Security Protocols with Isotropic Channels

Madhukar Anand, Eric Cronin, Micah Sherr, Matt Blaze, and Sampath Kannan
Department of Computer and Information Science
University of Pennsylvania
{anandm,ecronin,msherr,blaze,kannan}@cis.upenn.edu

Abstract

We investigate the security properties of *isotropic channels*, broadcast media in which a receiver cannot reliably determine whether a message originated from any particular sender and a sender cannot reliably direct a message away from any particular receiver. We show that perfect isotropism implies perfect (information-theoretic) secrecy, and that asymptotically close to perfect secrecy can be achieved on any channel that provides some (bounded) uncertainty as to sender identity. We give isotropic security protocols under both passive and active adversary models, and discuss the practicality of realizing isotropic channels over various media.

Keywords: Unconditional security, secret-key agreement, provable security, isotropic channels

1 Introduction

Secure communication in the presence of an adversary is the fundamental problem of cryptology, for which solutions generally involve compromises or constraints of one kind or another. When the communication channel provides sufficiently generous access to the attacker, long keys must be shared in advance (as in a one-time pad) or computational techniques (e.g., Diffie-Hellman [14]) must be employed. Unfortunately, pre-sharing arbitrarily long keys is often impractical, and computational cryptography (for all its practical merit) makes assumptions about the computational capabilities of an eavesdropper and depends on (perhaps incorrect) bounds on the difficulty of solving some underlying problem. Another approach to secure communication is the use of channels that provide *intrinsic* security properties. Quantum communication [2, 5], for example, exploits physical uncertainty in the communication medium itself to prevent undetected third party eavesdropping. Quantum channels, unfortunately, are not yet practical under every configuration and application for which security is required.

In this paper, we examine channels that behave approximately like a broadcast medium, in which eavesdroppers can receive (and possibly transmit), but with the constraint that receivers cannot reliably determine whether a given message originated from any particular sender and senders cannot prevent a message from reaching any particular receiver. We refer to such "directionless" media as *isotropic channels*[1].

We show that a perfectly isotropic channel can yield unconditional confidentiality, in which a passive eavesdropper learns nothing about message content and an active eavesdropper will always be detected with compromised bits discarded before confidential traffic is sent.

Perhaps more interestingly, protocols can still asymptotically approach perfect secrecy even if the channel is only *partially isotropic* (i.e., an attacker can identify the source of a message, but only probabilistically). As long as the ability to source messages can be bounded, the protocols described in this paper are sufficient for achieving unconditionally secure message exchange, even in the presence of active adversaries.

This paper takes first steps toward a theory of isotropic cryptography, comprised of the following contributions: (1) a formalization of isotropic channels; (2) an attacker model for isotropic channels; and (3)

[1] Our use of the term "isotropic" is inspired by, but not identical to, the analogous concept in physics and communications theory.

protocols for achieving reliable communication and information-theoretic confidentiality in isotropic channels.

As well as being an interesting theoretical formulation in its own right, isotropic cryptography may be of practical significance in certain applications. The protocols we introduce are sufficiently general to achieve asymptotically close to perfect secrecy in any abstract (partially) isotropic channel and are not dependent on any specific communication medium (e.g., copper, radio, optical, quantum channels, etc.). Isotropic cryptography may provide an attractive alternative to computational or quantum cryptography in applications where a degree of directional ambiguity can be assured. We informally discuss the practical realizability of isotropic channels in the context of various communications media at the end of this paper. However, we note here that many emerging physical- and network- layer technologies appear to be amenable to isotropism.

2 Related Work

We are not the first to observe that channels lacking sender authentication are potentially useful for non-computational cryptography. Alpern and Schneider's 1983 paper [1] is the first mention of such channels we are aware of in the literature, and the idea has resurfaced occasionally since then. Recently, the topic has been revisited in the context of emerging communications technologies, namely RFID and wireless sensor networks [7, 8] and confusion [11]. Although earlier works suggested that confidentiality could intuitively be obtained with a strictly passive adversary, none contain a proof of security or formally model the degradable channel properties on which security depends. In this paper, on the other hand, we provide a formal definition of isotropic channels and show how provable security can be achieved. In addition, we introduce and model the concept of degraded isotropic channels, and show that as long as observable directionality is bounded, confidentiality that asymptotically approaches perfect secrecy can still be obtained, even when the eavesdropper is active. The specific examples proposed in the previous literature thus represent a subset of the situations covered by our model, and we discuss them further when examining the realizability of isotropic channels.

Information-theoretic key exchange has been a topic of research for many years. In addition to more recent quantum techniques [2, 5], there have been a number of techniques relying only on classical communication. Unfortunately, these prior techniques often rely on primitives that are difficult (or exceedingly expensive) to realize in practice, and none has succeeded as a viable alternative to standard techniques.

A number of classical key exchange protocols [13, 21, 15, 6, 10, 18] have been proposed that leverage a small amount of shared (or merely correlated) information to produce large keys which an eavesdropper has negligible information about. A typical example of such schemes uses a shared random source broadcasting over noisy channels (e.g. a low-power satellite). Using the shared bits not known to the eavesdropper, these schemes amplify the secret to arbitrarily large keys. Unlike these schemes, we require no initial shared secrets to establish unconditional security in the presence of both active and passive eavesdroppers. Our technique utilizes active participation by the recipient in order to achieve this result.

Independently from our work, Ishai et al. have explored cryptography from anonymity [19]. However, they focus on leveraging anonymity for private information retrieval and assume the existence of anonymous channels, similar to isotropic channels. In this work, we develop the theory of isotropic channels, show the feasibility of secure key exchange, and present protocols to realize security in isotropic channels with practical considerations such as losses and collisions.

3 Isotropic Channels

Typical communication channels often have some notion of *directionality*. Any party (including an eavesdropper) can identify the sender of a received message. Directionality can be the result of physical properties of the medium (e.g., voltage signatures on a wire, point-to-point radio links, etc.) or of logical requirements for message delivery (e.g., the source and destination fields in IP packets, authenticated headers in IPSec datagrams, etc.). Although this directionality can sometimes be obfuscated or confused [12, 11], such measures are rarely taken due to their negative impact on reliability.

However, it may be possible (and, as we show, useful) to design and implement communication networks in which messages convey little or no information concerning their true senders. These channels can be established using some physical property of the communication medium (e.g., the difficulty of locating the source of a wireless transmission when parties are mobile or move their transmitters [8]), or they may be constructed using logical overlay networks such as anonymity networks [9]. A more detailed discussion of the realizability of these isotropic channels is presented in Section 7.

Isotropic channels are useful for security since they give honest parties an inherent advantage over even the most perceptive eavesdropper. An honest communicating party knows which messages it sent by virtue of having sent them. The eavesdropper, on the other hand, does not know the sender of an intercepted message and must hypothesize as to its true originator. As we show, this asymmetry between the participating parties and the eavesdropper can be used to convey secret bits.

In this paper, we consider isotropic channels in which there are three principals: Alice and Bob, who are honest participants without any *a priori* shared secrets, and Eve, a (potentially active) eavesdropper. If, in reality, there are multiple eavesdroppers, we assume that they are colluding and can combine their knowledge and capabilities, and we model them collectively as Eve. Although we do not consider multiple senders and receivers sharing the same isotropic channel, such a system can easily be incorporated by prefixing each message with a conversation identifier. In this work, we are concerned only with two-party key agreement and do not consider one-to-many secret sharing. Definitions below and in the rest of the paper assume this scenario.

Formally, an isotropic channel is defined as follows:

Definition 1. *(ρ-bounded Isotropic Channel) A communications channel is a ρ-*bounded isotropic channel *if all messages are broadcast to all parties, an honest party cannot discern the sender of a message not from itself (although it may reason about a message's origins) and the probability that an eavesdropper E can correctly identify the sender of a message not originating from E is at most ρ, where $\frac{1}{2} \leq \rho < 1$.*

We assume that ρ represents the maximum probability that Eve learns the identity of the sender, taking into consideration possibilities such as multiple points of eavesdropping. Since we model two honest parties, $\rho \geq \frac{1}{2}$. Furthermore, ρ is a constant probability and does not vary over time.

A special case of ρ-bounded isotropic channels is the *perfectly isotropic channel*:

Definition 2. *(Perfectly Isotropic Channel) A communications channel is a* perfectly isotropic channel *if it is a ρ-bounded isotropic channel and $\rho = \frac{1}{2}$.*

In the following subsections, we further describe our assumptions of isotropic channels and delineate the capabilities of an adversary.

3.1 Abstract Channel Assumptions and Characteristics

Instances of isotropic channels may differ vastly in their implementations and may demonstrate properties not shared by other isotropic channels. We have therefore attempted to be as general as possible in our definition. The channel assumptions below are intended to represent an intuitively "minimal" set of abstract requirements for isotropism, and are sufficient for achieving the security guarantees that we derive in later sections of this paper.

- An eavesdropper (active or passive) can infer the true sender of an intercepted message with probability $\frac{1}{2} \leq \rho < 1$.
- Since messages communicated via isotropic channels contain no source or destination information, relaying of messages is not possible. Consequently, isotropic channels must employ broadcast communication and not communication directed to a particular party.

- Messages take some time to be delivered, and therefore may collide.
- A party X who is not the originator of the message receives the message with probability $1 - \pi$, $0 \leq \pi < 1$, where π is a *constant loss probability*. We take π to be a per-message loss and not per-bit loss. This can be easily achieved by enforcing checks such as CRCs[2] for every message and therefore, even if a few bits are lost, the whole message is discarded. Furthermore, we assume that all honest parties experience the same loss rate. (Asymmetric loss rates may help identify the sender of a message, and can therefore be modeled by using a larger ρ or by having the party with the lower loss rate probabilistically drop received message so that the effective loss rates are equal.)
- All parties have synchronized clocks.
- All parties have a source of pure randomness.

