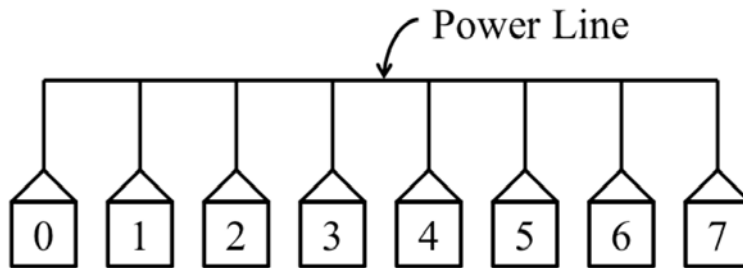


Figure 2. Example of one-dimensional grid: a chain network. This example has a network of size $N=7$.

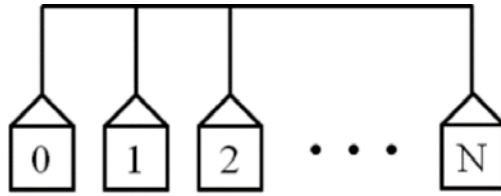


Figure 3. The chain network with $N+1$ hosts and size N .

Intermediate hosts in the line can be in two different states according to the need:

- α) Drawing power and executing a KLJN key exchange both left and right.
- β) Drawing power and not executing KLJN key exchange.

Hosts at the two ends can be in similar situations except that they can communicate in only a single directions, thus they are special, limited cases of the intermediate hosts in which we are focusing our considerations when discussing filters.

Filter boxes at each host will distribute the KLJN signals and the power and they are responsible for connecting the proper parties for the KLJN key exchange and to supply the hosts with power, see Figure 4. The filters boxes can be controlled either by a central server and/or an automatic algorithm. In the following section we discuss the protocol of this control. Each filter box has three switched filters and a corresponding output wire, see Figure 4:

- a) The Left KLJN Filter for the KLJN key exchange toward left,
- b) The Right KLJN Filter for the KLJN key exchange toward right.
- c) The Power Filter to supply power to host, and to separate the left and right KLJN filters, when needed.

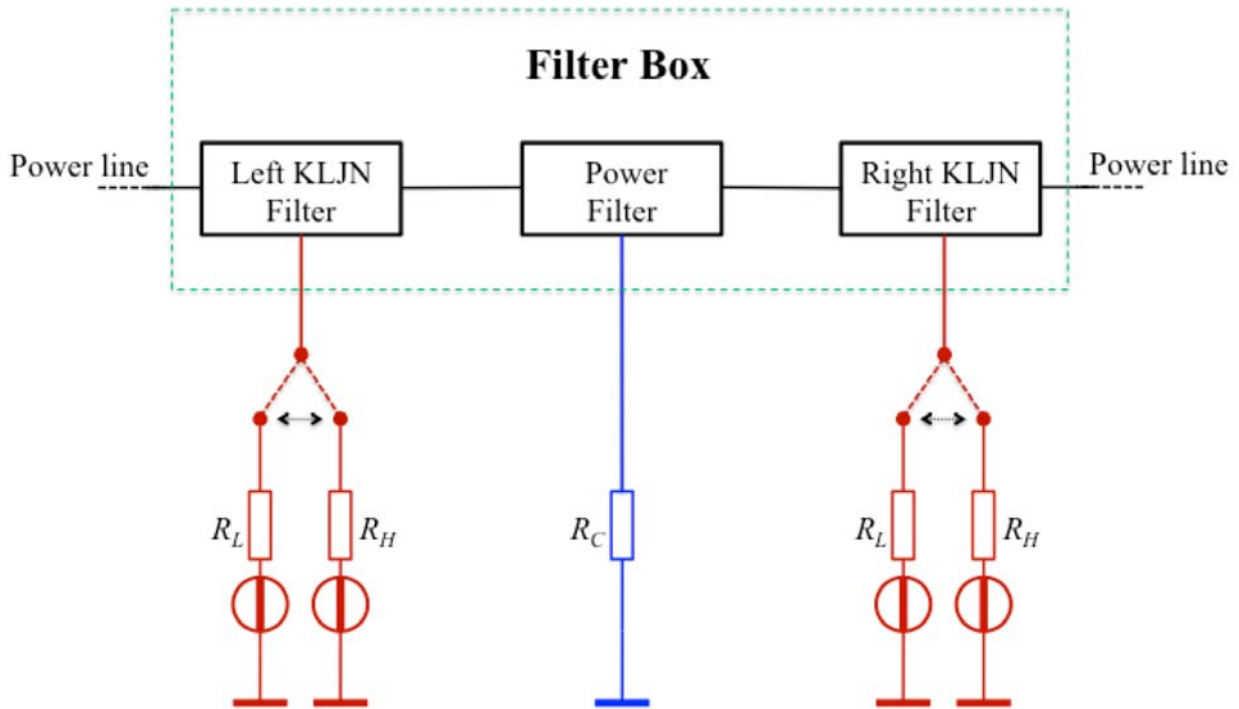


Figure 4. Building blocks in a filter box.

The properly controlled filter boxes will provide non-overlapping KLJN loops between the hosts, see below. The reason for having two KLJN units per host is to decrease the time needed to connect every host by having simultaneous loops toward left and right, without overlapping. Figure 5 shows an example for $N=7$. The solid black line means that both KLJN bandwidth and power frequency is passing through (ordinary wire: the original line). The short red dashes represent B_{kljn} and that f_p is rejected. The longer blue dashes indicate the opposite situation: only the power frequency is passing and the KLJN bandwidth is rejected.

For example, Host 3 can execute a key exchange toward left with Host 1 and simultaneously (and independently) with Host 6; and at the same time simultaneously provide independent key exchanges between Hosts 0 and 1 and 6 and 7.

