G-RIPS Sendai 2025

Mitsubishi-B Group

Final Report

Key Relay Protocol for Quantum Key Distribution

Authors:
Ed Chen[1]
Maho Maruyama[2]
Derek Zhang[3]
Donald Zvada[4]

Mentors:
Yuji Ando[*]
Natsuo Miyatake[*]
Toyohiro Tsurumaru[†]
Go Kato[‡]

[1] New York University
[2] Ochanomizu University
[3] University of Maryland
[4] AIMS South Africa
[*] Academic Mentor, Tohoku University
[†] Industry Mentor, Mitsubishi Electric
[‡] External Advisor, NICT

August 7, 2025

Contents

1 Introduction

This is the final report for the Mitsubishi-B team at Graduate-level Research in Industrial Projects for Students (G-RIPS) – Sendai 2025. Our initial question involved a security analysis of quantum cryptographic key distribution with insecure relay nodes. A full description can be found at this link.

The core motivation is that novel quantum technologies may render classical cryptography obsolete. Current encryption schemes rely on computational hardness. For instance, RSA cryptography derives its security from the difficulty of large number prime factorization, which requires infeasible classical resources to solve. However, quantum methods such as Shor's algorithm can solve such problems efficiently, threatening existing security infrastructure.

Simultaneously, quantum key distribution has emerged as an information theoretically secure alternative which is provably unbreakable by quantum algorithms. However, this unconditionally secure method is limited to short distance links between adjacent nodes. We are interested in exploiting these links over a network to enable greater applicability on larger scales.

1.1 Key Takeaways

We explored a variety of research directions in this project. Our main results surround the properties that enable feasibility and equivalence of KRP and SNC, which are then applied to a few classes of simple graphs that we make results on. A concrete linear algebraic formulation of the KRP design problem is presented, which motivates an alternative interpretation of the problem which could be used to script a protocol discovery technique. A few other directions regarding protocol-specific counterexamples, network planarity investigations, and the complexity class of the KRP feasibility problem were investigated and are included in the report as well.

2 Background

2.1 Computer Communication

Modern internet and digital infrastructure are rooted in the construction of scalable and flexible computer networks that enable communication across digital devices. Computer networks consist of nodes which are interconnected to each other through links. The terms "node" and "link" may refer to a variety of technologies with the defining characteristic that nodes are sources or sinks of information, and links are means to transmit bits between these nodes. For instance, with the internet, hosts and routers would be examples of nodes, whereas optical, wired, and wireless connections are examples of links.

In determining where to send information in these networks, a plethora of protocols and security practices have been developed. There are over 2 billion computers in the world. It is obviously infeasible to connect each pair of computers that may want to communicate to each other. In practice, computers are attached in a local web of other computers that are close to each other, networks are attached together with an access network, which are attached together in a regional network, and so on and so forth. Data is split into packets which are then transmitted through these networks.

The question then arises as to how these networks should signify where a certain packet should be sent. This motivates the concept of network routing, which operates not unlike a cell phone number. Headers specifying destination and routing information are stored in a prefix, which is read by intermediary nodes to determine where information should be routed.

We now draw a distinction between network routing and network coding, which are two methods for data transmission. Recent research has shown that network routing alone does not achieve the maximum throughput possible across a communication network. Network coding refers to the process of having intermediary nodes transmitting output data that is an encoded combination of their received data. The throughput benefit of this computational mixing of intermediary nodes is the primary motivation behind network coding.

2.1.1 Security and Encryption

Computer networks are naturally susceptible targets for malicious attacks. Much of the traffic on the internet and security-critical computer networks utilize encryption protocols to prevent adversarial actors from gaining knowledge of sensitive data transmissions. It is helpful to define two types of malicious actors. The **passive adversary** has the ability to intercept transmissions, but does not change the behavior of the transmissions themselves. The **active adversary** has the ability to intercept transmissions and furthermore, the ability to change the content of those transmissions.

The **One-time pad** is an uncrackable encryption technique that operates on the *one-time* use of a preshared secret key. Given that the key is truly random, pre-shared and secret, and critically, only used once, no correlative results can be deduced by an adversarial party, and it is guaranteed to be information-theoretically secure. **Information theoretic security** is defined as a method from which an adversarial party with infinite computation power cannot break. This is as opposed to **computational security**, which operates on the computational hardness of a method to break.

2.2 Quantum Key Distribution

The key distribution protocol we will consider is BB84 [1]. The protocol requires an authenticated (classical[1]) public channel. The essence of BB84 is that Alice sends photons (e.g. through an optical fiber) to Bob, and a secret key is constructed from the transferred information. These photons have polarization in either the horizontal $0°/90°$ basis or the diagonal $45°/135°$ basis. Polarization is a quantum mechanical property, so that measurement along a specific basis will collapse the polarization state to one of the two states of that basis. For example, suppose Alice sends Bob a photon in the horizontal basis with polarization $0°$. An eavesdropper Eve doesn't know which basis that photon was prepared in. Half of the time she can guess the state correctly, but the other half, she measures the photon in the incorrect diagonal basis. Then the photon will be measured to have either $45°$ polarization, or $135°$ polarization. It is a fact from quantum mechanics that Eve destroys the original state of the photon and its accompanying quantum information. If Eve attempts to resend the photon along to Bob so as to not arouse suspicion, it may differ from the photon Alice sent. Thus, Alice and Bob can check for the presence of an eavesdropper by confirming the bases used over a public channel, and then checking if any photons have been adjusted. This public communication occurs *after* Bob has received (and destroyed) the photons Alice sent, meaning Eve cannot use that information to measure the photons. Moreover, Alice and Bob can determine with arbitrarily high confidence whether there is an eavesdropper, making the protocol unbreakable.

By checking the bases over the public channel, Alice and Bob can also detect whether the environment has adjusted the photons. This is the concept of **quantum decoherence**: a quantum system can only retain its original state for a short period of time before its quantum information is lost into the environment as noise. Greater physical distances separating Alice and Bob mean longer travel times for photons, making them more likely to decohere. At short distances, errors are few, so if Alice and Bob do detect an error, they can simply restart the protocol. At long distances, the protocol will need to be restarted many times before an error-free instance happens. This slows down communication greatly, eventually making it impossible.

Whilst quantum communication is theoretically unbreakable [6, 9, 11, 12, 13], there are a number of physical constraints and practical challenges preventing wider applicability. Firstly, the physical limitations of qubits mean they attain nontrivial error rates beyond 50-100km. Unlike classical bits, using any sort of optical amplifier to extend this link will collapse the state and disrupt the correctness of the quantum channels. In the rest of this work and project, QKD is treated as an unconditionally secure primitive, a method which can transmit a secret securely between two nodes, that has physical distance limitations.

Using QKD to generate secret keys for OTP, the secret key generation rate becomes

[1] We will describe any process that does not involve quantum mechanics as **classical**.

a bottleneck factor. As OTP requires equally-long keys to the secret to transmit, the secret key generation rate of QKD is directly equivalent to the secret throughput of data transmissions made through a QKD-enabled network. It is known that the rate of secret bit agreement falls with longer distances, and features a significant drop off beyond a maximum communication distance that can be reached. For current optical fibers, this distance is roughly limited to 100km, and the key generation rate is roughly on the order of 10^2 kbits/s, depending on the distance [11].

To reiterate, the base QKD primitive is between two parties, who over a relatively short distance, now with QKD have an unconditionally secure way to share secret keys. Indeed, a natural way to extend the applicability of this emerging technology is to construct a **QKD Network** through the connection of several QKD links. An immediate and obvious challenge is how we are then able to maintain the unconditional key exchange security that is resultant of a single QKD link over an entire QKD network, which has a plethora of nodes which may be intercepted by adversarial parties.

2.3 Graph Theory

Networks are modeled mathematically by graphs, which are the object of study in graph theory.

Definition 2.1. (Graphs)
A **graph** is a pair $G = (V, E)$ with **vertices** or **nodes** $V = \{1, \ldots, N\}$ and a multiset of **edges** $E \subset \{\{a, b\} \mid a, b \in V, a \neq b\}$ (this defines a **multigraph**, where a pair of nodes can have more than one shared edge).

- Two nodes are **adjacent** or **neighbors** if they share an edge. The **degree** of a node is the number of its neighbors.
- A **path** between two nodes a and b is an ordered set of distinct nodes $(a, \ldots, b)$ such that consecutive nodes are adjacent. A **cycle** is a path with least two distinct nodes, and $a = b$. A graph is **connected** if there is a path between any pair of nodes.
- **Edge contraction** on an edge $e = \{a, b\}$ in a graph involves deleting e and merging a and b into a new node c. All edges previously connected to either a or b are now connected to c.

Definition 2.2. (Minimum Edge Cut)
A **minimum edge cut** (min-cut) is defined as the smallest cut $\delta(S, \bar{S})$ which partitions a graph $G = (V, E)$ into two spanning node sets $S, \bar{S}$, which satisfy some condition. In our work, the minimum cut we often work with regards the minimal edge cut that separates the sender and receiver nodes of our user pairs.

$$\min |\delta(S, \bar{S})| \quad \text{subject to} \quad a_i \in S, b_i \in \bar{S} \forall (a_i, b_i) \in U$$

2.4 Information Theory

Information Theory is a framework for measuring or quantifying the amount of information that is gained or remain secure given some information has been learned or leaked to an adversary.