3.2 Attacker Capabilities

We consider both passive and active adversaries. (An adversary is also referred to as an eavesdropper, or as Eve.) Passive adversaries monitor the network but never transmit. Active adversaries monitor the network and can transmit, but do not have to obey any network protocols. In both cases, we assume an adversary, E, who can reliably intercept all messages (i.e., $\pi_E = 0$).

We view either the passive or active adversary as successful when she can learn at least one bit of the exchanged key. An active adversary is also successful when it can either influence the exchanged key or force Alice and Bob to agree on different keys. An active adversary has the following capabilities:

- She can hear all messages ($\pi_E = 0$) and identify the sender of a message with probability $\rho < 1$.
- She can insert messages into the network. Since all messages are broadcast, she cannot direct her message to a particular recipient. Eve's messages are heard by all parties, except when an honest party experiences loss.
- She may block any message. *Blocking* is distinct from *deletion*. Since messages are broadcast, the adversary must induce a collision in the network in order to block a message. We make several (perhaps overly) conservative assumptions in favor of the adversary. First, we assume that Eve's attempts to block messages are always successful. We further assume that no honest party, including the originator of the message, can discern whether a message is blocked. Only the originator of the message and Eve are aware of the existence of the blocked message. Finally, we assume that Eve can learn the contents of blocked messages.
- She may modify any message, but may do so only by blocking a message and transmitting a modified version. She cannot modify messages in real-time, since a broadcast medium is used and the channel is unusable while active blocking occurs.

Denial-of-service (DoS) attacks are trivially achieved, and we do not consider them in this paper.

4 Collision Avoidance and Sender Selection (CASS) Protocol

The key agreement protocols introduced later in this paper rely on isotropism for secure key agreement. Intuitively, in the case in which Eve is passive, Alice knows which messages she transmitted and she knows that all received messages originated from Bob. Since both Alice and Bob can identify the source of messages, they can agree upon a coding scheme – e.g., use `0` as the next bit of the key if the sender is Alice and `1` if it is Bob. The channel's isotropism bounds Eve's ability to identify the source of a transmission to a function of ρ, and as we discuss in Section 5, we can utilize simple privacy amplification techniques to achieve

[2] We do not depend on CRCs or any other error detection code to derive security guarantees. CRCs are used only to detect transmission errors. This is independent of an active adversary's ability to inject messages (either with valid or invalid CRCs).

confidentiality that asymptotically approaches perfect secrecy. Protocols for achieving key agreement in the presence of active adversaries derive similar confidentiality guarantees and are discussed in Section 6.

A requisite for isotropism-based key agreement is that Alice and Bob must have equal probability of being the sender each time a bit is shared in the key agreement protocol, else the key will not be random. Since Alice and Bob cannot agree on sender order prior to key agreement (otherwise, the sender order itself constitutes a key), there is also a need to prevent both parties from simultaneously sending (i.e., colliding). The Collision Avoidance and Sender Selection (CASS) protocol is a media access control (MAC) layer protocol that achieves three goals. CASS guarantees that at most one sender is elected, sender anonymity is preserved, and that both parties have equal probability of being elected.

CASS itself is not used to securely exchange key material. Rather, it is used to elect a sender and reserve the channel. If a winner is chosen, the channel is reserved for the winner and he or she may transmit data without fear of collision (assuming, of course, Eve does not inject packets). If no winner is elected, another instance of CASS must be carried out before any party can send a message.

We assume that Alice and Bob have agreed upon some time interval $[t_s, t_e]$ during which CASS messages are sent. In each instance of the protocol, Alice and Bob independently flip a fair coin to determine their desire to transmit. A party sends a `1` at time t picked uniformly at random from $[t_s, t_e]$ if it does *not* want to be selected as the sender. The desire to be elected is indicated by not transmitting during the interval.

At the end of the time interval, a party transmits iff it sent no messages during the CASS instance (indicating it wanted to win the election) and it received a `1` from the other party (indicating the other party did not want to transmit). In all other cases, no party is elected as the sender, and all parties must wait until the next instance of CASS. Note that if one or more messages are lost or if Alice and Bob both transmit and their messages collide, no sender will be selected.

Lemma 1. *The* Collision Avoidance and Sender Selection (CASS) *protocol guarantees that, in the absence of an adversary who can inject messages, at most one sender will be chosen and if there are no losses, then there is a consensus at both ends about the sender.*
Proof. A proof is provided in the Appendix.

Lemma 2. *The* Collision Avoidance and Sender Selection (CASS) *protocol guarantees that, in the absence of an adversary who can inject messages, Alice and Bob have equal probability of being selected.*

Proof. The proof follows from the definition and symmetry (with respect to Alice and Bob) of the protocol. □

Lemma 3. *In the absence of an adversary who can inject messages, the expected number of iterations of CASS required to elect a sender using the* Collision Avoidance and Sender Selection (CASS) *protocol is* $2/(1 - \pi)$.
Proof. A proof is provided in the Appendix.

5 Secure Key Agreement in the Presence of Passive Eavesdroppers

We now consider the problem of securely exchanging keys in an isotropic channel. We exploit the asymmetry in knowledge between Eve and the legitimate parties – while Alice knows definitely whether or not she broadcast a given message, Eve must make a guess as to the originator of that message. In this section, we describe a protocol for achieving information-theoretically secure key agreement under the assumption that all eavesdroppers are passive. The protocol is therefore appropriate when active eavesdroppers need not be considered (for example, when active eavesdropping is immediately detectable by all parties). In Section 6, we introduce a slightly less efficient key agreement protocol that offers protection against active adversaries.

5.1 Perfectly Isotropic Channels

We now introduce the Secure Key Agreement in Perfectly Isotropic Channels (SKA) Protocol for achieving perfect secrecy in perfectly isotropic channels.

SKA begins when one party (Alice) informs the other party (Bob) via broadcast that it wishes to communicate a secret. Since we assume only passive eavesdroppers, we note that Eve cannot masquerade as either Alice or Bob. Likewise, Eve cannot influence sender selection.

At the start of the protocol, Alice and Bob each have a counter (c_A and c_B, respectively) initialized to 0. We assume that the parties agree on timeouts for each stage of the protocol and that they will share an n-bit key. SKA proceeds as iterations of steps in which one bit of the key is securely shared. Each iteration is as follows:

1. Alice and Bob elect a sender via CASS. If no party is elected, then neither party performs any action during the remaining steps (i.e., they wait patiently for the next iteration), and the protocol repeats from this step. Otherwise, from Lemma 1, we have at most one sender. We denote the sender as S and the receiver as R, and likewise the counter belonging to S as c_S and the counter belonging to R as c_R.
2. S broadcasts the message $\{c_S\}$.
3. If R does not receive the message $\{c_S\}$, R performs no action, causing S to timeout waiting for an acknowledgment from R (see step 4). If R does receive the message, there are three cases to consider:
   (a) $c_R = c_S$**:** R records the next bit of the key as follows: If R is Alice (meaning the sender is Bob), the next bit in the key is `1`. Otherwise, if R is Bob and the sender is Alice, the next bit in the key is `0`. Additionally, R increments its counter ($c_R \leftarrow c_R + 1$) and broadcasts the acknowledgment message $\{\texttt{ACK}\}$.
   (b) $c_R = c_S + 1$**:** This case occurs when R previously received a message from S and its acknowledgment was not received by S. R deletes the previously recorded bit of key material and sets $c_R \leftarrow c_S$. R does not send an acknowledgment, causing S to timeout.
   (c) $c_R = c_S - 1$**:** R performs no action, causing S to timeout.

   (As proved in Theorem 4, $|c_S - c_R| \leq 1$ at all times.)
4. If an $\{\texttt{ACK}\}$ is received by S, S records the next bit of the key. As in case (a) above, S records a `0` if S is Alice, otherwise S records a `1`. Additionally, S sets $c_S \leftarrow c_S + 1$. Otherwise, if an $\{\texttt{ACK}\}$ is not received, S does not record a bit and the iteration is wasted.

The protocol continues from the first step until one party receives an $\{\texttt{ACK}\}$ for the $(n + 1)$th bit. Upon receiving an $\{\texttt{ACK}\}$ for the $(n+1)$th bit, that party repeatedly broadcasts a $\{\texttt{Done}\}$ message until it receives a $\{\texttt{DoneACK}\}$ acknowledgment message from the other party. The party that transmitted the $\{\texttt{Done}\}$ message considers the key to be successfully exchanged after it has received the $\{\texttt{DoneACK}\}$ response.

If at any point in the protocol either Alice or Bob receives a $\{\texttt{Done}\}$ message, s/he immediately considers the key to be successfully shared and responds with a $\{\texttt{DoneACK}\}$ message. A $\{\texttt{DoneACK}\}$ message should be transmitted for each received $\{\texttt{Done}\}$ message (multiple $\{\texttt{Done}\}$ messages are received if previous $\{\texttt{DoneACK}\}$ messages are lost).

Alice and Bob consider only the first n bits as the key.[3]

Theorem 4. (SKA Message Consistency) *In the absence of an active adversary, SKA ensures that both parties agree on the same sequence of n bits at the conclusion of the protocol.*
Proof. A proof is provided in the Appendix.

[3] Note that if the channel is not perfectly isotropic (i.e., $\rho > \frac{1}{2}$), the security of the $(n + 1)$th bit is reduced since at least one $\{\texttt{Done}\}$ message is sent by S and these messages can be correlated with messages sent in steps 1 through 3 to improve Eve's ability to identify S. For this reason, the $(n + 1)$th bit is discarded. Recall that each iteration of the protocol is independent, and the leakage of the discarded $(n + 1)$th bit does not reduce the security of the other shared bits.

Lemma 5. (Progress) *In the absence of an active adversary, if in an iteration of SKA c_A equals c_B, progress will eventually be made (i.e., a bit is shared and again, $c_A = c_B$). The expected number of iterations required to exchange one bit of an n-bit secret is $\frac{3+5\pi-2\pi^2}{(1-\pi)^2}$.*
Proof. A proof is provided in the Appendix.