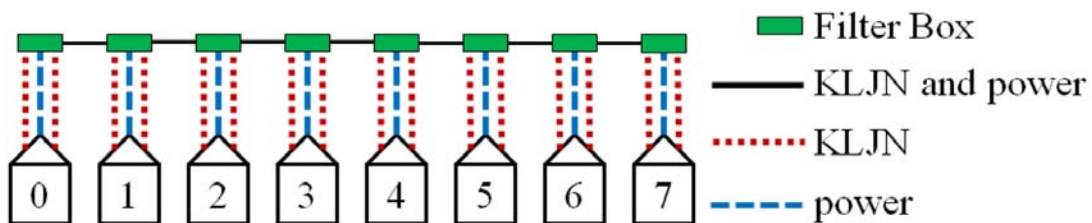


Figure 5. Example for network of size $N=7$. Each host is connected to a filter box and the filters boxes are connected to the power grid. Note how each host has three wire connections to its filter box.

Then the filter boxes of the inactive hosts 2, 4 and 5 must separate their hosts from the KLJN band, and the same time let them supply by power. We call this working mode of the filter boxes of non-active hosts *State 1*. The wiring and frequency transfer of the Filter Box in State 1 are shown in Figure 6 and Tables 1,2.

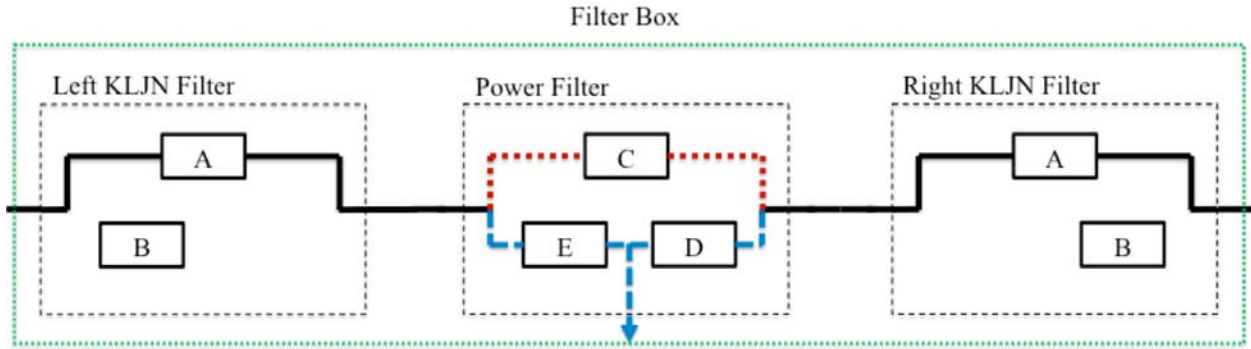


Figure 6. The filter box of the inactive host (when the it is not executing KLJN key exchange): State 1. Everything is passing between left and right and the host can access only the power. Filter A is passing everything (shorted). Filter B is disconnected. Filter C is passing B_{kljn} only. Filters and E and D are passing f_p only.

KLJN Filters	Filter A	Filter B
KLJN B_{kljn} Allowed	Yes	No
Power Frequency Allowed	Yes	No

Table 1. Truth table of the KLJN Filters in State 1 (inactive host).

Power Filter	Filter C	Filter D	Filter E
KLJN B_{kljn} Allowed	Yes	No	No
Power Frequency Allowed	No	Yes	Yes

Table 2. Truth table of the Power Filter in State 1 (inactive host).

At the same time, the filter boxes of the active Hosts 1 and 6 must allow the KLJN band to their hosts; the right KLJN filter to Host 1 and the left KLJN filter to Host 6 to realize a single KLJN loop between the resistors of Host 1 and 6. Similarly, they must allow the B_{kljn} band between the right of Host 0 and left of Host 1 and between the right of Host 6 and the left of Host 7. At the same time, the power filters of the active hosts (0,1,3,6,7) must separate the KLJN loops by rejecting B_{kljn} . We call this working mode of the filter boxes of hosts executing key exchange *State 2*. The wiring and frequency transfer of the Filter Box in *State 2* are shown in Figure 7 and Tables 3,4.

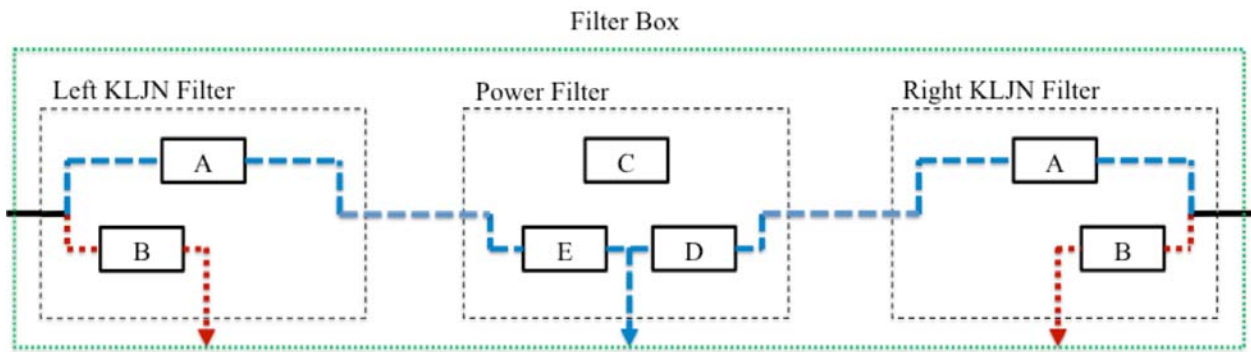


Figure 7. The filter box of the active host (when it is executing a KLJN key exchange): *State 2*. The power is passing between left and right but the KLJN band is not and the left and right KLJN units are separated while doing a key exchange toward left and right.

KLJN Filter	Filter A	Filter B
B_{kljn} allowed	No	Yes
f_p allowed	Yes	No

Table 3. Truth table of left KLJN filter when a host is in stage two.

Power Filter	Filter C	Filter D	Filter E
B_{kljn} allowed	No	No	No
f_p allowed	No	Yes	Yes

Table 4. Truth table of power filter when a host is in stage two.

In this section we have shown that the line can be packed with non-overlapping KLJN loops to execute simultaneous key exchanges between selected hosts. In the next section, we propose a network protocol to provide secure keys for each host to be able to communicate securely via the internet or other publicly accessible channels between arbitrary pairs of hosts. The time requirement of key exchange over the whole smart grid will also be analyzed versus the network size N .