Definition 2.3. (Entropy)
Let X be a random variable with a Probability Mass Function (PMF) $P_X(x) = Pr[X = x]$. The **entropy** of X denoted by $H(X)$, quantifies the level of uncertainty or randomness in X and is defined as:

$$H(X) = -\sum_{\forall x} P_X(x) \log P_X(x)$$

Entropy gives us the number of bits we gain from the random variable X. In networking, entropy measures how unpredictable a secret key or message to an adversary.

Definition 2.4. (Conditional Entropy)
Let X and Y be jointly distrusted random variables. The **conditional entropy** of X and Y, denoted by $H(X|Y)$ measures the amount of uncertainty in the random variable X given we know the value(s) of Y.

$$H(X|Y) = \sum_{\forall y} P_Y(y) H(X|Y = y) = -\sum P_{X,Y}(x, y) \log P_{X|Y}(x|y)$$

This is useful in showing the amount of information that remains hidden to the adversary given some bits of information have been leaked.

Definition 2.5. (Mutual Information)
Mutual Information is a correlation between two random variables X and Y, denoted $I(X;Y)$, measures how much knowing Y reduces the uncertainty about X, and vice versa:

$$I(X;Y) = H(X) - H(X \mid Y) = H(Y) - H(Y \mid X)$$

Alternatively,

$$I(X;Y) = \sum_{x,y} P_{X,Y}(x, y) \log \frac{P_{X,Y}(x, y)}{P_X(x) P_Y(y)}$$

In a network coding, if $I(K; P_E) = 0$, this means the adversary gains no information about the key K from observing the edge set E. The key and the set of edges wiretapped by the adversary are linearly independent.

Definition 2.6. (Conditional Mutual Information)
The **conditional mutual information** between two variables X and Y given bits information about another variable Z. This is denoted by $I(X, Y \mid Z)$ measures the mutual information between X and Y when Z is known.

$$I(X, Y|Z) = H(X|Z) - H(X|Y, Z)$$

This is equivalent to,

$$I(X, Y \mid Z) = \sum_{\forall x,y,z} P_{X,Y,Z}(x, y, z) \log \frac{P_{X,Y,Z}(x, y \mid z)}{P_{X|Z}(x \mid z) P_{Y|Z}(y \mid z)}$$

This is useful when analyzing the protocols that must remain secure given some bits of information of the variable Z to an adversary, for example, if Z is a set of information that has been made publicly available via Public Channels.

2.5 Computer Networking

A **network** (G, U) is a graph G where nodes are defined to be users and edges are links between them, along with a set of communicating user pairs $U = \{(u_1^i, u_2^i) \quad \text{or} \quad (a_i, b_i) \mid i = 1, \ldots, n\}$ who want to communicate to each other (see Figure 1). An **adversary** is an outside observer not in the network. A **primitive** is a basic tool of communication which can be used on a network. Edge primitives release information that is either secret (known to only a specified set of users) or public (known to all users and adversaries). A **network protocol**, or simply a protocol, is a description of a set of primitives which can be used to communicate information, along with a stated goal of communication. In this report, all information occurs as single bits unless otherwise specified. An **instance** of a protocol is a specific set of instructions for using tools of a protocol to achieve the specified goal.

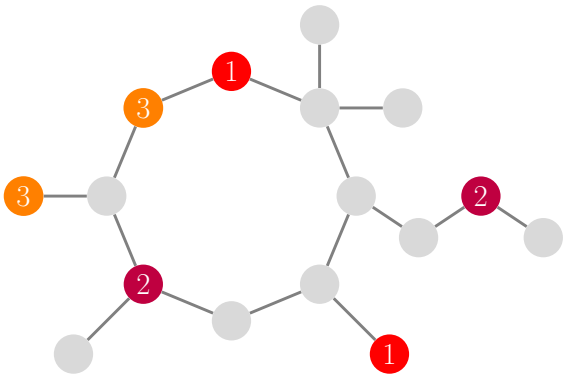

Figure 1: An example of a network with three communicating user pairs labeled.

For all of the following protocols, each node can perform bit addition (equivalently, XOR or addition in $\mathbb{Z}_2$) with the information it knows.

Definition 2.7. (Secret Channels, Public Channels, Local Key Sources, Random Bit Generation at a Node)

A **secret channel** is defined as a primitive node to node link that privately sends, through some encrypted means, information from one node to another. This is in practice realized through techniques such as a one-time pad, and hence in our work it is

assumed that in any given protocol a secret channel admits only a *single use*. If a secret channel is wiretapped, it sends the transmitted information to the adversary as well.

A **public channel** is defined as a primitive collection of node to node links that publicly broadcasts unencrypted information entered from any of its constituent nodes to every other node. The critical attribute is that information in the public channel is unencrypted and universally broadcast. The public channels admit unlimited usage and any information sent through a public channel is accessible to an adversarial party.

A **local key source** is defined as a device lying on an edge that upon an activation input from either endpoint, returns a randomly generated key to each endpoint. When wiretapped, the local key source also sends this key to a wiretapper. The definition of a local key source is designed to be more admissible to different technologies and protocols, however, in our work this is assumed to be realized by a QKD link. As such, it is in practice unable to be passively wiretapped, and can be treated as an information theoretically secure primitive.

Node-generated random bits is defined as the capability of intra-network nodes to generate random bits and use these as shared keys between two separate parties. In other words, it allows the ability for the random shared key to not be generated at an endpoint, but instead within a node besides the endpoints.

Remark. Local key sources are implemented by QKD links, and are unconditionally secure. In practice, KRPs follow the node adversary model, where nodes are wiretapped instead of edges. Specifically, if a node is wiretapped then all of information on the node's edges is made available to the adversary. Thus, our the node adversary model can be entirely contained within the edge adversary model, and it suffices to consider the latter.

Definition 2.8. (Key Relay Protocol)
Key Relay Protocol (KRP) makes use of local key sources, public announcements, and random bit generation at a node (see Figure 2). Each edge is associated with a local key source. The goal is for each user pair to share a random bit.

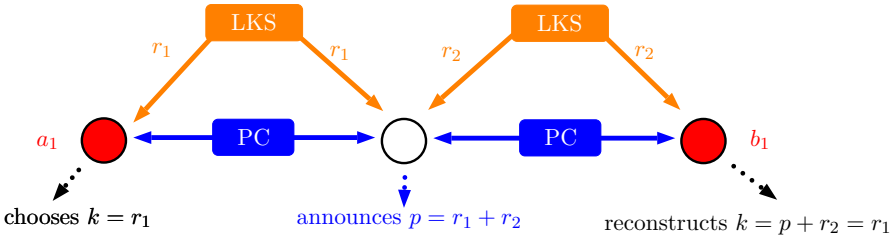

Figure 2: A basic example of KRP. Each edge e_i is associated with a local key source which distributes a random bit to both ends. Each node can also use public channels freely. User pair (a_1, b_1) wishes to share a relayed key k $= (k^1, k^2)$. To this end, the midpoint announces $p = r_1 + r_2$, enabling user a_1 and b_1 to each calculate $k^1 = k^2 = r_1$.

Definition 2.9. (Secure Network Coding)
Secure Network Coding (SNC) makes use of secret channels (see Figure 3). Each edge is associated with a secret channel. The goal is for each u_1^i to send a predefined random bit to u_2^i.

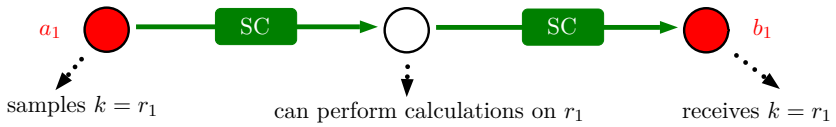

Figure 3: A basic example of SNC. The key propagates in a single direction.

Definition 2.10. (Secure Network Coding with Symmetric User Pairs)
Secure Network Coding with Symmetric User Pairs (SNC-S) makes use of secret channels. Each edge is associated with a secret channel. The goal is slightly modified compared to SNC: in each pair, either u_1^i or u_2^i can send a predefined random bit to the other.

Definition 2.11. (KRP by SNC settings)
KRP by SNC settings (KRP-by-SNC) makes use of secret channels and random bit generation at arbitrary nodes. Each edge is associated with a secret channel, where one user can send a secret bit along the edge. Wiretapping the edge reveals the bit to the adversary. The goal of KRP-by-SNC is the same as in KRP: each user pair must share a random bit.

SNC-S and KRP-by-SNC can be thought of as variants of SNC. Note that SNC $\subseteq$ SNC-S $\subseteq$ KRP-by-SNC in the sense that each protocol has more allowed primitives and/or is more flexible than the previous. The meaning of $\subseteq$ is formalized in Definition 2.15. For results in this paper comparing SNC and KRP, we will generally use KRP-by-SNC in place of standard SNC. The properties of KRP-by-SNC, specifically random bit generation and symmetry of user pairs, are more suited for the goal of key distribution. Thus, KRP-by-SNC is a "fairer" comparison with KRP, while retaining the core functionality of SNC.