Lemma 5 implies that SKA will eventually terminate, since initially $c_A = c_B$ $(= 0)$, and after each bit is exchanged $c_A = c_B$ once again.

Theorem 6. (Perfect Secrecy) *In the absence of an active adversary, SKA achieves* perfect secrecy *in a perfectly isotropic channel and each bit in the shared key has equal probability of being 0 or 1.*
Proof. A proof is provided in the Appendix.

5.2 ρ-Bounded Isotropic Channels

If $\rho > \frac{1}{2}$, Eve can determine whether Alice or Bob is the sender in an iteration of the protocol with probability greater than ρ (here, we denote the sender as the party that sends $\{c_S\}$). When a bit is exchanged (i.e., the iteration is not discarded), Eve knows that the party who is not the sender sent a $\{\mathtt{1}\}$ in step 1 (CASS) and an $\{\mathtt{ACK}\}$ in step 3. Since SKA forces a particular ordering of transmissions (although the roles of Alice and Bob are unknown), Eve can correlate her beliefs to better identify the sender in a given iteration.

We define ρ' to be the probability that Eve can identify the sender during an iteration of SKA.

Lemma 7. *In the absence of an active adversary, the probability that Eve can identify the sender during a non-discarded iteration of SKA is ρ', where $\rho' = \rho^2(3 - 2\rho)$.*
Proof. A proof is provided in the Appendix.

(Note that in the special case of perfect isotropism, ρ' also equals $\frac{1}{2}$.)

Alice and Bob can achieve confidentiality that is asymptotically close to perfect secrecy through *privacy amplification* [3, 4]. Below, we describe a simple privacy amplification technique. It is possible to greatly improve the communication efficiency of our scheme using other privacy amplification techniques [3]. However, at this point we are not concerned about optimizing for efficiency since our goal is to provide a proof-of-concept protocol that demonstrates the feasibility of achieving information-theoretic key exchange in isotropic channels.

When $\rho > \frac{1}{2}$, we compensate for Eve's ability to probabilistically identify the source of a message by sharing additional bits. These extra bits are then condensed into n bits of usable key. By sending a linear number of extra bits, Alice and Bob can exponentially increase the confidentiality of their key, asymptotically approaching perfect secrecy. We call our slightly modified technique the Secure Key Agreement in ρ-Bounded Isotropic Channels (SKA$_\rho$) Protocol.

SKA$_\rho$ is nearly identical to SKA. However, rather than communicating n bits to share an n-bit key, $s \cdot n$ bits are shared. The $s \cdot n$ bits are then split into n strings of s bits. Each s bit string is compressed into one bit by taking the exclusive-or (xor) of all bits in the string. The resulting n bits are then used by Alice and Bob as the shared key.

Lemma 8. *In the absence of an active adversary, each bit in a key shared using SKA$_\rho$ has equal probability of being a 0 or a 1.*
Proof. A proof is provided in the Appendix.

Theorem 9. *In the absence of an active adversary, SKA$_\rho$ achieves confidentiality that asymptotically approaches perfect secrecy in a ρ-bounded isotropic channel. The probability that an eavesdropper identifies a bit of the key correctly is $\frac{1+(2\rho'-1)^s}{2}$.*
Proof. A proof is provided in the Appendix.

Lemma 10. (SKA$_\rho$ Progress) *In the absence of an active adversary, if in an iteration of SKA$_\rho$ c_A equals c_B, progress will eventually be made (i.e., a bit is shared and again, $c_A = c_B$). The expected number of iterations required to exchange one bit of an n-bit secret is $\frac{s(3+5\pi-2\pi^2)}{(1-\pi)^2}$.*

Proof. The result follows from Lemma 5, noting that s bits are sent for every one bit of key material. □

Although achieving perfect security in ρ-bounded isotropic channels may not be feasible, by using privacy amplification techniques, the expected information about the shared secret can be reduced to be exponentially small quite efficiently (cf., [3]). It is quite common for a cryptosystem to leak some information and yet be secure with respect to a certain desired property. Although such a cryptosystem would not be perfectly secure, it would still be information-theoretically secure. Therefore, the security properties guaranteed in such a system would hold even under the assumption of an computationally unbounded adversary.

6 Secure Key Agreement in the Presence of an Active Adversary

The above protocols offer strong confidentiality guarantees only when the adversary is passive.

In this section, we introduce a key exchange protocol that offers strong confidentiality guarantees even when Eve is active. An active Eve hears all messages and may, at will, insert messages into the channel and block messages from being delivered (see Section 3.2 for a full description of the capabilities of an active eavesdropper). Since the channel is isotropic, Eve cannot discriminate messages on the basis of their senders with probability greater than ρ. Nor can Eve transmit a message to one particular party. Because Eve can block all messages, denial-of-service is trivially realizable, and we therefore do not consider such attacks[4].

We divide the task of confidential key exchange in the presence of an active adversary into two phases. In the first phase, Alice and Bob (or possibly Alice and Eve, if Eve pretends to be Bob) exchange keys. As before, the key exchange protocol offers strong probabilistic guarantees (achieving asymptotically close to perfect secrecy) in the presence of a passive eavesdropper. We further claim that by being active, an active Eve becomes detectable to at least one honest party.

The second phase of key agreement, described in Section 6.2, allows the honest parties to object to the exchanged key (e.g., because Eve's presence is detected). Although an active Eve may ascertain bits of the key or otherwise influence the exchange in the first phase, her compromise will be reported in phase two and the key will be discarded. The verification process is run twice, permitting both Alice or Bob to notify the other party of a compromised or invalid key. The key must not be used until this verification process has completed without an objection being raised.

Thus, unlike traditional key agreement techniques such as unauthenticated Diffie-Hellman and quantum key exchange, our approach offers the interesting property that it prohibits an adversary from learning a key through active techniques (for example, man-in-the-middle attacks). We do not assume any *a priori* shared keys between the legitimate participants, and are nevertheless able to thwart active attacks. As we will show, provided that the honest participants are online, an active Eve's presence will be detected (except with an exponentially small probability due to channel loss) and the compromised key will be aborted.

6.1 Phase One: Key Agreement

In the first phase, Alice and Bob participate in a key agreement protocol that we call SKA$_A$. We assume that the parties have agreed on (1) the number of bits n that they wish to communicate, (2) positive integers k and γ, (3) time intervals for each step in the protocol, and (4) a *key transmission period* for sharing an n-bit key. It is assumed that Eve is aware of these parameters. Like SKA, SKA$_A$ proceeds as iterations of steps in which at most one bit of the key is shared. Each iteration is as follows:

1. Alice and Bob elect a sender via CASS. (Recall that CASS is used to mitigate the risk of collision.) If no party is elected, then neither party performs any action during the remaining steps – i.e., they wait patiently for the next iteration. Otherwise, from Lemma 1, we have at most one sender. We denote the sender as X and the receiver as Y.

[4] Note that many other key exchange schemes (e.g., Diffie-Hellman and quantum communication) are also vulnerable to DoS.

2. X broadcasts a number, r_X, chosen uniformly at random from the range $[0, 2^\gamma - 1]$.
3. Y broadcasts a number, r_Y, chosen uniformly at random from the range $[0, 2^\gamma - 1]$.
4. In the unlikely event that $r_X = r_Y$, Alice stops participating in the iteration. Additionally, if Alice does not receive r_B from Bob, she also stops participating.

   Otherwise, Alice broadcasts either the tuple $\{r_X, r_Y\}$ or $\{r_Y, r_X\}$, each with probability $\frac{1}{2}$.
5. Bob takes one of the following actions, depending upon the messages he has received thus far:
   (a) If in step 4 of the protocol Bob did not receive a tuple from Alice, he waits for a retransmission (see step 6). (Note that if Alice stops participating in the protocol, then eventually the time intervals for this and the next step will expire, neither party will record a bit, and the protocol will resume from step 1.)
   (b) If Bob received the tuple from step 4 but did not receive the message from Alice containing r_A (step 2 or 3), he broadcasts a $\{\texttt{NACK}\}$ (negative acknowledgment) message.
   (c) If Bob received the tuple from step 4 and received Alice's message containing r_A (step 2 or 3), he records the next bit of the key to be a `0` if Alice's tuple corresponds to $\{r_A, r_B\}$ or a `1` if the tuple corresponds to $\{r_B, r_A\}$. Bob then sends an *acknowledgment tuple* by broadcasting the tuple he received from Alice in the previous step.
6. Alice and Bob repeat steps 4 and 5 in lockstep at least k times in the following manner: Assuming Alice has not stopped participating, in every repetition she resends the tuple she transmitted in step 4. If Bob receives the rebroadcast tuple, he then transmits the same response that he sent in step 5 (i.e., either an acknowledgment tuple or $\{\texttt{NACK}\}$). These repetitions continue until Alice receives k responses from Bob or the time allotted for this step in the protocol expires. Although Bob responds to each received tuple from Alice, he records a bit only when he receives the first tuple, since all received tuples are identical and convey the same bit.

   If Alice receives k acknowledgment tuples, then she records the next bit in the key using the encoding scheme described in step 5(c). If Alice receives k $\{\texttt{NACK}\}$ messages, she does not record a bit and waits until the next iteration. If the iteration expires (due to timeout) before k responses are received by Alice, Alice does not record a bit during the iteration.

   If the key transmission period has not elapsed, a new iteration begins and both parties return to step 1.

A party stops participating in the protocol after the conclusion of the iteration in which it has obtained the nth bit of key. If at the end of the key transmission period a party has not recorded n bits, that party invalidates its key.

Note that the timeout value for step 6 should be sufficient to allow Alice to receive k responses from Bob, assuming a reasonable amount of loss.

We first show that SKA$_A$ protects the confidentiality of the key in the absence of an active adversary.

Theorem 11. *In the absence of an active adversary, the following must be true at the conclusion of SKA$_A$:*
(a) either Alice and Bob record the same key or Alice knows that a consistent n-bit key has not been shared,
(b) if Alice and Bob record the same key, each bit in the key has equal probability of being `0` or `1`, and
(c) if Alice and Bob record the same key, Eve can learn a bit of the key with probability $\rho' = \rho^2(3 - 2\rho)$.
Proof. A proof is provided in the Appendix.