2.2. Protocol and Speed

To quickly and efficiently connect every host with all other host in the same one-dimensional network we need to establish a protocol. The protocol must make every possible connection in the network, must not overlap loops, and must be quick and efficient by making as many simultaneous loops without overlapping as possible.

To determine the time and speed requirements to establish a KLJN secure key exchange we must first define terms. In the classical KLJN system, where only the noise existed in the wire, the low-frequency cutoff of the noise was 0 Hz and the high-frequency cut-off was B_{kljn} . In the case of KLJN in a smart grid, this situation will be different because of the power frequency. However, at short distances (less than 10 miles), the B_{kljn} band can be beyond the power frequency f_p and the difference is negligible. Then the shortest characteristic time in the system is the correlation time τ_{kljn} of the noise ($\tau_{kljn} \approx 1 / B_{kljn}$). B_{kljn} is determined by the distance L between Alice and Bob so that $B_{kljn} \ll c / L$ [21] (for example, $B_{kljn} \ll 100$ kHz for $L=1$ kilometer). Alice and Bob must make a statistics on the noise, which typically requires around $100 \tau_{kljn}$ duration [24] (or 0.01 seconds if we use $B_{kljn} = 10$ kHz) to have a sufficiently high fidelity (note, faster performance is expected in advanced KLJN methods [30]). A bit exchange (BE) occurs when Alice and Bob have different resistor values, this occurs on average of 200

τ_{kljn} or 0.02 seconds if $B_{kljn} = 10$ kHz. The length of the secure key exchange can be any arbitrary length. For example if we have a key length of 100 bits then, we need 100 BE which requires on average $20000 \tau_{kljn}$ which is approximately 2 seconds if B_{kljn} is 10 kHz. Once the KLJN secure key has been exchanged the total amount of time needed to complete this is one KLJN secure key Exchange (KE). If B_{kljn} is 10 kHz and if the length of the key is 100 bits then one KE is approximately 2 seconds. While the key exchange is slow, the system has the advantage that it is running continuously (not only during the handshake period like during common secure internet protocols) thus large number of secure key bits are produced during the continuous operation.

The protocol we propose here first connects the nearest neighbor of every host; this allows the highest number of simultaneous non-overlapping loops per KE and only requires one KE to complete this first step. The protocol then connects the second nearest neighbors; this allows the second highest numbers of simultaneous loops per KE. However, due to the requirement of avoiding overlapping loops, connecting each pairs of second nearest neighbors requires two KEs. The protocol then connects the third nearest neighbors which requires 3 KEs to complete and connects the third most simultaneous loops per KE. This procedure continues until the i -th nearest neighbor is equal to or less than half of the size of the network. If the number of steps i between the i -th nearest neighbors satisfies the relation $i > N / 2$ then, to avoid overlapping loops, only one connection per KE is possible.

As an example, we will show in the next section that for $N=7$ (see Figure 2) 16 KEs (or approximately 32 seconds if B_{kljn} is 10 kHz) are required when the keys are 100 bits long. Using this protocol, the analytic form of the exact time required to fully arm every host with enough keys to securely communicate with everybody in the network is dependent on the size of the network and whether the network has an even or odd size. In the following sections we will deduce the analytic relations and show examples.

2.2.1 The network size N is an odd number

We illustrate the calculation of time requirement with examples shown in the following figures. A general formula for an arbitrary size network when N is odd is given after this example. In this example we have a network of size $N = 7$. We have 8 host with index $i \{i \in \mathbb{N} : 0 \leq i \leq 7\}$. We have 7 intermediate connections between the first and last host.

The first step in the protocol connects the nearest neighbors, see Figure 8.

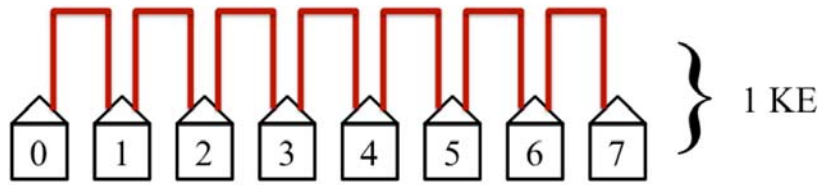


Figure 8. The first step in the protocol connects the nearest neighbors. This step is the quickest and most efficient. It has the most non-overlapping simultaneous loops and requires only 1 KE to complete.

The second step in the protocol will then connect the second nearest neighbors, see Figure 9.

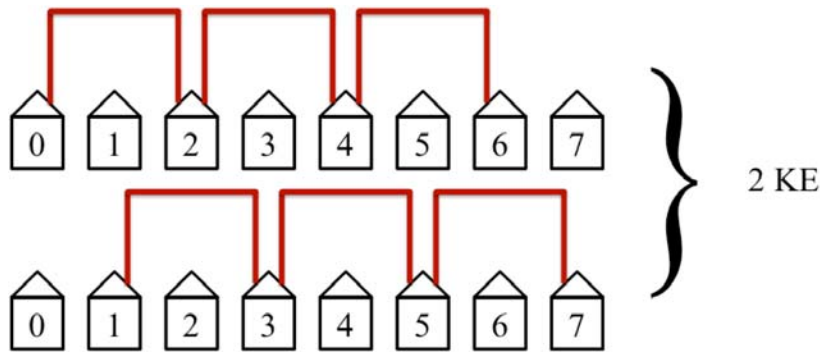


Figure 9. The second step in the protocol connects the second nearest neighbors. This step is the second quickest and the second most efficient. It has the second most non-overlapping simultaneous loops and requires 2 KEs to complete.

The protocol will then connect the third closest neighbors as shown in Figure 10. This will take 3 KEs to complete and is not as efficient as the first two steps in the protocol but still has simultaneous loops in two of its KE steps.