Definition 2.12. (SNC with Public Channels)
Secure Network Coding with Public Channels (SNCwPC) makes use of secret channels and public announcements. Each edge is associated with a secret channel, where one user can send a secret bit along the edge. Wiretapping the edge reveals the bit to the adversary. Each node can announce an arbitrary number of bits. The goal is for each u_1^i to send a predefined message m_i to u_2^i (note that m_i is a key in a restricted sense).

Definition 2.13. (KRP with Constant Public Announcements)
If we restrict the KRP protocol as defined to only allow for fixed public announcements, it becomes a conventional SNC protocol and admits the security and capabilities of SNC.

Note this is a change of definition as the local key sources are by definition, independent and random. Hence, this definition is for illustrating purposes, and not a practical protocol. If we fix every public announcement P, any two adjacent edges are definitively correlated. For ease of consideration, we fix every $P = \mathbf{0}$. This augments the ability to "fix" the transmissions sent on the local key sources on each edge, which then admits no extra information hidden from the adversary.

In an information theoretic context, the public channels then admit no additional information on the shared keys $I(K; L_w, P) = I(K; L_w)$ and from a capability standpoint it is equivalent to SNC.

Definition 2.14. (Security of Protocols)
A **wiretap set** E is a set of edges whose associated information is made available to an adversary. A **wiretap collection** $\mathcal{E}$ is a collection of wiretap sets. A protocol $\mathbf{A}$ is said to be **secure** on $(G, U, \mathcal{E})$ if there exists an instance of $\mathbf{A}$ on (G, U) satisfying the following:

- For each user pair $(u_1^i, u_2^i) \in U$, both members deterministically reconstruct the same independent, random **key** k_i.
- Define all public information to be P. For each $E \in \mathcal{E}$, define I_E to be the information associated with E. Then $H(k_1, \ldots, k_n \mid P, I_E) = n$, i.e. the adversary gains no information about the keys.

Then we write $(G, U, \mathcal{E}) \in \mathbf{A}$. Note that the same instance of $\mathbf{A}$ must simultaneously tolerate every wiretap set in $\mathcal{E}$ (one cannot choose a different instance for each E in $\mathcal{E}$).

Protocol $\mathbf{A}$ is said to be **sound** on (G, U) if it is secure on $(G, U, \emptyset)$, i.e. communication is possible in the absence of adversaries.

Definition 2.15. (Equivalence of Protocols)
Protocol $\mathbf{A}$ is said to be more secure than protocol $\mathbf{B}$ if

$$\forall (G, U, \mathcal{E}), \mathbf{B} \in (G, U, \mathcal{E}) \implies \mathbf{A} \in (G, U, \mathcal{E}).$$

Then we write $\mathbf{B} \subseteq \mathbf{A}$: any level of security attained by $\mathbf{B}$ can also be attained by $\mathbf{A}$. If $\mathbf{A} \subseteq \mathbf{B}$ and $\mathbf{B} \subseteq \mathbf{A}$, then $\mathbf{A} \cong \mathbf{B}$ and the two protocols are said to be equivalent, achieving the same level of security.

Ref. [6] proved that SNC $\subseteq$ KRP-by-SNC $\subsetneq$ KRP $\cong$ SNCwPC. Given KRP

Theorem 2.16. (SNCwPC = KRP, Ref. [6])
We provide a summary of Theorem 1 in Ref [6], demonstrating that KRP and SNC schemes with Public Channels are equivalent by showing that each protocol can simulate the other's core components using its own resources. Formally, for any secure KRP protocol L, there exists a secure SNCwPC protocol L' such that for every user

pair (u_i^1, u_i^2), the relayed key $k_i = (k_i^1, k_i^2)$ established in L can be used in L' to securely transmit a message m_i using the one-time pad $c_i = m_i \oplus k_i^1$, ensuring correctness $\hat{m}_i = c_i \oplus k_i^2 = m_i$ and secrecy from the adversary.

Conversely, for any secure SNCwPC protocol L, there exists a secure KRP protocol L' such that every secret channel SC_e in L is replaced by a local key source LKS_e and a public channel PC_e, using a shared random key $r_e \in_R \{0,1\}$ and OTP encryption $p_e = s_e \oplus r_e$. The construction ensures that the simulated message delivery and adversary's view in L' are identical to those in L. Hence the two are equivalent.

Theorem 2.17. (KRP-by-SNC $\subsetneq$ KRP, Ref. [6])
We will provide an abridged summary of Theorem 2 in Ref [6]. Through the use of a counterexample graph G_0, which is constructed of 18 butterfly gadgets wired in a specific order, the authors show a working KRP protocol with two-round public announcements, whilst using **causality** to show that an equivalent KRP-by-SNC is impossible. This shows a definitive gap between the KRP-by-SNC and KRP protocols.

The source of the impossibility of a KRP-by-SNC protocol succeeding on G_0 is due to a requirement on the sequence of edge transmissions within the butterfly gadget, where certain traversal patterns require the simultaneous input of two edges so the single use of the secret channel on the middle edge can encode both inputs to the downstream nodes. The contradiction in their example arises from the fact that the positioning of the user pairs requires a butterfly gadget to fire before both inputs are able to reach the gadget, hence coined a causality-based contradiction.

3 Results

3.1 Mathematical Formulation

We introduce a framework for studying KRP using linear algebra.

3.1.1 Notation

Definition 3.1. (Incidence Vector Representation)
On any graph $G = (V, E)$ with edges $E = e_i$, we can represent any set of edges as an incidence vector, a vector v of length $|E|$ which admits a 1 in position v_i if the edge exists and a 0 if the edge does not exist.

Any information generated within G can be seen to have a incidence vector representation. Whether it be public announcements or the key transmissions, each is generated by some combination of the random primitive keys that are housed on each edge. As we are operating in bit wise arithmetic, combinations of these vectors are calculated additively in $\mathbb{Z}_2^n$, otherwise equivalently $GF(2)$. Equipped with this definition, we can say that the incidental edge vectors e_i are the canonical basis of the vector space $\mathbb{Z}_6^n$ where $n = |E|$.

3.1.2 Adversarial Secrecy

Definition 3.2. (Linear Algebraic Secrecy Formulation)
The adversary with wiretap set E_w has access to all information in the vector subspace A_{E_w} that is spanned by the incidence vectors of the edges it has wiretapped, and all the public announcements that have been made.

$$A_{E_w} = \text{span}\{[v_{e_i} | e_i \in E_w] \cup P\}$$

We say that a chosen key is insecure if it is linearly dependent on the information the adversary has. Equivalently, this means that the incidence vector of the key, v_k

$$v_k \in A_{E_w}$$

or equivalently that

$$\text{rank}(A_{E_w} \cup v_k) = \text{rank}(A_{E_w})$$

Definition 3.3. (Linear Algebraic KRP Statement)
Given a graph $G = (V, E)$, with a wiretap set E_w, and a number of user pairs

$$U = \{(a_1, b_1) \dots (a_i, b_i)\}$$

the Key Relay Protocol is defined as a choice within a design space of the timing and content of public announcements P at each node n_i. For each node n_i, allowable public announcements are within the span of the adjacent edges to m_i and the existing public information.

$$P = p_1, p_2, \dots p_n \quad \text{where node } n_i \text{ can make} \quad p_n \in \text{span}\{[v_e | n_i \in e] \cup P_{1 \dots n-1}\}$$

The protocol requires for a choice of keys

$$k \in \text{span}\{E\}$$

and that the design of public announcements must allow for the key of a relevant user pair to be constructed by both endpoints, that is

$$k \in \text{span}\{[v_e | a_i \in e], P\} \qquad k \in \text{span}\{[v_e | b_i \in e], P\}$$

whilst requiring privacy from an adversarial party, such that

$$k \notin \text{span}\{A_{E_w}\}$$

Note that this description of the choice of key applies both when the key is generated with the random primitive keys housed on each edge or at the nodes themselves, as in both cases we are primarily concerned with which of the random primitive keys remain applied in the choice of shared key.

An alternative but equivalent formulation is to consider the difference between the accessible information by each user pair (a, b). The difference of these values must lie in the span of the public information, $\mathbf{P}$.

$$k = \sum_{a \in e} \alpha_i v_e - \sum_{b \in e} \beta_i v_e \in \text{span}(\mathbf{P})$$

Example. (Linear Algebraic KRP Example)
We demonstrate the linear algebraic formulation on a single user pair, split path case. A single user pair (a, b) is connected with the following wiring of 4 relay nodes and 6 edges, where each edge e_i admits a secret l_i.

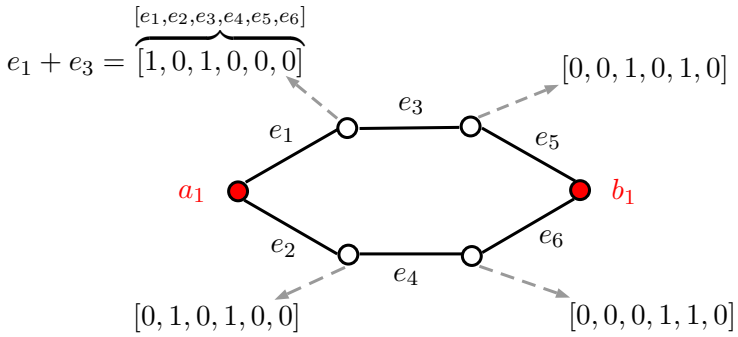

Figure 4: A simple one user pair, four relay node KRP with the choice of public announcements that each relay node announces the sum of its incident edges.