If the timeout for step 6 is insufficient for Alice to receive k acknowledgment messages in any iteration in which Bob records a bit, Alice and Bob will have inconsistent keys. In such a case, Alice is aware of the inconsistency (by Theorem 11) and she can cause Bob to abandon the key by using the notification protocol described in the following subsection.

Lemma 12. (Progress) *In an iteration of SKA$_A$, assuming (1) a sufficiently large γ such that the probability that $r_A = r_B$ is negligible, (2) the absence of an active adversary, and (3) the key transmission period does not elapse, progress is eventually made (i.e., a bit is shared) and the expected number of iterations required to exchange one bit of an n-bit secret is $\frac{1}{\sigma}$, where $\sigma = \left(\frac{1}{2}\right)(1-\pi)^{(2k+3)} \sum_{i=k-1}^{z-1} \binom{i}{k-1} (2\pi - \pi^2)^{(i-k+1)}$ and z is the number of repetitions of steps 4 and 5 in a single iteration considering the time allotted.*
Proof. A proof is provided in the Appendix.

We now show that if Eve causes an inconsistency or learns bits by being active, she is detectable by either Alice or Bob with probability that asymptotically reaches 1.

Theorem 13. (Security in the Presence of an Active Eavesdropper) *Assuming a sufficiently large γ such that the probability that $r_A = r_B$ is negligible, the following consistency and confidentiality conditions hold at the conclusion of the protocol:*
(Consistency) *One of the following must be true:*
(a) Alice and Bob agree on the same n bits,
(b) at least one honest party knows the protocol did not succeed, or
(c) Alice and Bob each record n bits, differ on i bits ($0 < i \leq n$), and Eve is detected by at least one honest party with a probability lower bounded by $1 - \pi^{ki}$.
(Confidentiality) *If Alice and Bob agree on the same n-bit key, then both of the following must be true:*
(a) an active Eve can remain undetected and learn each bit of the key with probability at most ρ', where $\rho' = \rho^2(3 - 2\rho)$, and
(b) she can learn j bits ($0 \leq j \leq n$) and be discovered with a probability lower bounded by $1 - \pi^{kj}$.
Proof. A proof is provided in the Appendix.

If $\rho > \frac{1}{2}$, Alice and Bob can compensate for the partial isotropism by using privacy amplification. As with SKA$_\rho$, Alice and Bob can share $s \cdot n$ bits of secret and use the xor technique described in Section 5.2 to combine the bits into an n-bit key. Again, as long as ρ can be bounded less than 1, asymptotically close to perfect secrecy can be achieved at the cost of sharing a linear number of extra bits.

6.2 Phase Two: Key Verification

Immediately after the key transmission period, Alice and Bob initiate the *Secure Key Validation* (SKV) protocol. SKV gives one honest party the opportunity to notify the other that the key exchanged via SKA should be invalidated. SKV is executed twice, allowing both honest parties the opportunity to object to the key. We describe the case in which Alice wishes to inform Bob that the key should either be accepted or discarded. The protocol is symmetric when Bob wants to inform Alice.

SKV proceeds as iterations of steps. As before, we assume timeouts for each step of the protocol. If at any point in the protocol Alice receives a $\{$`Done`$\}$ message (see step 4), she stops participating in SKV.

1. Alice selects an integer r_A uniformly at random from the range $[1, 2^\gamma - 1]$ and a time t_A uniformly at random from $[0, \tau]$. Similarly, Bob chooses r_B uniformly at random from $[1, 2^\gamma - 1]$ and t_B uniformly at random from $[0, \tau]$. At time t_A, Alice broadcasts r_A. At time t_B, Bob broadcasts r_B. (Note that t_A precedes t_B with probability $\frac{1}{2}$.)
2. After time τ, Alice broadcasts r_B if wants Bob to accept his key or `0` if she she does not (note that $r_B \neq 0$). If Alice wants Bob to accept his key and she did not receive r_B in the previous step, she does not transmit.
3. If Bob did not receive a message in the previous step, the iteration is wasted and no progress is made. If Bob receives multiple messages in the previous step (indicating the presence of an active Eve), he invalidates his key. Otherwise, Bob checks whether the received message equals r_B. If the received message does not equal r_B, then Bob invalidates his key.

SKV repeats from the first step unless Bob wishes to terminate the protocol. This occurs when either Bob has invalidated his key or he has participated in $w \geq 1$ iterations of SKV in which he received r_B in step 2. Bob publicizes his desire to stop participating by transmitting a $\{\texttt{Done}\}$ message. Bob may have to send multiple $\{\texttt{Done}\}$ messages due to loss (in the case of loss, Alice will start a new round of SKV, at which point Bob can rebroadcast $\{\texttt{Done}\}$).

Eve cannot influence the sending of r_A and r_B in step 1 (she can of course block or insert messages, but doing so does not help distinguish r_A from r_B). Since the sender of one message in step 1 determines the identity of the other, Eve can use both messages to identify the senders of the intercepted messages. Eve identifies the senders if she either correctly attributes both messages or correctly attributes only one and guesses with probability $\frac{1}{2}$. Thus, the probability that Eve can identify the senders of the two messages transmitted in step 1 is $\binom{2}{2}\rho^2 + \frac{1}{2}\binom{2}{1}\rho(1-\rho) = \rho$.

Suppose that Alice wishes to cause Bob to invalidate his key. Alice therefore transmits $\texttt{0}$ in step 2. Eve can block such a message, but she must then replace it with r_B (otherwise, no progress is made), and she can identify r_B with probability ρ.

Theorem 14. *Assuming a sufficiently large γ such that the probability that $r_A = r_B$ is negligible, if Alice wants Bob to invalidate his key, then one of the following must be true:*

(a) at the conclusion of SKV, Bob will invalidate his key with a probability lower bounded by $1 - \max(\rho^w, \pi^w)$, or

(b) Eve conducts a denial-of-service attack.

Proof. A proof is provided in the Appendix.

Lemma 15. *In the absence of an active adversary, SKV will terminate.*

Proof. A proof is provided in the Appendix.

SKV is executed twice, allowing each party the opportunity to notify the other to discard its recorded key. Thus, Eve is left with two options. She can either remain passive, allowing Alice and Bob to share a key with confidentiality that asymptotically approaches perfect secrecy (and reaches perfect secrecy in the case of perfectly isotropic channels), or she can become active, causing the key to become invalidated. At best, Eve can perform a denial-of-service attack.

Limitation of SKV If both parties perceive the key to be valid prior to SKV, then Eve can cause one party to invalidate the key and the other to accept it. In the first execution of SKV (in which Alice notifies Bob that she accepts the key), Eve remains passive, and Bob does not invalidate his key. In the second execution, Eve remains passive until the wth iteration. Then, in step 2, she blocks Bob's transmission of r_B and inserts $\texttt{0}$. If Alice receives the $\texttt{0}$ message and Bob does not (the probability of this occurring is $\pi(1-\pi)$), Alice will invalidate her key while Bob will perceive his to be valid.

However, if this attack takes place and Alice subsequently uses the key to exchange a secret, Bob may safely accept the message. That is, Bob knows that Alice would not have used the key if it was compromised in the key agreement phase, since Theorem 14 guarantees that if either party wants to invalidate the key, they both will. Moreover, the confidentiality guarantees given in Theorem 13 apply, since for the attack to take place, both parties must have accepted the key prior to engaging in SKV.

7 Realizability of Isotropic Channels

As the requirements for sender and receiver are modest with our protocols (each needs a source of randomness), the practicality depends mainly on the realizability of isotropic channels over which they can communicate. If channels meeting the model in Section 3 do not exist in practice, then the protocols presented in this paper are of little use. Below, we show that perfectly isotropic channels, ρ-bounded isotropic channels, and channels that are effectively isotropic due to eavesdropper limitations are all potentially realizable, and in some cases already present in existing and emerging areas of communication.

Perfectly isotropic channels that convey no information about directionality are hard to come by. Even when higher level protocols contain no identifying information, the physical characteristics of the sender and receiver (voltage, power, etc.) may give away identity to an ideal eavesdropper. There is no indication, however, that such channels are theoretically impossible. The preliminary work in [20], for example, shows one type of classical channel that may provide isotropism (although the results of this paper have recently been disputed [17]). Additionally, recent work has proposed that isotropic communication is possible through laser oscillations [22]. Finally, as noted previously, information-theoretic anonymizing networks such as those proposed in [9] may also provide a way to build perfectly isotropic channels.

Fortunately, perfect isotropism is not required in order to achieve security. As long as there is some bounded uncertainty as to the true sender of messages, the SKA_ρ and SKA_A protocols may be employed. Such channels are much easier to imagine, even against an ideal eavesdropper. Mobile wireless networks, for example, allow an eavesdropper with direction-finding capabilities to guess who a transmission is from based on past position, but do not allow this to be done with certainty as each message comes from a different location. As networks with dynamic characteristics such as ad hoc sensor networks and near-field radio (e.g. RFID) become increasingly popular, we expect more ρ-bounded isotropic channels to appear in practice.

Under the very conservative model we have chosen in this paper, no restrictions are placed on the eavesdropper's budget or technical capacity. Even for channels that are not perfectly or ρ-bounded isotropic, limitations of more practical eavesdroppers may result in channels that are effectively isotropic. While we do not have room to formally analyze such weaker eavesdroppers in this paper, it is worth noting how they may arise. Limitations can be environmental, such as the eavesdropper's position, or internal, such as the eavesdropper's sensitivity or capacity. The channels described in [1], [8], and [7] are all examples of effectively isotropic channels. Additional examples include hubbed Ethernet segments, where the eavesdropper's position allows the hub to normalize physical signatures, or a wireless eavesdropper using commodity hardware, where power and timing information is insufficiently precise. Because these limitations may all be overcome with better positioning (or more eavesdroppers) or more money, these channels lack the strength of those described above. Unlike computational cryptography, however, these limitations must be dealt with in real-time. Advances in technology that incorporate more sensitive instruments into commodity devices do not help with data already intercepted.