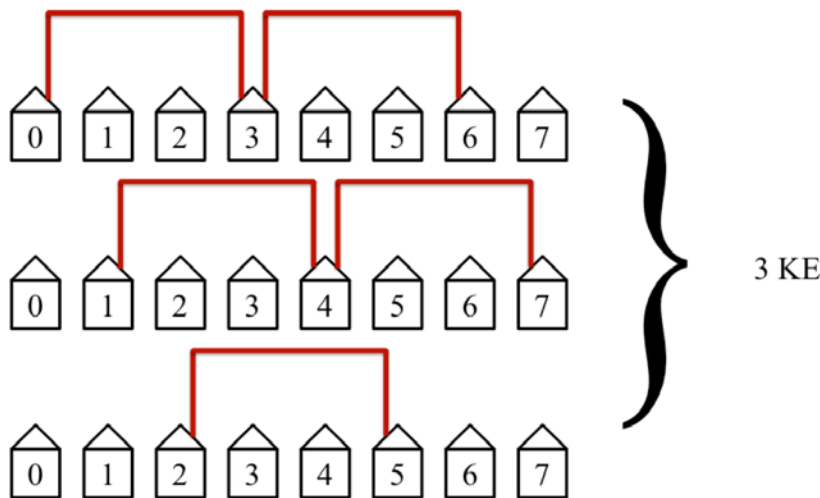


Figure 10. The third step in the protocol connects the third nearest neighbors. This step is not as efficient as the first two steps but still has simultaneous loops. This step requires 3 KEs to complete.

The protocol will then connect the fourth nearest neighbors as shown in Figure 11. This is above the midpoint for our example with $N=7$ and is the slowest and least efficient step in the protocol. The midpoint is considered to when the distance between Alice and Bob is equal to half the length of the network. These steps will take 4 KEs to complete. Simultaneous loops with disconnected hosts are no longer possible beyond the midpoint. The slowest and least efficient steps occurs at the midpoint of the protocol.

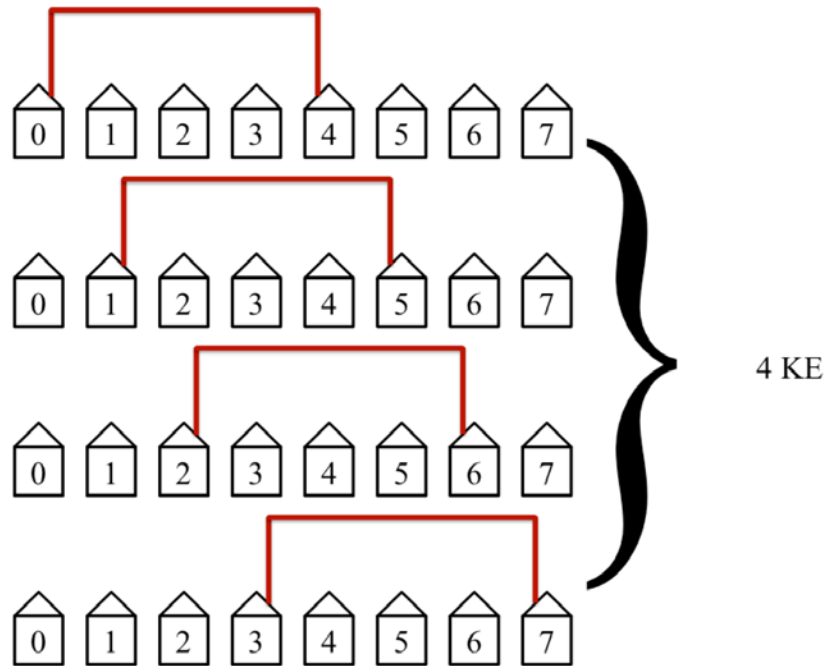


Figure 11. The fourth step in the protocol connects the fourth nearest neighbors. This step is the slowest and least efficient step in the protocol at our example of $N=7$. This step requires 4 KEs to complete.

The protocol will then connect the fifth nearest neighbors as shown in Figure 12. This step will take 3 KEs to complete. It is also inefficient since it is beyond the midpoint thus only a single loop is possible, but it requires fewer KEs since there are only three such pairs.

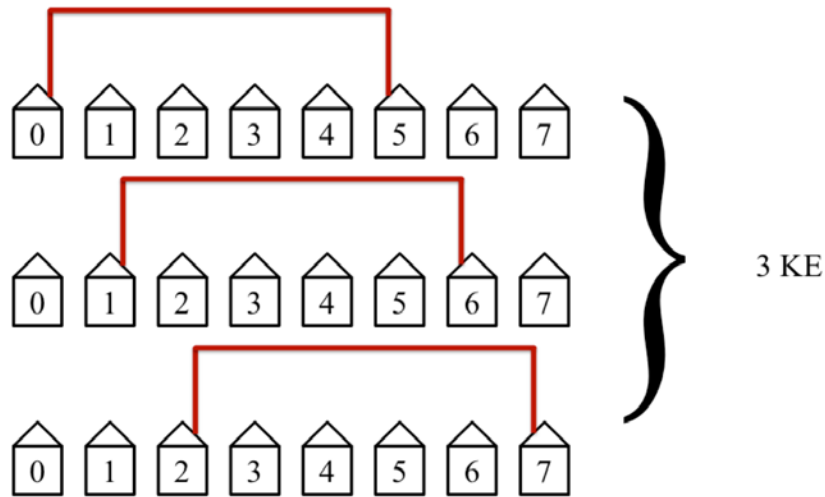


Figure 12. The fifth step in the protocol connects the fifth nearest neighbors. This step is not efficient since simultaneous non-overlapping loops with disconnected hosts cannot occur.

The protocol will then connect the sixth nearest neighbors as shown in Figure 13. This step will take 2 KEs because there are only two possibilities.