We demonstrate the KRP example protocol where each node announces the sum of all the local key sources on its attached edges. Evidently, the edge vectors are the canonical basis to $\mathbb{Z}_2^6$. We make a choice of a key

$$k = l_1 + l_2 \coloneqq [1,1,0,0,0,0]^T$$

and the public announcements are given as

$$P = \{[1,0,1,0,0,0]^T, [0,0,1,0,1,0]^T, [0,1,0,1,0,0]^T, [0,0,0,1,0,1]^T\}$$

corresponding to the expressions at the end of each dashed line in the figure. It is evident that k is in the span of b's accessible information

$$k \in \text{span}\{[e|b_i \in e], P\}$$

$$[1,1,0,0,0,0]^T \in \text{span}\{[0,0,0,0,1,0]^T, [0,0,0,0,0,1]^T \cup P\}$$

Now consider an adversary which has access to wiretap set $E_w = \{e_3, e_4\}$. Constructing the adversary's subspace, we have that

$$A_{E_w} = \text{span}\{[0,0,1,0,0,0]^T, [0,0,0,1,0,0]^T \cup P\}$$

In this specific wiretap set, we can then see that the KRP we have described is insecure, as in $\mathbb{Z}_2^6$

$$k = [1,1,0,0,0,0]^T = [0,0,1,0,0,0]^T + [0,0,0,1,0,0]^T + [1,0,1,0,0,0]^T + [0,1,0,1,0,0]^T$$

and as such lies in the adversary's subspace,

$$k \in A_{E_w}$$

Question. (Wiretap Set and Collection Relation)
If a KRP or KRP-by-SNC L_{KRP} or L_{KSNC} on a graph and user pair configuration (G, U) achieves the same security on every possible wiretap set on G, that is, the $2^{|E|}$ possible subsets of edge sets, do they achieve the same security on every possible wiretap collection $\mathcal{E}$?

That is, if two protocols achieve the same security on the set $\mathcal{E}_A$ of every possible wiretap configuration on a graph $G = (V, E)$, do they achieve the same security on every set $\mathcal{E} \subset \mathcal{E}_A$. We have differing speculations about this question, as the wiretap sets a specific instance of a protocol is admissible towards may not be the same with another protocol.

3.1.3 Discussion

In a one user pair scenario, a reduction can be made to allow only simultaneous public announcements, removing one dimension of the KRP design space. Any public announcement lies in the same vector space generated by the random bits in the network. In a single user pair, any intermediate computation that a public announcement would make that is dependent on previous public announcements can be moved to the receiver instead, and it is just a cosmetic difference if that computation is executed through subsequent public announcements or by the receiving node.

In addition, there are several restrictions we can place on the design of public announcements. A natural one is to require that all public announcements be the combination of an even number of random bits. This is because then every subsequent derivations will always be the XOR of an even number of bits, which reduces entropy loss in contrast with allowing odd combinations of random bits. In a similar vein, a public announcement would not announce a single random bit. This would simply remove entropy from the graph, and only aid the adversary. An example argument we would seek to make is that on planar networks, *effective* public announcements should only announce sums of faces for instance, which we could then draw results from.

3.2 Theory

Lemma 3.4. (Edge-Disjoint Existence)
If there exists an edge-disjoint non-wiretapped path between each user pair, there exists a successful KRP and SNC. This is a trivial result.

Theorem 3.5. (User Pair Minimum Cut Feasibility Bound for KRP)
Given n user pairs on an undirected graph G, with a minimum cut of size m separating the set of nodes denoted as senders S and the set of nodes denoted as the receivers R, a necessary but not sufficient condition for the KRP feasibility is that

$$n \leq m$$

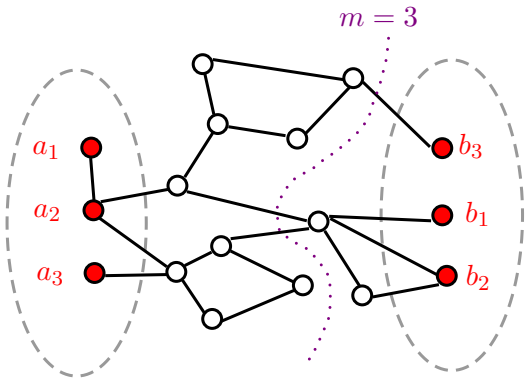

Figure 5: A visual explanation of the minimum cut feasibility bound, the grouping of the user pairs into two distinct vertex sets and then the minimum edge cut that disconnects these vertex sets.

We have n user pairs (a_i, b_i) who want to each share a secure, random key k_i. We refer to the smallest cut $\delta(S, \bar{S})$ to which S contains all a_i and $\bar{S}$ contains all b_i as the minimum cut. The size of this minimum cut is taken to be $|\delta(S)| = m$.

On each edge of this minimum cut, we number them e_j where $j = 1 \dots m$ and on each edge there lives a local key source realized by a QKD link, which shares the primitive local key l_j securely between one node in S and one node in $\bar{S}$. Assume each key is of a unit bit length.

The public channel announcements are treated as a all-encompassing P. Define the multi-index vector variables $K = \{k_i\}_{i=1\dots n}$ and $L = \{l_j\}_{j=1\dots m}$. We make a brief argument on a deterministic result for the KRP.

Lemma 3.6. (Deterministic Condition for KRP)
Given a graph G, for a KRP to attain correctness, that is each sender and receiver pair agree on the shared key in any situation, it is necessary but not sufficient to say

$$H(K|L, P) = 0$$

or equivalently, that there exists a deterministic function such that

$$K = f(L, P)$$

Proof. Suppose that $H(K|L, P) \neq 0$. Then there exists at least two possible execution states a and b, of which we denote their L, K, P as $L_{\{a,b\}}$, $K_{\{a,b\}}$, and $P_{\{a,b\}}$. Then we know that $L_a = L_b$ and $P_a = P_b$ but $K_a \neq K_b$. Every possible piece of public information on the cut is some function of L, P, and the randomness generated within the nodes of that side of the cut, say X_s. As L, P are identical, then the information on the cut in both executions is necessarily identical.

We can then construct a third execution state where we take execution a's state from the subset of nodes S, and execution b's state from the subset of nodes $\bar{S}$. In this scenario, the user pair halves within S resolve K_a while the user pair halves within $\bar{S}$ resolve K_b. Evidently, this violates correctness and thus $H(K|L,P) = 0$ necessarily. □

We return to our discussion of the original theorem. The reconstruction of the keys by b is performed with some bijective linear function $f_i(L,P)$ for all b_i. Evidently, the security of our protocol relies on three properties

1. $H(K|L,P) = 0$, that is given L and P, one can deterministically resolve the keys K. We call this the **constructibility** condition. This is taken from Lemma 3.6.
2. $I(K;P) = 0$, that is, the public channel announcements do not admit information gain on the secret keys. We call the **privacy** condition.
3. $H(K) = n$, that is the secret keys are independent and uniformly generated. We call this the **independence** condition.

We appeal to the following information-theoretic inequality

$$\begin{aligned} H(K) &= \underbrace{\cancel{H(K|L,P)}}_{\text{via constructibility}} + \underbrace{\cancel{I(K;P)}}_{\text{via privacy}} + I(K;L|P) \\ &= I(K;L|P) \\ &\leq H(L|P) \\ &= H(L) \\ &= m \end{aligned}$$

The first line can be intuitively thought to be that the deterministic information about the keys is entirely dependent on L, conditioned on prior knowledge of P. This gives, for **constructibility** and **privacy** we require $H(K) = m$ while for **independence** of the keys we require $H(K) = n$. As such, the feasibility constraint is that given n user pairs and a min-cut set m, a necessary but not sufficient condition is that

$$n \leq m$$

Lemma 3.7. (Wiretap and General Extensions of Theorem 3.5)
We can further refine this to wiretapped edges by considering the information on the wiretapped edges W. Consider the following relation with $L - W$ and $P + W$. The analagous relation holds, and $m' = |L - W|$ can be thought of the wiretap-restricted feasibility constraint.

To see why this generalizes to any linear network code, note that under a bijective function f_b,

$$H(X) = H(f_b(X))$$

It also generalizes to unequal link capacities. Treat each link larger than unit capacity as multiple links in our described construction. Lastly, this may work for SNC as well, simply treat $P = \emptyset$.

Theorem 3.8. (Generalized Menger's Theorem)
Menger's theorem states that the size of the minimum edge cut $\delta(S, \bar{S})$ between any two sets of nodes, S and $\bar{S}$, is equivalent to the size of the number of pairwise edge-disjoint paths between these two sets.

Proof. We provide a brief proof sketch. Denote an AB path to be a path between A and B. Denote an AB separator is a set of k nodes S such that $G - S$ contains no AB path. Denote an AB connector of size k is a union of k node-disjoint AB paths. The minimum size of an AB separator is the maximum size of an AB connector. The proof of these is accomplished via an induction on the number of edges in G. □

Lemma 3.9. (Edge-Disjoint Interpretation of 3.5)
A necessary but not sufficient condition for the existence of a successful KRP is that for the partition of n user pairs into two equally sized node sets A and B, there exists at least n pairwise edge-disjoint paths between A and B. This is a direct application of Theorem 3.8 to Theorem 3.5.