8 Conclusions

We have introduced steps toward a theory of isotropic cryptography for information-theoretically secure key agreement in the presence of directional ambiguity. If Alice and Bob can bound an eavesdropper's ability to disambiguate the sender of an intercepted message, then Alice and Bob can conduct key agreement with provable security guarantees. The confidentiality of their secret reaches perfect secrecy in perfectly isotropic channels and asymptotically approaches perfect secrecy when $\rho > \frac{1}{2}$. Since our protocols rely on unconditional security, the confidentiality guarantees are independent of the (present or future) computational capabilities of an adversary.

The key agreement protocols presented in this paper are well-suited for message confidentiality and authentication. After a key has been shared, a Vernam cipher [24] can then be used to encrypt plaintext with perfect secrecy or confidentiality that is arbitrarily and exponentially close to perfect secrecy. Since the key agreement protocols remain efficient even when isotropism is not perfect, obtaining one secure key bit per message bit is feasible. In addition, the key agreement protocols can be used to share hash keys that can then be used in concert with unconditionally secure hash functions [25, 23, 16] to authenticate messages. Alice and Bob can share K_e, an encryption key, along with K_h, a hash key, and send messages of the form $\{m, h_{K_h}(m)\}_{K_e}$, resulting in (asymptotically close to) perfectly secure communication that is capable of being authenticated.

References

[1] Bowen Alpern and Fred B. Schneider. Key exchange using "keyless cryptography". *Information Processing Letters*, 16(2):79–81, 1983.

[2] Charles H. Bennett, François Bessette, Gilles Brassard, Louis Salvail, and John A. Smolin. Experimental quantum cryptography. In *Advances in Cryptology - EUROCRYPT '90*, volume 473, May 1990. Lecture Notes in Computer Science.

[3] Charles H. Bennett, Gilles Brassard, Claude Crépeau, and Ueli M. Maurer. Generalized privacy amplification. *IEEE Transactions on Information Theory*, 41(6):1915–1923, 1995.

[4] Charles H. Bennett, Gilles Brassard, and Jean-Marc Robert. Privacy amplification by public discussion. *SIAM J. Comput.*, 17(2):210–229, 1988.

[5] Charles H. Bennett and Peter W. Shor. Quantum information theory. *IEEE Transactions on Information Theory*, 44(6):2724–2742, 1998.

[6] Christian Cachin and Ueli M. Maurer. Unconditional security against memory-bounded adversaries. *Advances in Cryptology - CRYPTO '97*, 1294:292–306, 1997. Lecture Notes in Computer Science.

[7] Claude Castelluccia and Gildas Avoine. Noisy tags: A pretty good key exchange protocol for RFID tags. In *Seventh Smart Card Research and Advanced Application IFIP Conference*, Apr 2006.

[8] Claude Castelluccia and Pars Mutaf. Shake them up!: A movement-based pairing protocol for CPU-constrained devices. In *MobiSys '05: Proceedings of the 3rd international conference on Mobile systems, applications, and services*, pages 51–64, New York, NY, USA, 2005. ACM Press.

[9] David Chaum. The dining cryptographers problem: Unconditional sender and recipient untraceability. *Journal of Cryptology*, 1:65–75, 1988.

[10] Claude Crépeau. Efficient cryptographic protocols based on noisy channels. *Advances in Cryptology - EUROCRYPT '97*, 1233:306–317, 1997. Lecture Notes in Computer Science.

[11] Eric Cronin, Micah Sherr, and Matt Blaze. Listen too closely and you may be confused. In *Thirteenth International Workshop On Security Protocols*, 2005.

[12] Eric Cronin, Micah Sherr, and Matt Blaze. On the reliability of current generation network eavesdropping tools. In *Second Annual IFIP WG 11.9 International Conference on Digital Forensics*, Jan 2006.

[13] Imre Csiszár and János Körner. Broadcast channels with confidential messages. *IEEE Transactions on Information Theory*, 24(3):339–348, 1978.

[14] Whitfield Diffie and Martin E. Hellman. New directions in cryptography. *IEEE Transactions on Information Theory*, (6):644–654, Nov 1976.

[15] Michael J. Fischer and Rebecca N. Wright. Bounds on secret key exchange using a random deal of cards. *Journal of Cryptology*, 9(2):71–99, 1996.

[16] E. N. Gilbert, F. J. MacWilliams, and N. J. A. Sloane. Codes which detect deception. *Bell System Technical Journal*, 53(3), Mar 1974.

[17] Feng Hao. Kish's key exchange scheme is insecure. In *IEE Information Security*, 2006. *To appear*.

[18] Thomas Holenstein and Renato Renner. One-way secret-key agreement and applications to circuit polarization and immunization of public-key encryption. *Advances in Cryptology - CRYPTO '05*, 3621:478–493, 2005. Lecture Notes in Computer Science.

[19] Yuval Ishai, Eyal Kushilevitz, Rafail Ostrovsky, and Amit Sahai. Cryptography from anonymity. In *Proceedings of the 47th Annual IEEE Symposium on Foundations of Computer Science (FOCS'06)*, pages 239–248, 2006.

[20] Laszlo B Kish. Totally secure classical communication utilizing Johnson (-like) noise and Kirchoff's law. *Physics Letters A*, 2005.

[21] Ueli M. Maurer. Secret key agreement by public discussion from common information. *IEEE Transactions on Information Theory*, 39(3):733–742, 1993.

[22] Jacob Scheuer and Amnon Yariv. Giant fiber lasers: A new paradigm for secure key distribution. *Physical Review Letters*, 97(14), Oct 2006.

[23] Douglas R. Stinson. Universal hashing and authentication codes. *Advances in Cryptology - CRYPTO '91*, 576:74–85, 1991. Lecture Notes in Computer Science.

[24] Gilbert S. Vernam. Cipher printing telegraph systems for secret wire and radio telegraphic communications. *Journal of the American Institute of Electrical Engineers*, 55:109–115, 1926.

[25] Mark N. Wegman and J. Lawrence Carter. New hash functions and their use in authentication and set equality. *Journal of Computer and System Sciences*, 22:265–279, 1981.

Appendix: Selected Proofs

Lemma 1 *The* Collision Avoidance and Sender Selection (CASS) *protocol guarantees that, in the absence of an adversary who can inject messages, at most one sender will be chosen and if there are no losses, then there is a consensus at both ends about the sender.*

Proof. Since there are two parties, Alice and Bob, we denote the set of senders as perceived by Alice to be S_A, and similarly, S_B for Bob. At the start of the protocol, $S_A = S_B = \{A, B\}$. If a party X does not want to transmit, then, it performs $S_X \leftarrow S_X \setminus \{X\}$. If and when a notification (`1`) is received from the other party Y, the set $S_X \leftarrow S_X \setminus \{Y\}$. And if a round is discarded, then the sets S_A and S_B are set to $\emptyset$. We wish to prove that, at the conclusion of the protocol, (1) $|S_X| \leq 1, X = A, B$, (2) if $S_A = \{A\}$ then, $S_B \neq \{B\}$, (3) if $S_B = \{B\}$ then, $S_A \neq \{A\}$, and (4) if messages are not lost, $S_A = S_B$. Since we are assuming an adversary who cannot inject messages, we do not have to consider any other messages other than those sent by Alice and Bob. We consider all possible cases:

- *Both Alice and Bob want to send messages:* In this case, both Alice and Bob do not transmit anything and subsequently $S_A = S_B = \{A, B\}$ and the round is discarded. As a consequence, the sender sets are reset (i.e., $S_A = S_B = \emptyset$). Therefore $|S_X| \leq 1, X = A, B$ and $S_A = S_B$ hold vacuously.
- *Only Alice wants to send messages:* In this case, since Bob does not want to send, he performs $S_B \leftarrow S_B \setminus \{B\}$, and transmits a `1`. If this is not lost, it results in $S_A \leftarrow S_A \setminus \{B\}$ at Alice's end. At the end of the round, $S_A = S_B = \{A\}$ and properties (1), (2), and (4) hold.

  If the `1` from Bob is lost, Alice discards the round and we have $S_A = \emptyset$ and $S_B = \{A\}$. Again, we have that $|S_X| \leq 1, X = A, B$.

  The case where only Bob wants to send messages can be treated analogously.

- *Both Alice and Bob do not want to send messages:* Here, Alice will perform $S_A = S_A \setminus \{A\}$ and Bob, $S_B = S_B \setminus \{B\}$. There are the following subcases to consider.

  - *There are no collisions and Alice's message is lost:* Here, Alice receives a `1` from Bob and performs, $S_A = S_A \setminus \{B\}$ leaving $S_A = \emptyset$. At the end of the round, $S_B = \{A\}$. Thus, we see that the desired property (1) holds.
  - *There are no collisions and Bob's message is lost:* This case can be handled similarly as the one above.
  - *There are no collisions and both Alice's and Bob's messages are lost:* At the end of the round, $S_A = \{B\}$ and $S_B = \{A\}$ and again, the desired property, $|S_X| \leq 1, X = A, B$ holds.
  - *There is a collision:* By assumption, if messages collide, they are lost and therefore, this case is similar to the case of no collisions with both messages being lost.

Thus, we see that $|S_X| \leq 1, X = A, B$ holds in all cases and if there are no losses, $S_A = S_B$.

□

Lemma 3 *In the absence of an adversary who can inject messages, the expected number of iterations of CASS required to elect a sender using the* Collision Avoidance and Sender Selection (CASS) *protocol is* $2/(1-\pi)$.