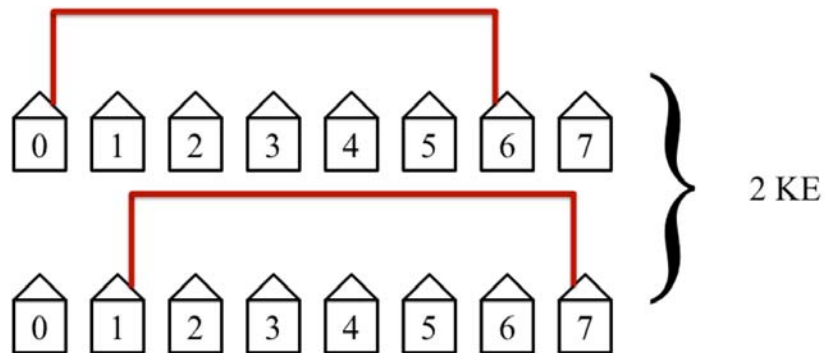


Figure 13. The sixth step in the protocol connects the sixth nearest neighbors. This step requires only 2 KEs since there are only two possibilities.

The protocol will then connect the sixth closest neighbors as shown in Figure 14. This will take 1 KE since there is only one such pair of hosts.



Figure 14. The seventh and last step. This step is not efficient but only requires one KE since there is only one such pair of hosts.

This completes the protocol for an example of size $N = 7$. Notice the pattern that occurs for N being odd. We have a pattern of 1 KE, 2 KE, 3 KE, 4 KE, 3 KE, 2 KE, and 1 KE. This is essentially Gauss's counting technique up to $N/2$ and back. The total number of KEs needed will be $1KE + 2KE + 3KE + 4KE + 3KE + 2KE + 1KE = 16KE$.

The speed or time requirement of the protocol for a network of arbitrary size N with N being odd is $\left(\frac{N+1}{2}\right)^2$ KEs and can be derived as follows.

Since N is odd we can express it as;

$$N = 2n + 1. \quad (2)$$

To find the midpoint we can solve n and express it in terms of N , this gives the following;

$$\frac{N-1}{2} = n. \quad (3)$$

The pattern when N is odd has the following form;

$$1 + 2 + \dots + (n-1) + n + (n-1) + \dots + 2 + 1 = \left(\frac{N-1}{2}\right)^2. \quad (4)$$

Expressing n in terms of N gives;

$$1 + 2 + \dots + \left(\frac{N-1}{2} - 1\right) + \left(\frac{N-1}{2}\right) + \left(\frac{N-1}{2} - 1\right) + \dots + 2 + 1 = \left(\frac{N-1}{2}\right)^2. \quad (5)$$

We know from Gauss's counting method that,

$$1 + 2 + \dots + N = \frac{N(N+1)}{2}. \quad (6)$$

In our pattern we can use Gauss's counting method twice to find the sum as follows.

$$\underbrace{1 + 2 + \dots + \left(\frac{N-1}{2} - 1\right)}_{\frac{\left(\frac{N-1}{2} - 1\right)\left(\frac{N-1}{2}\right)}{2}} + \left(\frac{N-1}{2}\right) + \underbrace{\left(\frac{N-1}{2} - 1\right) + \dots + 2 + 1}_{\frac{\left(\frac{N-1}{2} - 1\right)\left(\frac{N-1}{2}\right)}{2}} = \left(\frac{N-1}{2}\right)^2. \quad (7)$$

$$\left(\frac{\binom{N-1}{2} \binom{N-1}{2} - 1}{2} \right) + \binom{N-1}{2} + \left(\frac{\binom{N-1}{2} \binom{N-1}{2} - 1}{2} \right) = \left(\frac{N-1}{2} \right)^2. \quad (8)$$

This simplifies to

$$\left(\frac{N-1}{2} \right)^2 = \left(\frac{N-1}{2} \right)^2. \quad (9)$$

Thus the speed of the network is proportional to $\frac{N^2}{4}$ with N being odd and the size of the network. The pattern for when N is even is similar.

2.2.2 The network size N is an even number

For the sake of easier understanding, we will again illustrate the calculation of time requirement with examples shown in the following figures. In this example we have an even number as network size $N=8$. We have 9 host with index $i \{i \in \mathbb{N} : 0 \leq i \leq 8\}$. We have 8 intermediate connections between the first and last host.

The first step in the protocol connects the nearest neighbors. This step is the quickest and most efficient. It has the most simultaneous non-overlapping loops and requires only one KE to complete. The following figure illustrates this first step in the protocol.

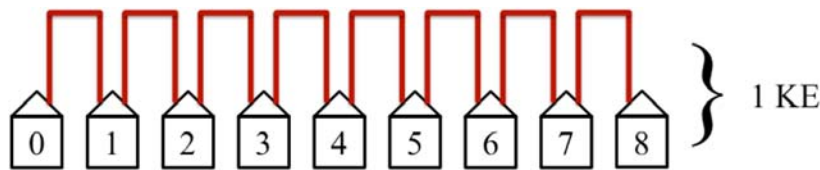


Figure 15. The first step in the protocol connects the nearest neighbors. This step is the quickest and most efficient. It has the most non-overlapping simultaneous loops and requires only 1 KE to complete.

The second step in the protocol will then connect the second nearest neighbors as shown in the following figure. This step will take two KEs to complete and has the second most simultaneous non-lapping loops. It is the second quickest and second most efficient step.

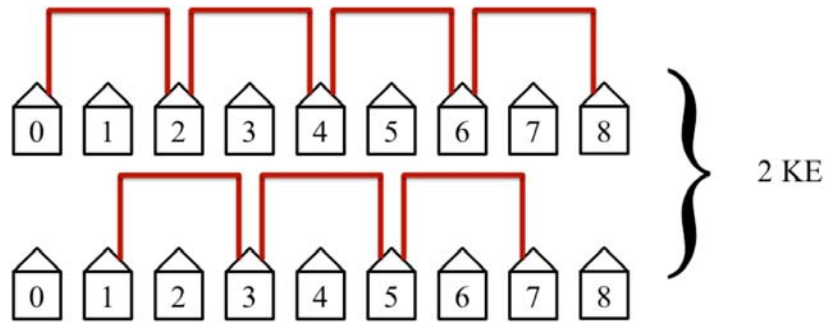


Figure 16. The second step in the protocol connects the second nearest neighbors. This step requires 2 KEs to complete.

The protocol will then connect the third nearest neighbors as shown in the following figure. This will take 3 KEs to complete and is not as efficient as the first two steps in the protocol but still has simultaneous loops in this example of $N = 8$.