Theorem 3.10. (Path Interpretation of KRP Feasibility)
Given a graph G and n user pairs, if there are n edge-disjoint paths connecting each of the user pairs, then a KRP and SNC are both equivalently feasible. If there do not exist n edge disjoint paths between the user pairs, but there exists n edge disjoint paths between the node sets encompassing a separating partition of the user pairs, where of every pair (a_i, b_i) one is in S and one is in $\bar{S}$, lies a more complicated situation.

Conjecture 1. (Min-Cut Result Extension)
Given a graph G and n user pairs, if the sender and receiver sets satisfy Theorem 3.5 but there does not exist an edge disjoint path connecting each specific user pair (a_i, b_i) as stated in Lemma 3.4 it is conjectured the minimum cut bound can be tightened to impose a stronger bound on the minimum cut between the vertex sets.

This is justified by the butterfly coding network, which presents the smallest coding example in which the user pairs are switched between the top and bottom links. The minimum cut between the vertex sets needs to be 3, which is larger than the 2 distinct user pairs, since the pairwise sum of the two keys needs to be communicated to the two later nodes to allow them to derive their resulting keys.

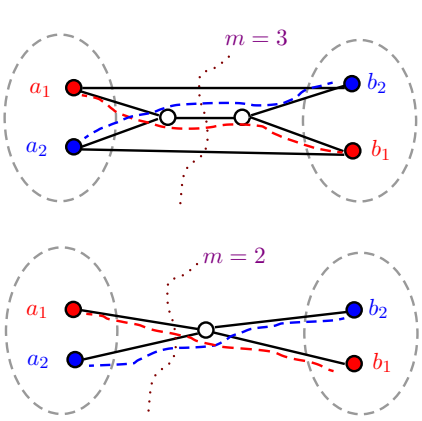

Figure 6: A demonstration of two user pair configuration where in the upper diagram the user pairs do not attain edge-disjoint paths, but in the lower diagram edge-disjoint paths are attainable. The conjecture deals with the upper diagram.

3.3 Graph Generalizations

This section contains tools for studying networks. The first two reduction tools allow us to simplify graphs without changing the equivalence or non-equivalence of SNC, KRP-by-SNC, and KRP.

A **leaf node** is defined to be a node with a single neighbor, which is attached by a **leaf edge**.

Theorem 3.11. (Leaf Reduction) Contracting a leaf edge does not change the security of SNC, KRP-by-SNC, or KRP. If the edge is incident on exactly one communicating user, that user becomes the combined node. This does not hold if the edge is incident on two communicating users.

Proof. Suppose l is a leaf node which has neighbor a. We would like to contract leaf edge e between l and a, which become combined node a'. Rigorously, we want to show that SNC, KRP-by-SNC, or KRP are secure on $(G, U, \mathcal{E})$ iff SNC, KRP-by-SNC, or KRP are secure on $(G', U', \mathcal{E}')$ modified in this way.

$\Longrightarrow$: Given an SNC, KRP-by-SNC, or KRP protocol on $(G, U, \mathcal{E})$, construct a protocol on $(G', U', \mathcal{E}')$ where

- The random bit generation at l and a is performed by a'.
- In KRP, the local key source on e is replaced with random bit generation on a'.

Then a' has all the information needed to make the announcements of both l and a. All remaining nodes know the same amount of information or more, and the adversary knows the same amount or less (less if e is in $\mathcal{E}$). Thus, the protocols proceed successfully as before.

$\Longleftarrow$: Now we want to add a leaf node, which involves considering additional wiretap sets. Given an SNC, KRP-by-SNC, or KRP protocol on $(G', U', \mathcal{E}')$, construct a protocol on $(G, U, \mathcal{E})$ which performs the same actions while ignoring the new leaf edge and node. The protocols proceed successfully as before. Additionally, define $\mathcal{E}$ to be $\mathcal{E}'$ with e appended to every wiretap set $E' \in \mathcal{E}'$. Since e is not used, this does not change the security. This modification accounts for every new wiretap set created by the addition of e. If l is attached to a member of a user pair, we have the choice to move that user to l. Construct $(G, U, \mathcal{E})$ as before; a knows the user pair's key. Then send that key along e via secret channel or simulation thereof. Then by the min-cut bound, the e cannot appear in any wiretap set, which accounts for every new wiretap set created. □

Theorem 3.12. (Degree Two Reduction) Given a degree two node which is not a communicating user, contracting one of its incident edges does not change the security of SNC, KRP-by-SNC, or KRP.

Proof. Suppose nodes a, b, and c are adjacent in that order, where b has degree two and is not a communicating user. Let a and b be connected by edge e, and b and c by edge f We would like to contract e, so a and b become combined node a'. Rigorously, we want to show that SNC, KRP-by-SNC, or KRP are secure on $(G, U, \mathcal{E})$ iff SNC, KRP-by-SNC, or KRP are secure on $(G', U', \mathcal{E}')$ modified in this way.

$\Longrightarrow$: Given an SNC, KRP-by-SNC, or KRP protocol on $(G, U, \mathcal{E})$, construct a protocol on $(G', U', \mathcal{E}')$ where

- The random bit generation at a and b is performed by a'.
- In KRP, the local key source on e is replaced with random bit generation on a'.

Then a' has all the information needed to make the announcements of both a and b. All remaining nodes know the same amount of information or more, and the adversary knows the same amount or less (less if e is in $\mathcal{E}$). Thus, the protocols proceed successfully as before.

$\Longleftarrow$: Now we want to add a degree two node b, which involves considering additional wiretap sets. Given an SNC, KRP-by-SNC, or KRP protocol on $(G', U', \mathcal{E}')$, construct a protocol on $(G, U, \mathcal{E})$ as follows:

- For SNC or KRP-by-SNC, WLOG a' sends a message m to c on G'. On G, a sends m to b, which then sends m to c.
- For KRP, on G', a' and c both know a local key k_f on their shared edge. On G, let b announce the sum of its incident local keys. Then a and c know the same local key k_f.

The node b has no other actions. The protocols proceed successfully as before. Now WLOG $\mathcal{E}'$ is maximal, i.e. the addition of an edge to any of its wiretap sets will result in loss of security. Define $\mathcal{E}$ to be $\mathcal{E}'$ with e appended to every wiretap set $E' \in \mathcal{E}'$

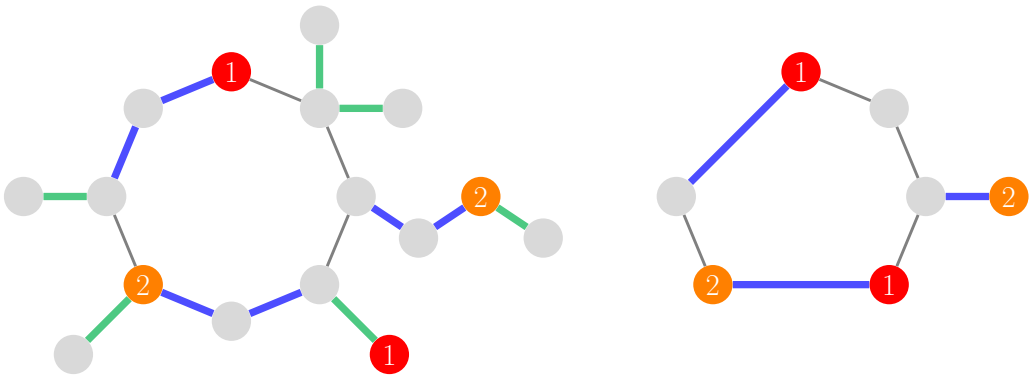

Figure 7: These two networks have the same security in SNC, KRP-by-SNC, and KRP. Leaf reductions and degree two reductions are indicated by green and blue edges, respectively

which has $f \in E'$. Since the local keys on e and f are totally correlated, this does not change the security. Since $\mathcal{E}'$ is maximal, we cannot append e to a wiretap set E' not containing f. This is because such a $(G, U, \mathcal{E})$ would be equivalent to $(G', U', \mathcal{E}')$ with the addition of f to E'. This modification accounts for every new wiretap set created by the addition of e. □

Leaf reduction and degree two reduction are demonstrated in Figure 7.

Lemma 3.13. (User-Pair Permutation Non-Equivalence)
Given a working KRP G with defined user pair locations u, a permutation of these user pair locations given by u' may not result in a KRP protocol that works.

Proof. This can be seen as a direct result of Theorem 3.5. We construct a motivating counterexample. Take a working KRP on a graph G with n user pairs and assume the minimum cut $\delta(S, \bar{S})$ which divides a_i into S and b_i into $\bar{S}$ is exactly of size n, that is, $m = n$.

Permute any two users a_j and b_k, so that a_j is moved from S to $\bar{S}$, and b_k vice versa. Then with respect to the original cut, (a_j, b_j) and (a_k, b_k) lie both on the same split of the graph. In some graph structures, the updated min cut $\delta(S', \bar{S}')$ may admit that $|\delta(S', \bar{S}')| < |\delta(S, \bar{S})| = m$ hence violating Theorem 3.5.