Proof. Define A as the event that Alice wishes to transmit and B as the event that Bob wishes to transmit. By the definition of the protocol, $Pr[A], Pr[B], Pr[\neg A], Pr[\neg B] = \frac{1}{2}$. Define X as the event that a given `1` message (sent when a party does not want to transmit) is received by the other party. Hence, $Pr[X] = 1-\pi$. A sender is elected iff exactly one party wishes to transmit and the `1` message from the party that does not want to transmit is received by the party that does want to transmit. The probability that a sender is elected in an iteration of CASS is $Pr[A, \neg B, X] + Pr[\neg A, B, X] = \left(\frac{1}{2} \cdot \frac{1}{2} \cdot (1-\pi)\right) + \left(\frac{1}{2} \cdot \frac{1}{2} \cdot (1-\pi)\right) = \frac{(1-\pi)}{2}$.

Since we are interested in the expected number of iterations until the first successful instance of CASS, we can consider a geometric distribution with $p = (1-\pi)/2$ (the probability of success on each trial). Therefore, the expected number of iterations to elect a sender is $1/p = 2/(1-\pi)$. □

Theorem 4 (SKA Message Consistency) *In the absence of an active adversary, SKA ensures that both parties agree on the same sequence of n bits at the conclusion of the protocol.*

Proof. We prove the theorem by showing that if a bit is accepted, the same bit is accepted by both parties. Without loss of generality, assume that Alice and Bob have agreed upon i bits, where $0 \leq i < n+1$, and therefore $c_S, c_R = i$. (Note that at the start of the protocol, $c_S, c_R, i = 0$.) By Lemma 1, at step 1 of the protocol, either Alice or Bob will be elected the sender via CASS or the iteration will be wasted. In the latter case, no progress is made, and we therefore only consider the case in which CASS succeeds in electing a sender, S. We denote the receiver as R. Following step 2 of SKA, S broadcasts the message $\{i\}$. Due to loss, there are three cases to consider:

1. R **receives** $\{i\}$ **and its** $\{$`ACK`$\}$ **is received by** S**:** By the definition of the protocol, both parties record the same bit (depending on the identities of S and R) and increment their counters (i.e., $c_R \leftarrow i+1, c_S \leftarrow i+1$).

2. R **does not receive** $\{i\}$**:** Since R does not receive the message, it does not send an $\{$`ACK`$\}$. (R may be unaware that S has been elected via CASS, since Lemma 1 ensures that at most one sender will be elected, but does not guarantee consensus among both parties. Regardless of whether R can differentiate between the loss of the message $\{i\}$ and the case in which no sender is elected, R does not send an $\{$`ACK`$\}$.) Neither party will record a bit (recall that S records a bit only when it receives an

$\{\texttt{ACK}\}$), and neither party increments its counter. The iteration is effectively wasted and no progress is made.

3. R **receives** $\{i\}$**, and its** $\{\texttt{ACK}\}$ **is not received by** S**:** Since R receives $\{i\}$, it records a bit of key and increments its counter. S neither records a bit nor updates its counter. Now, $c_S = i$, $c_R = i + 1$, and R has one more bit than S. The protocol will then continue from step 1, and by Lemmas 1 and 2, a sender S_2 and receiver R_2 will be randomly chosen. There are two cases to consider:

   (a) $R_2 = R, S_2 = S$**:** S_2 will broadcast $\{i\}$. If R fails to receive this message, then by case 2, no progress is made. In the next subsequent iteration in which a sender is chosen via CASS, we return to either this subcase (3a) or the next (3b). If R_2 receives $\{i\}$, by the definition of the protocol, it erases the previously recorded bit, sets $c_{R_2} \leftarrow i$, and does not transmit an $\{\texttt{ACK}\}$. At this point, $c_R, c_S = i$ and both R and S have agreed on the same set of bits.

   (b) $R_2 = S, S_2 = R$**:** S_2 will broadcast $\{i + 1\}$. If R_2 fails to receive this message, then by case 2, no progress is made. In the next subsequent iteration in which a sender is chosen via CASS, we return to either this subcase (3b) or the previous (3a). If R_2 receives $\{i+1\}$, by the definition of the protocol it does not respond (since $c_{R_2} = c_{S_2} + 1$), and no progress is made (since S_2 does not receive an $\{\texttt{ACK}\}$ from R_2).

Note that progress is only made in cases 1 and 3. However, as we see from the subcases of case 3, any progress made in case 3 will eventually be rolled back (since case 3a rolls back the bit previously recorded only by R and sets $c_R \leftarrow i$, and no progress is made in case 3b). Hence, for a bit to be accepted by either party as part of the key at the conclusion of the protocol, case 1 must occur, in which case they both agree on the bit.

As a consequence of the above, $|c_S - c_R| \leq 1$ at all times.

We now consider the terminal case in which one party, whom we denote as S, receives an $\{\texttt{ACK}\}$ for the $(n+1)$th bit. This corresponds to case 1 above. We then have that $c_S = c_R = n+2$. That is, both Alice and Bob have shared $n + 1$ bits. At this point, S will broadcast $\{\texttt{Done}\}$ messages until it receives a $\{\texttt{DoneACK}\}$ response from R. The protocol will then terminate, and Alice and Bob will use the first n bits as their key. As shown above, both parties will agree on this sequence of n bits. □

Lemma 5 *In the absence of an active adversary, if in an iteration of SKA c_A equals c_B, progress will eventually be made (i.e., a bit is shared and again, $c_A = c_B$). The expected number of iterations required to exchange one bit of an n-bit secret is $\frac{3+5\pi-2\pi^2}{(1-\pi)^2}$.*

Proof. Let us denote the expected number of iterations required to exchange a 1-bit secret as x. By lemma 3, the expected number of iterations for sender selection is $\frac{2}{1-\pi}$. After the sender is selected, we have to consider different possibilities for message loss as in Theorem 4.

1. R **receives** $\{i\}$ **and its** $\{\texttt{ACK}\}$ **is received by** S**:** The probability of this happening is $(1-\pi)^2$. In this case, both parties reach an agreement and therefore the number of iterations is 1.

2. R **does not receive** $\{i\}$**:** The probability that R will not receive the message is $(1 - \pi)$. Since in this case R will not respond, there will be no progress made and the protocol starts over. Therefore the expected number of iterations is $(1 + x)$.

3. R **receives** $\{i\}$**, and its** $\{\texttt{ACK}\}$ **is not received by** S**:** The probability of this happening is $\pi(1 - \pi)$. In this case, the protocol makes no progress and several iterations must be carried out before the extra bit recorded by R is rolled back. Let us denote the number of iterations required to roll back the extra bit to be y. Therefore, the expected total number of iterations is $x+y+1$. We now proceed to analyze the sub-cases to evaluate y.

We define S_2 and R_2 as the sender and receiver, respectively, chosen in the next iteration of CASS in which a sender is elected. We do not consider iterations in which no sender is elected, as these iterations are discarded and no progress is made.

(a) $R_2 = R, S_2 = S$**:** The probability of this case occurring is $\frac{1}{2}$ by Lemma 2. If R fails to receive the message sent by S, no progress will be made. The probability of this happening is π. The expected number of iterations spent in this case is $(y+1)$. If it is received, then the rollback will happen and the number of iterations will just be 1. This will occur with a probability $(1-\pi)$.

(b) $R_2 = S, S_2 = R$**:** Again, the probability of this happening is $\frac{1}{2}$. If R_2 fails to receive message sent by S_2, then no progress is made. If R_2 receives the message, then it does not respond, and again no progress is made. Therefore, in either case, the expected number of iterations wasted is $(y+1)$.

Therefore, from the sub-cases, we can write the following equation for y:

$$y = \frac{2}{1-\pi} + \frac{1}{2}\{\pi(y+1) + (1-\pi)\} + \frac{1}{2}\{\pi(y+1) + (1-\pi)(y+1)\}$$

Solving for y yields $y = \frac{6-2\pi}{(1-\pi)^2}$.

Now using all the cases, we can write the equation for x as,

$$x = \frac{2}{1-\pi} + (1-\pi)^2 + \pi(x+1) + \pi(1-\pi)(x+y+1)$$

Substituting for y and solving for x yields $x = \frac{3+5\pi-2\pi^2}{(1-\pi)^2}$. The probability of successfully accepting a bit is geometrically distributed with $p = 1/x$, since a bit is accepted on the first successful receipt of the message and its acknowledgment. Therefore, the probability that it will terminate in i rounds is $p_i = (1-p)^{i-1}p$ and since $\sum_{i=1}^{\infty} p_i = 1$, progress is eventually made. □

Theorem 6 (Perfect Secrecy) *In the absence of an active adversary, SKA achieves* perfect secrecy *in a perfectly isotropic channel and each bit in the shared key has equal probability of being 0 or 1.*

Proof. We prove that SKA achieves perfect secrecy in exchanging a one-bit secret. Since bits are independent, this result can then be generalized to a secret of any length.

We consider only iterations in which a bit is shared. From the proof of Theorem 4, a bit is shared only if R receives $\{c_S\}$, the corresponding $\{\texttt{ACK}\}$ is received by S, and $c_R = c_S$ prior to the sending of $\{c_S\}$. In all other cases, either no bit is shared (case 2 of Theorem 4), or R records a bit that will eventually be discarded (cases 3a and 3b). Note that iterations are independent; a discarded or rolled-back iteration has no influence on sender selection in the subsequent iteration.

From the definition of the protocol, the bit shared in an iteration of SKA is determined by the outcome of CASS (i.e., if the sender is Alice, the bit is `0` and if the sender is Bob, the bit is `1`). From Lemmas 1 and 2, at most one sender will be elected via CASS and Alice and Bob have equal probability of being selected. We do not consider iterations in which neither Alice nor Bob send, as these iterations are discarded and no bits are shared. If a bit is shared (i.e., the iteration is not wasted), then the exchanged bit has equal probability of being a `0` or a `1`.