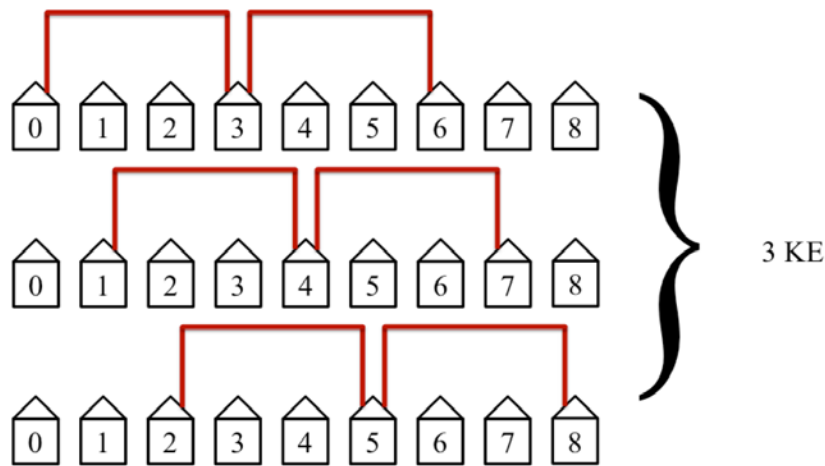


Figure17. The third step in the protocol connects the third nearest neighbors. This step requires 3 KEs to complete.

The protocol will then connect the fourth nearest neighbors as shown in the following figure. This is at the midpoint for our example with $N = 8$ and is the slowest and least efficient step in the protocol. The midpoint is defined when the distance between Alice and Bob is equal to half the length of the network. This step will take 4 KEs to complete. The slowest and least efficient steps occurs at the midpoint of the protocol.

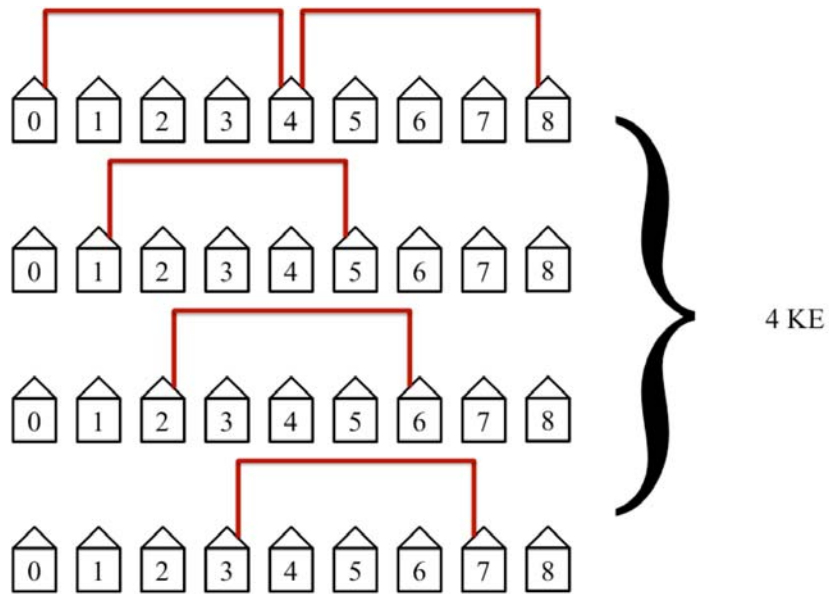


Figure 18. The fourth step in the protocol connects the fourth nearest neighbors. It requires 4 KEs to complete.

The protocol will then connect the fifth nearest neighbors as shown in the following figure. This step will take 4 KEs to complete. It is not efficient since it is at midpoint.

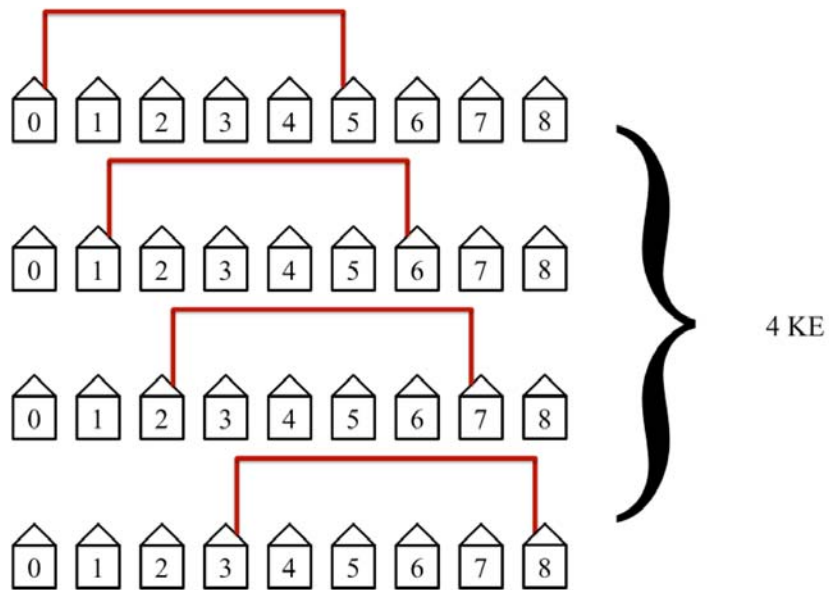


Figure 19. The fifth step in the protocol connects the fifth nearest neighbors. This step is not efficient since simultaneous non-overlapping loops with disconnected hosts cannot occur. It requires 4 KEs to complete.

The protocol will then connect the sixth nearest neighbors as shown in the following figure. This step will take 3 KEs because there are only three possibilities at this distance in this example of a network of size $N = 8$.