Hence when we permute the user pairs, the resulting minimum cut can be changed, such that $|\delta(S, \bar{S})| \neq |\delta(S', \bar{S}')|$. □

Theorem 3.14. (Min-Cut Bijection) Given n user pairs U on an undirected graph G along with wiretap set E, let the minimum cut separating the pairs contain n non-wiretapped edges. Define $K = \{k_i\}_{i=1\ldots n}$ and $L = \{l_j\}_{j=1\ldots n}$ as before. Then in any secure KRP protocol, K and L are in bijection.

Proof. We will provide an intuitive proof. For an information-theoretic proof of the same statement for KRP-by-SNC, see Lemma 11 in Ref. [6].

By constructibility, K can be determined from L. Define $f : \{0,1\}^n \to \{0,1\}^n$ as follows. Given $A \in \{0,1\}^n$, let $f(A)$ be the values of K determined from setting $L = A$. Without any wiretapping, the eavesdropper knows that the possible values for K lie in $\mathrm{Im}(f)$. If f is not surjective, then $|\mathrm{Im}(f)| < 2^n$, and the adversary has some information on K. Thus f must be surjective, and it defines the required bijection. □

4 Applications

The results presented above for KRP leads to a few results for simple graphs.

4.1 Tree Graphs

A **tree graph** is a graph with no cycles, equivalently a graph in which every node pair is connected by an unique path, or equivalently a connected acyclic undirected graph.

Theorem 4.1. (Tree Graph Equivalence)
On any tree graph, KRP and KRP-by-SNC are equivalent (see 2.15 for security definition). Equivalently, under any wiretap collection, there exists a successful KRP if and only if there exists a successful SNC. As such, a working SNC protocol can be constructed given a working KRP protocol on a tree graph, and vice versa.

KRP and KRP-by-SNC are both sound on a tree graph if and only if there are edge-disjoint paths connecting each of the user pairs.

Proof. A tree is defined as an undirected graph in which every distinct node pair is connected by an unique path. In graph theory terms, this is a connected acyclic graph. Consider n user pairs, of which the unique path between the sender and receiver in two user pairs, (a_1, b_1) and (a_2, b_2), utilize the same edge, e_s. Evidently, the edge cut includes e_s, whereas for each of the $n-2$ other user pairs, there exists an edge in $\delta(S, \bar{S})$ which uniquely separates a_i and b_i. Therefore, by definition we have that $m = n - 1$ and the edge cut condition in Theorem 3.5 is violated (see Lemma 3.9). This proves the equivalence in the negation case.

If the paths are edge-disjoint, then we simply appeal to Theorem 3.12 and both are sound by a protocol which involves sending a key down each path via secret channels (KRP-by-SNC) or simulation of secret channels (KRP). Then, the maximum wiretap set that KRP and KRP-by-SNC can tolerate includes every subset of edges that are not in one of the paths. Such edges do not affect the aforementioned protocol. Moreover, KRP and KRP-by-SNC are not resistant to any other wiretap set by the min-cut bound due to the unique path property of tree graphs.

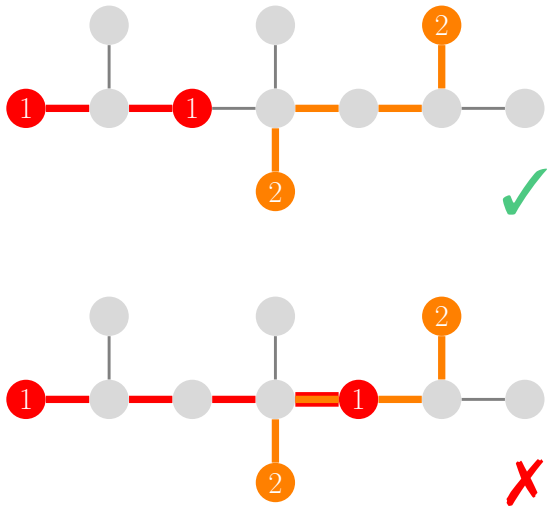

Figure 8: Examples of networks on tree graphs. The top graph is sound for both KRP and KRP-by-SNC, with edge disjoint paths labeled (and moreover, security on nontrivial wiretap sets is equivalent), while the bottom graph is not sound for both.

Example graphs are shown in Figure 8. □

4.2 Cycle Graphs

A **cycle graph** with n nodes is defined by $V = \{1, \ldots, n\}$ and $E = \{\{1,2\}, \{2,3\}, \ldots, \{n-1, n\}, \{n, 1\}\}$.

Theorem 4.2. KRP and KRP-by-SNC are equivalent on cycle graphs.

Proof. **One user pair**: There are two disjoint paths connecting the user pair. Both KRP and KRP-by-SNC can achieve a protocol which involves sending keys k_1 and k_2 down each path via secret channels (KRP-by-SNC) or simulation of secret channels (KRP). Then the resulting shared key is computed to be $k_1 \oplus k_2$. Then, the maximum wiretap set that KRP and KRP-by-SNC can tolerate involves only sets of edges that lie on a single path. Moreover, KRP and KRP-by-SNC are not resistant to any other wiretap set by the min-cut bound.

Two user pairs: Consider the order of user pairs as one traverses the ring counterclockwise. If the user pairs are adjacent, i.e. $a_1 < a_2 < b_1 < b_2$, then both KRP and KRP-by-SNC can achieve a protocol which involves sending keys down the path not containing the other user via secret channels (KRP-by-SNC) or simulation thereof (KRP). Then, the maximum wiretap set that KRP and KRP-by-SNC can tolerate involves only sets of edges that don't lie on such paths. Moreover, KRP and KRP-by-SNC are not resistant to any other wiretap set by Theorem 3.5.

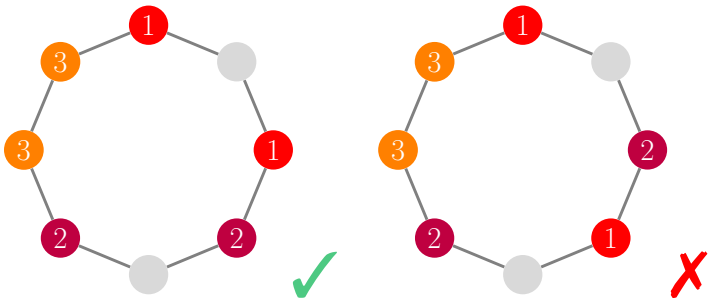

Figure 9: Examples of networks on cycle graphs. The graph on the left is sound for both KRP and KRP-by-SNC (and moreover, security on nontrivial wiretap sets is equivalent), while the graph on the right is not sound for both.

If user pairs are not adjacent, then we can apply Theorem 3.12 to simplify the graph to contain four nodes ordered as $a_1 < b_1 < a_2 < b_2$. KRP-by-SNC is not sound by a simple casework on the direction of the four edges. For KRP, we apply the linear algebra framework. The local key sources form a basis for $\mathbb{Z}_2^4$. Then following must be true for a valid KRP:

- None of the four local keys can lie in the span of the public information. Otherwise, the network is equivalent to a line graph obtained by cutting the leaked local key. This line graph is not sound by Theorem 4.1.
- The span of the public information must have dimension at most 2.

Then WLOG a_1 must make the first announcement, which is the sum of its adjacent local keys. The second announcement exhausts the dimensionality of public information, so there are only five more cases to consider. Up to rearrangement, b_1 has three possible nontrivial announcements, while a_2 has two possible nontrivial announcements. Each of these can be readily checked to not allow for the required key exchange.

Three or more user pairs: If every user pair is adjacent, then KRP-by-SNC and KRP are equivalent as before. If not every user pair is adjacent, then there exists at least three user pairs arranged in one of the following orders (up to rearrangement): $a_1 < b_1 < c_1 < a_2 < b_2 < c_2$, $a_1 < b_1 < c_1 < c_2 < a_2 < b_2$, or $a_1 < b_1 < c_1 < b_2 < a_2 < c_2$. In all of these cases there is a cut-set of size two separating the three user pairs, thus violating the min-cut bound. Then both KRP and KRP-by-SNC are not sound.

Example graphs are shown in Figure 9. □

4.3 Pseudotree Graphs

A **pseudotree graph** is a graph with exactly one cycle.

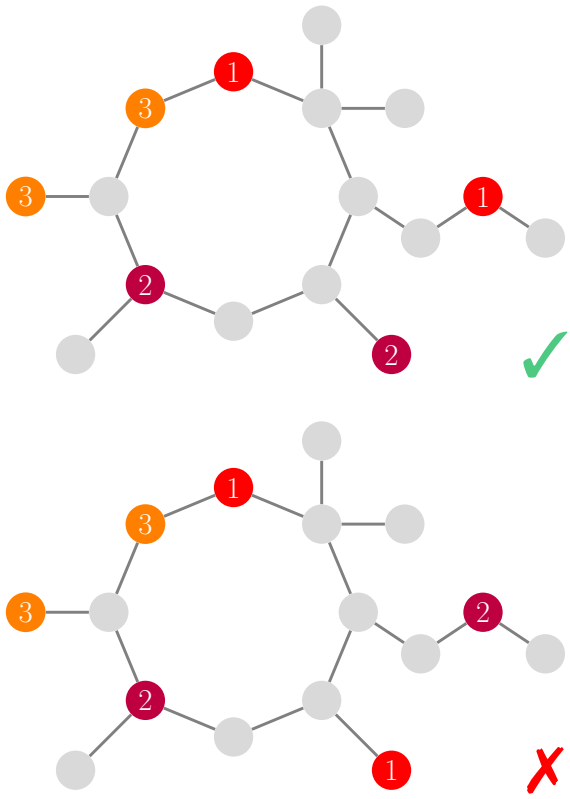

Figure 10: Examples of networks on pseudotree graphs. The top graph is sound for both KRP and KRP-by-SNC (and moreover, security on nontrivial wiretap sets is equivalent), while the bottom graph is not sound for both.