In the case in which a bit is exchanged (case 1 in the proof of Theorem 4), $c_R = c_S$ and therefore $c_A = c_B$. The content of messages $\{c_S\}$ and $\{\texttt{ACK}\}$ do not yield any sender information, as two identical messages would be sent if the outcome of CASS were reversed. By assumption, $\rho = \frac{1}{2}$, and therefore Eve cannot otherwise determine the source of either message. Since the shared bit is determined solely by the identities of S and R, the probability that Eve can learn the exchanged bit is $\frac{1}{2}$, the *a priori* probability of guessing a bit correctly. □

Lemma 7 *In the absence of an active adversary, the probability that Eve can identify the sender during a non-discarded iteration of SKA is ρ', where $\rho' = \rho^2(3-2\rho)$.*

Proof. From the definition of SKA and Theorem 4, a bit of the key is shared iff a sender (S) is elected via CASS, the receiver (R) receives a message $\{c_S\}$ from S, and S receives an $\{\texttt{ACK}\}$ from R. As shown in the proof of Theorem 4, if a message from any step of the protocol is lost or blocked, the iteration is discarded and no progress is made.

The probability that Eve can identify the source of a transmission is ρ. All three messages are dependent – i.e., the sender of one message determines the senders of the other two. Thus, Eve correctly identifies S if she correctly attributes the sender of two or three of the three intercepted transmissions. Hence, $\rho' = \binom{3}{2}\rho^2(1-\rho) + \binom{3}{3}\rho^3 = \rho^2(3-2\rho)$. □

Lemma 8 *In the absence of an active adversary, each bit in a key shared using SKA$_\rho$ has equal probability of being a 0 or a 1.*

Proof. From Lemma 2, Alice and Bob have equal probability of being elected the sender in CASS. Therefore, each of the $s \cdot n$ bits shared before the xor'ing will equally likely be a `0` or `1`. To complete the proof, we will now prove that the result of xor'ing s random bits will also be truly random.

Consider an s-bit string. The bits of the string will xor to `0` provided there are even number of `1`'s in the string. If there are odd number of 1's, the result of xor'ing them will be `1`. We denote these probabilities by p_0 and p_1 respectively. Since as every string in $[0, 2^s)$ is equally likely, $p_0 = \sum_{i=0}^{2\lfloor \frac{s}{2} \rfloor} \binom{s}{i}\frac{1}{2^s} = \frac{1}{2}$. Similarly, $p_1 = \sum_{i=1}^{2\lceil \frac{s}{2} \rceil - 1} \binom{s}{i}\frac{1}{2^s} = \frac{1}{2}$. Since $p_0 = p_1$, we conclude that the result of xor'ing s random bits is also truly random.

□

Theorem 9 *In the absence of an active adversary, SKA$_\rho$ achieves confidentiality that asymptotically approaches perfect secrecy in a ρ-bounded isotropic channel. The probability that an eavesdropper identifies a bit of the key correctly is $\frac{1+(2\rho'-1)^s}{2}$.*

Proof. We prove that asymptotically-close to perfect secrecy is achieved for a one-bit key. The general case of an n-bit secret would then follow from the independence of each one-bit key exchange. Throughout this proof, we will consider channels that are not perfectly isotropic (i.e., $\frac{1}{2} < \rho < 1$).

From Lemma 7, the probability that the adversary correctly guesses a bit exchanged in one iteration of SKA$_\rho$ is $\rho' = \rho^2(3-2\rho)$. Note that $\frac{1}{2} < \rho < 1$ implies $\frac{1}{2} < \rho' < 1$.

We will now calculate the probability p_k that Eve correctly computes the result of the xor'ing. The probability that she gets a bit correct, as proved above, is ρ'. She obtains a bit of key material even if she gets an even number of the s bits incorrect. The probability that Eve gets any $2i$ bits incorrect is $\binom{s}{2i} \cdot (1-\rho')^{2i} \cdot (\rho')^{s-2i}$. We therefore get the overall probability

$$\begin{aligned} p_k &= \sum_{i=0}^{\lfloor \frac{s}{2} \rfloor} \binom{s}{2i} \cdot (1-\rho')^{2i} \cdot (\rho')^{s-2i} \\ &= \frac{(\rho' + 1 - \rho')^s + (\rho' - (1-\rho'))^s}{2} = \frac{1 + (2\rho' - 1)^s}{2} \end{aligned}$$

Since we are xor'ing the s bits to generate one bit of the key, for perfect secrecy, we would want to find a value of s such that, $p_k = \frac{1+(2\rho'-1)^s}{2} \leq \frac{1}{2}$ which is the probability of getting either `0` or `1`. Since $\frac{1}{2} < \rho' < 1, 0 < 2\rho' - 1 < 1$, we have

$$\lim_{s\to\infty} \frac{1 + (2\rho' - 1)^s}{2} = \frac{1}{2}$$

□

Theorem 11 *In the absence of an active adversary, the following must be true at the conclusion of SKA$_A$:*
(a) either Alice and Bob record the same key or Alice knows that a consistent n-bit key has not been shared,
(b) if Alice and Bob record the same key, each bit in the key has equal probability of being `0` *or* `1`*, and*
(c) if Alice and Bob record the same key, Eve can learn a bit of the key with probability $\rho' = \rho^2(3-2\rho)$.

Proof. (a) We ignore iterations in which neither party records a bit, as no progress is made. We consider the following cases:

- *At the end of the key transmission period, Alice has recorded fewer than n bits.* The result trivial holds. Note that since Alice cannot record more bits than Bob (because Alice records a bit only in response to Bob recording a bit), this case also applies when Bob has recorded fewer than n bits.
- *There are no iterations in which only Bob records a bit.* If Alice records a bit, she does so only in response to Bob's acknowledgment tuple. Furthermore, since Alice and Bob use the same encoding scheme (i.e., the next bit in the key is `0` if r_A proceeds r_B in the tuple, else the next bit is a `1`), if Alice records a bit, Bob records the same bit. Hence, if Alice and Bob record n bits, then they record the same key. Otherwise, Alice records fewer than n bits and the previous case applies.
- *There is at least one iteration in which only Bob records a bit.* The case in which only Bob records a bit occurs when Bob receives Alice's initial tuple and the timeout occurs before Alice receives k acknowledgment tuples from Bob.

  If Bob records fewer than n bits at the end of the key transmission period, the first case applies. Otherwise, when Bob has recorded his nth bit, Alice will have recorded fewer than n bits, since in all iterations either both parties accept a bit or only Bob accepts a bit. Since Alice has less than n bits, she will continue to engage in SKA$_A$. Bob will not respond in any step of the protocol since he already has n bits, preventing Alice from obtaining additional bits. At the end of the key transmission period, the first case applies (i.e., Alice has fewer than n bits and therefore knows that a consistent n-bit key has not been shared).

(b) We note that since Eve cannot inject messages, the only messages received by Bob will be from Alice, and *vice versa*. The result then follows from the definition of the protocol, since Alice (who is honest) and flips a fair coin transmits $\{r_A, r_B\}$ with probability $\frac{1}{2}$ and $\{r_B, r_A\}$ with probability $\frac{1}{2}$.

(c) The proof is identical to that of Lemma 7, noting that Eve can correlate messages sent in steps 1, 2, and 3 of the protocol. Messages sent in steps 4 and 5 do not reveal information concerning the senders of messages messages sent in the first three steps since messages 4 and 5 are always sent by Alice and Bob, respectively. Thus, as with Lemma 7, Eve has three messages that she can correlate to probabilistically identify the sources of r_X and r_Y. Hence, the probability that a passive eavesdropper can learn the exchanged bit is $\rho' = \rho^2(3-2\rho)$. □

Lemma 12 *In an iteration of SKA$_A$, assuming (1) a sufficiently large γ such that the probability that $r_A = r_B$ is negligible, (2) the absence of an active adversary, and (3) the key transmission period does not elapse, progress is eventually made (i.e., a bit is shared) and the expected number of iterations required to exchange one bit of an n-bit secret is $\frac{1}{\sigma}$, where $\sigma = \left(\frac{1}{2}\right)(1-\pi)^{(2k+3)} \sum_{i=k-1}^{z-1} \binom{i}{k-1} (2\pi - \pi^2)^{(i-k+1)}$ and z is the number of repetitions of steps 4 and 5 in a single iteration considering the time allotted.*

Proof. A bit is shared in an iteration of SKA$_A$ if CASS results in a sender being selected, messages r_A and r_B are received, Alice receives k acknowledgment tuples, and $r_A \neq r_B$. The probability of CASS succeeding is $\frac{(1-\pi)}{2}$ (from Lemma 3), the probability that neither r_A nor r_B is lost is $(1-\pi)^2$, and the probability of having k successful transmissions of the bit by both Alice and Bob is $(1-\pi)^{2k} \sum_{i=k-1}^{z-1} \binom{i}{k-1} (2\pi - \pi^2)^{(i-(k-1))}$. We compute the latter probability by noting that the reception of messages sent in steps 4 and 5 of the protocol are independent events and the transmission stops when Alice successfully receives k notifications from Bob. Since the bit is agreed at the first success of an iteration of SKA$_A$, the distribution is geometric. Therefore, the probability that it will terminate in i rounds is $p_i = (1-\sigma)^{i-1}\sigma$ and since $\sum_{i=1}^{\infty} p_i = 1$, progress is eventually made. □

Theorem 13 (Security in the Presence of an Active Eavesdropper) *Assuming a sufficiently large γ such that the probability that $r_A = r_B$ is negligible, the following consistency and confidentiality conditions hold at the conclusion of the protocol:*
(Consistency) *One of the following must be true:*

(a) Alice and Bob agree on the same n bits,

(b) at least one honest party knows the protocol did not succeed, or

(c) Alice and Bob each record n bits, differ on i bits ($0 < i \leq n$), and Eve is detected by at least one honest party with a probability lower bounded by $1 - \pi^{ki}$.

(Confidentiality) *If Alice and Bob agree on the same n-bit key, then both of the following must be true:*

(a) an active Eve can remain undetected and learn each bit of the key with probability at most ρ', where $\rho' = \rho^2(3 - 2\rho)$, and

(b) she can learn j bits ($0 \leq j \leq n$) and be discovered with a probability lower bounded by $1 - \pi^{kj}$.