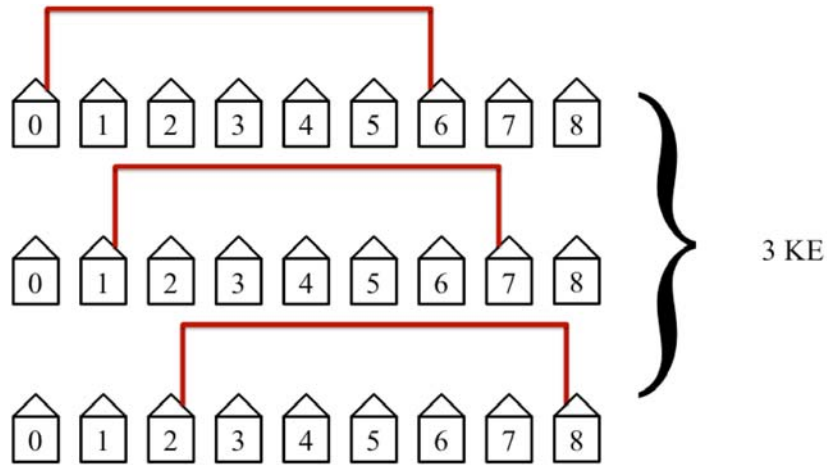


Figure 20. The sixth step in the protocol connects the sixth nearest neighbors. This step requires only 3 KEs since it is the third to last step and there are only three possibilities.

The protocol will then connect the seventh nearest neighbors as shown in the following figure. This will take 2 KEs since there are only two pairs of host with a length of seven hosts between them.

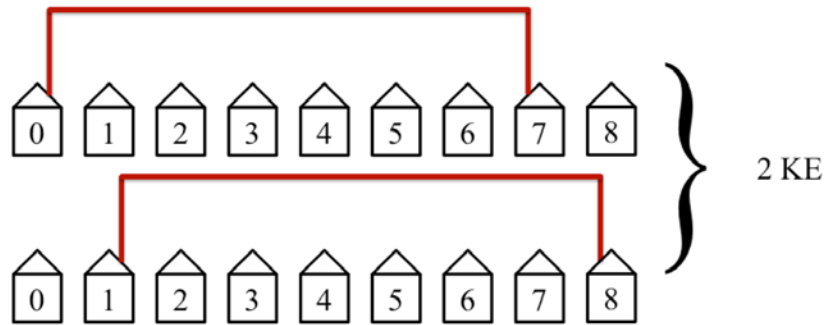


Figure 21. The seventh step in this example of a network of size $N = 8$. This step is not efficient but only requires two KEs since there are only two such pairs of host.

The last step in the protocol connects the first and last hosts. This step is the least efficient and requires the entire length of the network. Since there is only one pair of host at this length, this step requires only one KE.

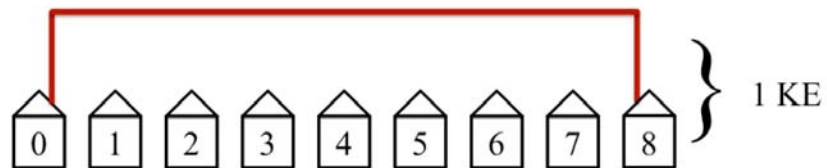


Figure 22. The last step in our example with $N=8$. This step is not efficient but only requires one KE since there is only one pair of host that are eight hosts apart.

Notice the pattern that occurs for N being even. We have 1 KE, 2 KE, 3 KE, 4 KE, 4 KE, 3 KE, 2 KE, and 1 KE. This is essentially Gauss's counting technique up to $N/2$ and back. The total number of KEs needed will be $1KE + 2KE + 3KE + 4KE + 4KE + 3KE + 2KE + 1KE = 20KE$. The time needed to connect the entire network will take 20 KEs which is approximately 40 seconds if B_{kljn} is 10 kHz and if the key is 100 bits long.

The speed or time requirement of the protocol for a network of size N with N being even between the first and last host is $\frac{N^2}{4} + \frac{N}{2}$ KEs and can be derived as follows.

With $N = 8$ the pattern in our example is;

$$\frac{N^2}{4} + \frac{N}{2} = 20KE \quad . \quad (10)$$

Since N is even we can express it as;

$$N = 2n \quad . \quad (11)$$

To find the midpoint we can solve n and express it in terms of N , this gives the following;

$$\frac{N}{2} = n \quad . \quad (12)$$

The general pattern when N is even has the following form;

$$1 + 2 + \dots + n + n + \dots + 2 + 1 = \frac{N^2}{4} + \frac{N}{2} \quad . \quad (13)$$

Expressing n in terms of N gives;

$$1 + 2 + \dots + \frac{N}{2} + \frac{N}{2} + \dots + 2 + 1 = \frac{N^2}{4} + \frac{N}{2} \quad . \quad (14)$$

We know from Gauss's counting method that,

$$1 + 2 + \dots + N = \frac{N(N+1)}{2} \quad . \quad (15)$$

In our pattern we can use Gauss's counting method twice to find the sum as follows.

$$\underbrace{1+2+\dots+\frac{N}{2}}_{\frac{\binom{N}{2}\binom{N+1}{2}}{2}} + \underbrace{\frac{N}{2}+\dots+2+1}_{\frac{\binom{N}{2}\binom{N+1}{2}}{2}} = \frac{N^2}{4} + \frac{N}{2}. \quad (16)$$

$$\frac{\frac{N}{2}\left(\frac{N}{2}+1\right)}{2} + \frac{\frac{N}{2}\left(\frac{N}{2}+1\right)}{2} = \frac{N^2}{4} + \frac{N}{2}. \quad (17)$$

This simplifies to

$$\left(\frac{N}{2}\right)\left(\frac{N}{2}+1\right) = \frac{N^2}{4} + \frac{N}{2}. \quad (18)$$

Thus the speed of the network is proportional to $\frac{N^2}{4}$ with N being the size of the network and even.

2.2.3 Protocol properties

The protocol is most efficient during the first half the network since this is when simultaneous non-overlapping loops can occur. It is least efficient during the second half since there are no possible non-overlapping loops that have not already occur.

The protocol is the quickest during the first and last steps; the second quickest step is the second and second to last steps. This continues until it reaches the middle steps; at these steps the protocol is the slowest.

The protocol is most active with the first and last host in the one-dimensional linear network. It is least active with the middle host. In the example with network of size $N = 8$ the first host was involved in 8 of the 20 KEs. Compare this the fifth host (which is in the middle of the network) which was involved with 4 of the 20 KEs. This is because the hosts located in the middle of the network are nearest to every single other host in the network and therefore requires fewer KEs. The hosts along the edges of the network are furthest from the other edge and requires more KEs to complete its key exchange with the other host in the network.