Theorem 4.3. KRP and KRP-by-SNC are equivalent on pseudotree graphs.

Proof. The proof is analogous to equivalence on tree and cycle graphs. Example graphs are shown in Figure 10 □

4.4 Complete Graphs

A complete graph is a graph in which every pair of distinct nodes is connected by an edge. A complete graph with n nodes is denoted by K_n.

Conjecture 2. On a complete graph with a single user pair, KRP and KRP-by-SNC are equivalent.

Consider the case where there is only one user pair on the complete graph K_n. In the absence of wiretap sets, by the definition of a complete graph there exists a direct edge connecting the two nodes of the user pair (a, b).
Furthermore, if the user pair is forced to employ intermediate nodes due to a wiretap

interference, the complete connectivity of the graph ensures that any intermediate node can relay the key to the receiving node under certain conditions. There are $n-1$ edge-disjoint redundant paths, therefore, the minimum cut size is $n-1$. It is hypothesized that there exists a choice of key and subsequent protocol construction for KRP-by-SNC and KRP that equivalently achieve or fail communication.

Remark. (One User Pair Equivalence)
Since any spanning graph can be embedded in a complete graph, and since the wiretap sets effectively determine which edges are secure, the conjecture for complete graphs implies a general equivalence between Key Relay Protocol (KRP) and Secure Network Coding (SNC) in the one-user-pair setting. In particular, by removing certain edges (e.g., those observed by the eavesdropper), any desired spanning graph structure can be induced within the complete graph. Therefore, the equivalence in the complete graph case suggests that KRP and SNC are also equivalent on arbitrary graphs, under appropriate wiretap conditions.

5 Feasibility

An interesting question becomes if we can determine the complexity class of an algorithm for determining whether a KRP is possible for a given network G and some user pairs (a_i, b_i). We think a similar technique of polynomial reduction from a similar secrecy capacity result for SNC Ref [4] would work, but have not made a concrete result for this.

Definition 5.1. (Clique Problem)
A clique is a subset of nodes C in a graph in which $\{(v_i, v_j) \forall v_i, v_j \in C, v_i \neq v_j\} \in E$. The problem of clique determination raises many related NP-hard problems. Specifically the one relevant for our discussion is the problem of finding whether a graph contains a k sized clique, which is a known NP-hard problem.

Cui et al's paper [4] proves that the secrecy capacity of a SNC network with unequal link capacities reduces to the clique problem, namely that if the graph G with an unknown wiretapping admits a clique of size k it can achieve a secrecy rate of k as well.

Theorem 5.2. (SNC Instance Implies a KRP Instance [6])
For every working SNC protocol in a graph G, a KRP can establish a equivalently working protocol. This is shown in [6].

It was initially our plan to use a polynomial reducibility argument to show that the KRP feasibility problem could be reduced to a known NP-hard problem, but this proved harder than initially thought. The technique applied in the SNC secrecy NP-hard proof cannot be directly adopted due to the non-equivalence with KRP in certain graphs. A more thorough treatment of the coding capabilities of KRP would aid in this analysis, touched on in Section 7.

6 Algorithm Verification

To verify the functionality and security guarantees of KRP, we developed a Python-based implementation that takes as input a graph $G = (V, E)$, a designated set of user pairs, and a set of adversarial wiretaps from $\mathcal{E} \subseteq 2^{2^E}$. The purpose of this algorithm, fully detailed in Appendix 0, is to systematically determine whether secure key distribution is possible under the KRP in a given network topology. The algorithm performs multiple layered checks such as, it first ensures the structural integrity of the graph, for instance, the connectivity and min-cut constraints, then simulates local key generation at edges, followed by public key announcements at nodes. Security is evaluated by checking for linear independence among the derived keys and public information to determine if an adversary can deduce any key. The final step ensures both users in each pair can reconstruct the shared key (K_i) for the user pair (a_i, b_i) using only their local keys and the public announcements.

This implementation builds on the incidence vector representation and linear algebraic formulation of the KRP, as defined in Section 3.1. Each primitive secret bit generated on an edge is represented as a canonical basis vector in the vector space $\mathbb{Z}_2^{|E|}$. The adversary's information space is captured by the subspace $A_{E_w} = \text{span}\{[v_{e_i} \mid e_i \in E_w] \cup P\}$, where P denotes the set of all public announcements. A key is considered insecure if its vector lies within this subspace. Thus, the algorithm verifies security by checking whether the rank condition $\text{rank}(A_{E_w} \cup v_k) > \text{rank}(A_{E_w})$ holds. By automating these procedures, the algorithm enables both mathematical validation and experimental exploration, particularly to identify counterexamples where KRP ensures security but traditional SNC does not, highlighting further non-equivalence between the two protocols on some specific graphs.

7 Further Research

We explored many directions throughout the lifecycle of this project and there remain several incomplete directions and promising avenues for future work. Below are some which we would think are the most promising.

Based on Section 3, we were able to make a couple useful results that have similar precedents in other studies of communication network paradigms. To get a concrete result and proof for Conjecture 1 and Question 3.1.2 would have immediate downstream results in applying these feasibility conditions for generalized graphs.

At our current stage, the theorems in 4 mostly reduce to a routing feasibility problem due to the structure of the simple graphs we consider. In more complicated graphs, for instance the butterfly network, we are able to leverage network coding to allow for more complicated user pair communication patterns. This however greatly complicates the feasibility and equivalence study, as we can no longer simply appeal to the existence

of *unique* edge-disjoint paths. In short, although the results in Section 3 hold for the KRP paradigm in general, they are not enough to strongly establish the capabilities of KRP, especially when we utilize network coding capabilities.

Also, the characterization of the KRP paradigm's capabilities on certain graph properties may lead to interesting results. Specifically, we think further study along these lines would require arguments on what the *best* choices of keys and public announcements are to enable communication. A designer can of course, select a key that is easily intercepted, but we can make arguments to what the natural choice of key and public announcement would be. These could possibly be used to argue that KRP and SNC can or cannot achieve the same capabilities on certain graphs, such as planar ones.

In Section 5, we explored avenues for studying whether an algorithm for determining the feasibility of KRP was NP-hard. The similar results for SNC provide a good starting point for that study, but are not exhaustive due to the previously established gap between KRP and SNC in some cases Ref [6].

Contributions

Ed Chen
Ed was the student project manager. He worked on proving the KRP mathematical formulation and the information theoretic minimum cut result, and worked towards the generalizations on simple graphs. He defined the exploratory feasibility and planarity research directions outlined, and wrote significant report portions.

Maho Maruyama
Maho's main contribution was in organizing and visualizing the theoretical background of the project. She closely studied the literature on network coding, gained an understanding of the fundamental theories behind key-sharing protocols such as KRP and SNC, and created slide materials and diagrams.

Derek Zhang
Derek worked on the equivalence and non-equivalence of protocols, including graph generalization tools, equivalence on simple graphs, and producing counterexamples. He proposed the KRP mathematical formulation and code implementations, and worked on presentation and report organization.

Donald Zvada
Donald's input was more on the information theory component of the project, reports writing and implementation of a python script to work as a verifier for checking if given a graph, is it possible for KRP to communicate having checked preliminaries in the background section.

Acknowledgements

This research project was made possible by several organizations and individuals. We would like to thank Tohoku University, the Advanced Institute for Materials Research (AIMR), the Mathematical Science Center for Co-creative Society (MathCCS), the Tohoku Forum for Creativity (TFC), and the Institute for Pure and Applied Mathematics (IPAM), which is supported by NSF grant DMS-1925919. We would also like to thank our mentors for their constant guidance, and Dr. Hiroshi Suito for organizing the G-RIPS Sendai 2025 program. Finally, we want to thank all of our fellow G-RIPS students, who made this program productive and enjoyable.

References

[1] Bennett, C. H. and Brassard, G. [2014], 'Quantum cryptography: Public key distribution and coin tossing', *Theoretical computer science* **560**, 7–11.

[2] Cai, N. and Yeung, R. W. [2002], Secure network coding, *in* 'Proceedings of the IEEE International Symposium on Information Theory', Lausanne, Switzerland, p. 323.

[3] Cai, N. and Yeung, R. W. [2010], 'Secure network coding on a wiretap network', *IEEE Transactions on Information Theory* **57**(1), 424–435.

[4] Cui, T., Ho, T. and Kliewer, J. [2010], On secure network coding with unequal link capacities and restricted wiretapping sets, *in* '2010 IEEE Information Theory Workshop', pp. 1–5.