Acknowledgements

Discussions with Mladen Kezunovich and Karen Butler-Perry are appreciated.

References

1. Amin SM, Wollenberg BF (2005) Toward a smart grid. *IEEE Power & Energy Mag.* 3: 34-41.
2. Kezunovic M (2011) Smart Fault Location for Smart Grids. *IEEE Trans. Smart Grid* 2: 11-22.
3. McDaniel P, McLaughlin S (2009) Security and privacy challenges in the smart Grid. *IEEE Security & Privacy* 7: 75-77
4. Kundur D, Feng X, Mashayekh S, Liu S, Zourntos T, Butler-Purpy KL (2011) Towards modeling the impact of cyber attacks on a smart grid. *Int. J. Security and Networks* 6: 2–13.
5. Liang Y, Poor HV, Shamai S (2008) Information theoretic security. *Foundations Trends Commun. Inform. Theory* 5: 355-580. DOI: 10.1561/01000000036.
6. Yuen HP (2012) Unconditional Security In Quantum Key Distribution. ArXiv e-prints.
7. Gerhardt I, Liu Q, Lamas-Linares A, Skaar J, Kurtsiefer C, et al. (2011) Full-field implementation of a perfect eavesdropper on a quantum cryptography system. *Nature Communications* 2.
8. Lydersen L, Wiechers C, Wittmann C, Elser D, Skaar J, et al. (2010) Hacking commercial quantum cryptography systems by tailored bright illumination. *Nature Photonics* 4: 686-689.
9. Gerhardt I, Liu Q, Lamas-Linares A, Skaar J, Scarani V, et al. (2011) Experimentally Faking the Violation of Bell's Inequalities. *Physical Review Letters* 107.
10. Makarov V, Skaar J (2008) Fakes states attack using detector efficiency mismatch on SARG04, phase-time, DPSK, and Ekert protocols. *Quantum Information & Computation* 8: 622-635.
11. Wiechers C, Lydersen L, Wittmann C, Elser D, Skaar J, et al. (2011) After-gate attack on a quantum cryptosystem. *New Journal of Physics* 13.
12. Lydersen L, Wiechers C, Wittmann C, Elser D, Skaar J, et al. (2010) Thermal blinding of gated detectors in quantum cryptography. *Optics Express* 18: 27938-27954.
13. Jain N, Wittmann C, Lydersen L, Wiechers C, Elser D, et al. (2011) Device Calibration Impacts Security of Quantum Key Distribution. *Physical Review Letters* 107.
14. Lydersen L, Skaar J, Makarov V (2011) Tailored bright illumination attack on distributed-phase-reference protocols. *Journal of Modern Optics* 58: 680-685.
15. Lydersen L, Akhlaghi MK, Majedi AH, Skaar J, Makarov V (2011) Controlling a superconducting nanowire single-photon detector using tailored bright illumination. *New Journal of Physics* 13.
16. Lydersen L, Makarov V, Skaar J (2011) Comment on "Resilience of gated avalanche photodiodes against bright illumination attacks in quantum cryptography" *Appl. Phys. Lett.* 98, 231104 (2011). *Applied Physics Letters* 99.
17. Sauge S, Lydersen L, Anisimov A, Skaar J, Makarov V (2011) Controlling an actively-quenched single photon detector with bright light. *Optics Express* 19: 23590-23600.
18. Lydersen L, Jain N, Wittmann C, Maroy O, Skaar J, et al. (2011) Superlinear threshold detectors in quantum cryptography. *Physical Review A* 84.
19. Lydersen L, Wiechers C, Wittmann C, Elser D, Skaar J, et al. (2010) Avoiding the blinding attack in QKD reply. *Nature Photonics* 4: 801-801.
20. Makarov V (2009) Controlling passively quenched single photon detectors by bright light. *New Journal of Physics* 11.

21. Kish LB (2006) Totally secure classical communication utilizing Johnson (-like) noise and Kirchoff's law. *Physics Letters A* 352: 178-182.
22. Kish LB (2006) Protection against the man-in-the-middle-attack for the Kirchoff-loop-Johnson(-like)-noise cipher and expansion by voltage-based security. *Fluctuation and Noise Letters* 6: L57-L63.
23. Mingesz R, Kish LB, Gingl Z, Granqvist CG, Wen H, Peper F, Eubanks T, Schmera G (2013) Information theoretic security by the laws of classical physics. In: Balas VE et al. (Eds.), *Soft Computing Applications*, AISC 195, pp. 11–25 (Springer)
24. Mingesz R, Gingl Z, Kish LB (2008) Johnson(-like)-Noise-Kirchoff-loop based secure classical communicator characteristics, for ranges of two to two thousand kilometers, via model-line. *Physics Letters A* 372: 978-984.
25. Kish LB, Saidi O (2008) Unconditionally secure computers, algorithms and hardware. *Fluct. Noise Lett.* 8:L95-L98.
26. Kish LB, Peper F (2012) Information Networks Secured by the Laws of Physics. *Ieice Transactions on Communications* E95B: 1501-1507.
27. Kish LB, Mingesz R (2006) Totally secure classical networks with multipoint telecloning (teleportation) of classical bits through loops with Johnson-like noise (vol 6, pg C 9, 2006). *Fluctuation and Noise Letters* 6: L447-L447.
28. Balog R, Krein PT, Hamill DC (2000) Coupled Inductors—a Basic Filter Building Block. *Proc. Electrical Manufacturing and Coil Winding Ass.* 217-278. http://www.hamill.co.uk/pdfs/ciabfbb_.pdf
29. Kim S, Enjeti PN (2002) A new hybrid active power filter (APF) topology. *IEEE Transactions on Power Electronics* 17:48-54.

30. Kish LB (2013) Enhanced secure key exchange systems based on the Johnson-noise scheme. (Manuscript submitted for publication). <http://vixra.org/abs/1302.0055> ; <http://arxiv.org/abs/1302.3901>