[5] Gisin, N., Ribordy, G., Tittel, W. and Zbinden, H. [2002], 'Quantum cryptography', *Reviews of Modern Physics* **74**, 145.

[6] Kato, G., Fujiwara, M. and Tsurumaru, T. [2023], 'Advantage of the key relay protocol over secure network coding', *IEEE Transactions on Quantum Engineering* **4**, 1–17.

[7] Katz, J. and Lindell, Y. [2014], *Introduction to Modern Cryptography*, second edn, Chapman & Hall/CRC.

[8] Kent, A. [1999], 'Unconditionally secure bit commitment', *Physical Review Letters* **83**, 1447.

[9] Nielsen, M. A. and Chuang, I. L. [2010], *Quantum Computation and Quantum Information: 10th Anniversary Edition*, Cambridge University Press, Cambridge.

[10] Peev, M., Pacher, C., Alléaume, R., Barreiro, C., Bouda, J., Boxleitner, W., Debuisschert, T., Diamanti, E., Dianati, M., Dynes, J. F., Fasel, S., Fossier, S., Fürst, M., Gautier, J.-D., Gay, O., Gisin, N., Grangier, P., Happe, A., Hasani, Y., Hentschel, M., Hübel, H., Humer, G., Länger, T., Legré, M., Lieger, R., Lodewyck, J., Lorünser, T., Lütkenhaus, N., Marhold, A., Matyus, T., Maurhart, O., Monat, L., Nauerth, S., Page, J.-B., Poppe, A., Querasser, E., Ribordy, G., Robyr, S., Salvail, L., Sharpe, A. W., Shields, A. J., Stucki, D., Suda, M., Tamas, C., Themel, T., Thew, R. T., Thoma, Y., Treiber, A., Trinkler, P., Tualle-Brouri, R., Vannel, F., Walenta, N., Weier, H., Weinfurter, H., Wimberger, I., Yuan, Z. L., Zbinden, H. and Zeilinger, A. [2009], 'The secoqc quantum key distribution network in vienna', *New Journal of Physics* .
URL: *https://dx.doi.org/10.1088/1367-2630/11/7/075001*

[11] Salvail, L., Peev, M., Diamanti, E., Alléaume, R., Lütkenhaus, N. and Länger, T. [2010], 'Security of trusted repeater quantum key distribution networks', *Journal*

of Computer Security **18**(1), 61–87. Preprint available at https://arxiv.org/abs/0904.4072.

[12] Sasaki, M., Fujiwara, M., Ishizuka, H., Klaus, W., Wakui, K., Takeoka, M., Miki, S., Yamashita, T., Wang, Z., Tanaka, A., Yoshino, K., Nambu, Y., Takahashi, S., Tajima, A., Tomita, A., Domeki, T., Hasegawa, T., Sakai, Y., Kobayashi, H., Asai, T., Shimizu, K., Tokura, T., Tsurumaru, T., Matsui, M., Honjo, T., Tamaki, K., Takesue, H., Tokura, Y., Dynes, J. F., Dixon, A. R., Sharpe, A. W., Yuan, Z. L., Shields, A. J., Uchikoga, S., Legré, M., Robyr, S., Trinkler, P., Monat, L., Page, J.-B., Ribordy, G., Poppe, A., Allacher, A., Maurhart, O., Länger, T., Peev, M. and Zeilinger, A. [2011], 'Field test of quantum key distribution in the tokyo qkd network', *Optics Express* **19**(11), 10387.
URL: *http://dx.doi.org/10.1364/OE.19.010387*

[13] Stacey, W., Annabestani, R., Ma, X. and Lütkenhaus, N. [2015], 'Security of quantum key distribution using a simplified trusted relay', *Physical Review A* **91**(1).
URL: *http://dx.doi.org/10.1103/PhysRevA.91.012338*

[14] Wikipedia [2025], 'Linear network coding — Wikipedia, the free encyclopedia', http://en.wikipedia.org/w/index.php?title=Linear%20network%20coding&oldid=1297072487. [Online; accessed 26-June-2025].

[15] Xu, F., Ma, X., Zhang, Q., Lo, H.-K. and Pan, J.-W. [2020], 'Secure quantum key distribution with realistic devices', *Reviews of Modern Physics* **92**, 025002.

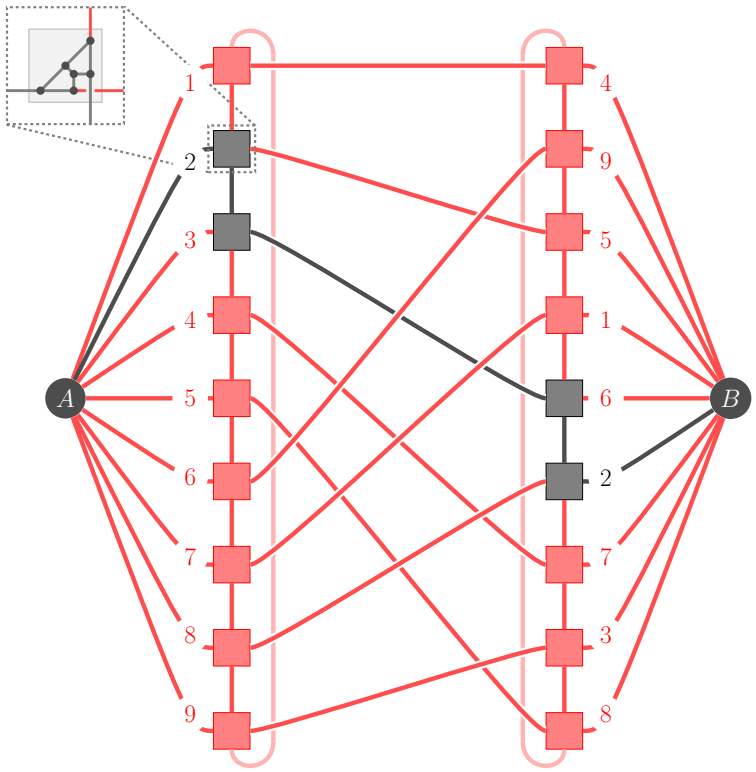

Figure 11: Conjectured counterexample network between KRP-by-SNC and KRP for one user pair, with one wiretap set shown.

A Appendix

We pursued counterexample networks along the lines of Theorem 2 of [6].

The network in Figure 11 is conjectured to be a counterexample for one user pair, which obtains security on KRP but not KRP-by-SNC. The wiretap collection consists of 9 wiretap sets: $\mathcal{E} = \{E_1, \ldots, E_9\}$, of which E_2 is depicted. The network is based on graph G_0 in [6]: user pairs have been condensed into two users. In G_0, there are nine separate information flows corresponding to the user pairs. It was proven that these flows must follow a standard path, corresponding to the unwiretapped edges in Fig. 11. The wiretap sets are chosen in an attempt to force information to flow along 9 paths, in much the same way as in G_0.

The network in Figure 12 is a counterexample for three user pairs, which obtains security on SNC-S but not SNC. The wiretap collection is empty. We set the senders to be A_i, and the receivers to be B_i. It can be readily seen that communication is not possible if all the senders are on one side. Allowing both members of any pair to be the sender, as in SNC-S, results in successful communication. Additionally, we believe

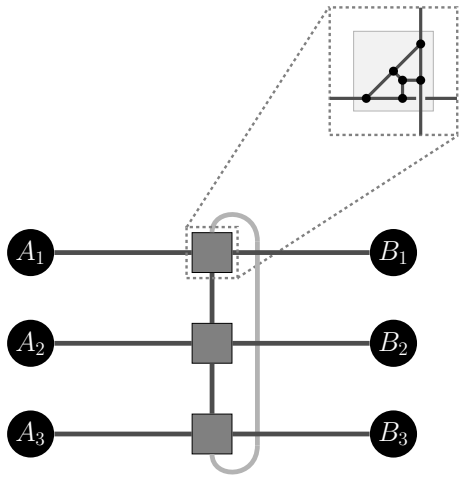

Figure 12: Counterexample network between SNC and SNC-S for three user pairs.

that a more interesting counterexample involving KRP-by-SNC can be constructed by combining multiple copies of this network.

Algorithm 1 KRP Algorithm and Code Repository

Input: Graph $G = (V, E)$
User specifies:

- Number of edges $|E|$
- Set of nodes V
- Set of user pairs $\{(a_i, b_i)\}$
- Wiretap sets $\mathcal{E}$ such that $E_w \in \mathcal{E}$

Step 1: Graph Validation
if Graph G is not connected **then**
    return "Graph is disconnected. Stop!."
end if
if Min-cut condition is not satisfied **then**
    return "Min-cut condition violated. Stop!."
end if

Step 2: Local Key Generation and Public Information
for all edges $e_i \in E$ **do**
    Generate local key r_{e_i}
    Distribute r_{e_i} to incident nodes
end for
for all nodes $v \in V$ **do**
    Generate **public keys** as some **linear combination** of some random bits from the set of edges.
end for

Step 3: Security Checks
Check linear independence of keys and public information
if Keys are not linearly independent **then**
    return "Keys not secure. Stop!."
end if

Step 4: User Key Delivery Verification
for all user pairs (a_i, b_i) **do**
    Verify both a_i and b_i can recover K_i
end for

Step 5: Visualization
Plot the graph G with user pairs and wiretap sets highlighted