Proof. The *consistency* result makes use of the following claims:

Claim 16. *In any iteration of SKA$_A$, Eve may cause only Alice to record a bit, but by doing so Eve is detected by either Alice or Bob with a probability lower bounded by $1 - \pi^k$.*

Proof. At the end of an iteration, Alice records a tuple iff she receives k acknowledgment tuples that are all identical to the tuple Alice transmitted in step 4. Since by assumption Bob does not record a bit, he does not send any acknowledgment tuples. Therefore, all acknowledgment tuples must be sent by Eve. Eve cannot direct her transmissions towards a particular party, so Bob receives the injected acknowledgment tuple with probability $1 - \pi$. If Bob receives any forged messages, he is alerted to Eve's presence. The probability that Eve is detected is lower bounded by $1 - \pi^k$.

Claim 17. *If in any iteration of SKA$_A$ only Bob records a bit, then either Eve will be detected with a probability lower bounded by $1 - \pi^k$ or Alice and/or Bob will know the protocol did not succeed at the conclusion of SKA$_A$.*

Proof. If at the conclusion of the protocol, Alice and Bob both share n bits, then the result immediately follows from Claim 16 since there must be at least one round in which only Alice records a bit.

It is not possible for a party to record more than n bits. If at least one party records fewer than n bits, then that party will invalidate its key.

Claim 18. *If in any iteration of SKA$_A$ Alice and Bob both record a bit and Eve causes a disagreement in the recorded bit, then Eve is detected with a probability lower bounded by $1 - \pi^k$,*

Proof. At step 4 of SKA$_A$, Alice has knowledge about r_A and X, where X is either r_B or a message implanted by Eve. Since Alice either transmits $\{r_A, X\}$ or $\{X, r_A\}$ by flipping a fair coin, Eve cannot influence her bit. In step 4 of the protocol, Bob is expecting a tuple that contains r_B. Suppose Alice sent $\{r_A, X\}$, encoding a 0. To cause Bob to record a 1, Eve should ensure that Bob receives $\{r_B, Y\}$ where Y is either r_A or a message implanted by Eve. Since $r_A \neq r_B$, the tuples $\{r_A, X\}$ and $\{r_B, Y\}$ must differ. For Alice to accept a bit, she must receive k identical acknowledgment tuples of the form $\{r_A, X\}$. Since Bob will never send such tuples (if he does, then he and Alice agree on the same bit), then Eve must insert all k acknowledgment tuples. If Bob receives any such forged tuple, he is alerted to Eve's presence. Thus, the probability that Eve is detected is lower bounded by $1 - \pi^k$.

The case in which Alice sends $\{X, r_A\}$ is symmetric.

Proof of consistency: If either Alice or Bob records fewer than n bits at the conclusion of the protocol, then s/he will invalidate her key, and the result holds. It cannot be the case that a party records more than n bits. We therefore focus on the case in which both Alice and Bob record exactly n bits.

If Alice and Bob record the same n bits, the result trivially holds. Assume that Alice's and Bob's recordings differ in i bit locations ($0 < i \leq n$). By Theorem 11, in the absence of an active adversary, Alice and Bob must agree on all n bits if both parties record n bits. Thus, the inconsistencies in the keys must be due to an active eavesdropper. For each of the i positions, one of the following three cases must apply. (Let j denote a bit-position in which Alice and Bob have opposite bits.)

1. If Alice and Bob recorded their jth bit in the same iteration, then from Claim 18, Eve is detected with a probability lower bounded by $1 - \pi^k$.
2. If Alice recorded her jth bit in an iteration prior to that in which Bob recorded his jth bit, then there must be at least one iteration in which only Bob records a bit (otherwise, Alice will have more than n bits). From Claim 17, Eve is detected with a probability lower bounded by $1 - \pi^k$.
3. If Bob records his jth bit in an iteration prior to that in which Alice records her jth bit, then there must be at least one iteration in which only Alice records a bit (otherwise, Bob will have more than n bits). From Claim 16, Eve is detected with a probability lower bounded by $1 - \pi^k$.

The *consistency* result then holds.

The *confidentiality* result makes use of the following Claims:

Claim 19. *If in any iteration of SKA$_A$, Alice sends a tuple in step 4 of the protocol that does not contain both r_A and r_B, then one of the following must be true:*
(a) the iteration is wasted,
(b) at the conclusion of the protocol, Alice and/or Bob will know that SKA$_A$ did not succeed, or
(c) Eve's presence is detected with a probability lower bounded by $1 - \pi^k$.

Proof. Using Claims 16 and 17, if only one party records a bit, then either Alice or Bob will invalidate her/his key(s) at the conclusion of the protocol or Eve is detected with a probability lower bounded by $1 - \pi^k$. If no party records a bit, then no progress is made. The result trivially holds for both cases.

We now consider the case in which both Alice and Bob record a bit. Since Alice transmits the tuple in step 4, it must contain r_A. We denote the other element of the tuple as X, and by assumption, $X \neq r_B$. Bob is alerted to Eve's presence if he receives one or more tuples from Alice since the tuple does not contain his value, r_B. Therefore, Bob will never send an acknowledgment tuple that contains r_A and X. Since in order to record a bit, Alice must receive k acknowledgment tuples that are identical to her tuple, Eve must insert at least k of these acknowledgment tuples. Eve cannot direct her transmissions towards Alice, and Bob learns of Eve's existence if he receives a single forged acknowledgment tuple. The probability that Eve is detected is therefore lower bounded by $1 - \pi^k$.

Claim 20. *If in any iteration of SKA$_A$, Alice and Bob record the same bit, then one of the following must be true:*
(a) each acknowledgment tuple sent by Bob in step 5 must be identical to the tuple Alice transmitted in step 4, or
(b) Eve is discovered with a probability lower bounded by $1 - \pi^k$.

Proof. Since, by assumption, Alice records a bit, she must receive k acknowledgment tuples that are identical to the tuple she transmitted in step 4. In each repetition of step 5, Bob broadcasts the same response. If Bob's response is identical to Alice's tuple, then the result trivially holds. Otherwise, Eve must forge all k acknowledgment tuples for Alice to accept the bit. If Bob receives any of these tuples, he is alerted to Eve's presence. In this case, the probability that Eve is discovered is lower bounded by $1 - \pi^k$.

Claim 21. *If in any iteration of SKA$_A$, Eve causes either both parties or neither party to be elected, then one of the following must be true:*
(a) the iteration is wasted,
(b) at the conclusion of the protocol, Alice and/or Bob will know that SKA$_A$ did not succeed, or
(c) Eve is detected with a probability lower bounded by $1 - \pi^k$.

Proof. There are two cases to consider:

- *Eve causes neither party to believe it has been elected.* As a result, neither Alice nor Bob will transmit in step 2 of SKA$_A$. If Eve does not transmit during the time window assigned to step 2, then both Alice and Bob will believe they have missed a tuple, and by Claims 16 and 17, the result holds. We now consider the case in which Eve broadcasts r_E. From Claims 16 and 17, the result holds if neither or only one honest party records a bit. Therefore, we focus on the outcome in which both parties record a bit. Here, Alice must receive k acknowledgment tuples that contain both r_A and r_E (note that she must receive r_E in step 2 or she will stop participating in the iteration). Since, by assertion, $r_A \neq r_B$, Bob will never transmit such acknowledgment tuples. Therefore, all k acknowledgment tuples must be transmitted by Eve. If Bob receives any such forged acknowledgment tuple, he is alerted to Eve's presence. The probability that Eve will be discovered is therefore lower bounded by $1 - \pi^k$.
- *Eve causes both parties to believe they have been elected.* If by being active Eve can convince both Alice and Bob that they are elected during CASS (this can occur when both honest parties want to transmit and Eve injects a 1), then Alice and Bob will both transmit during step 2 of SKA$_A$. Neither honest party will then transmit in step 3. The proof is therefore symmetric to the case above, swapping steps 2 and 3.

Proof of confidentiality: We first prove that Eve's ability to learn a bit through sender selection (CASS) is bounded by ρ. During CASS, Alice and Bob independently flip a fair coin to decide whether s/he wants to transmit. Eve cannot influence these coin flips. Additionally, since CASS is completely symmetric with respect to Alice and Bob, an active Eve cannot force a particular party to be elected with probability greater than ρ. That is, even if Eve can use some active technique to cause one of the two honest parties to become elected without her presence being revealed[5], she can only identify that party with probability ρ. Eve can also cause both parties or neither party to be elected, but from Claim 21, Eve will be detected at the conclusion of the protocol with a probability lower bounded by $1 - \pi^k$.

We now show that if Eve uses an active technique (i.e., blocking or broadcasting) to learn a bit with probability greater than ρ', then she will be detected with a probability lower bounded by $1 - \pi^k$. We consider iterations of SKA$_A$ in which Alice and Bob both record the same bit and neither party invalidates her/his key. From Claims 16, 17, and 18, in all other cases the result holds since either the iteration is wasted and no progress is made, a party invalidates his/her key (contradicting our assumption that Alice and Bob both record the same n bits), or Eve is detected with a probability lower bounded by $1 - \pi^k$.

Once a party X has been elected via CASS, X will transmit r_X in step 2 of the protocol and the other honest party, Y, will broadcast r_Y in step 3, despite any actions taken by Eve. Hence, prior to step 4, either Eve correctly guesses the bit with probability $\rho' = \rho^2(3 - 2\rho)$ (since by Theorem 11, Eve can correlate messages sent in steps 1, 2, and 3) or she is detected with a probability lower bounded by $1 - \pi^k$.

At step 4, Alice proposes the next bit of the key by broadcasting a tuple (again, Eve can do nothing to prevent Alice's broadcast, though she may prevent the tuple from being received). From Claim 19, either the tuple contains r_A and r_B or Eve is detected with a probability lower bounded by $1 - \pi^k$. Since the ordering of Alice's tuple is chosen based on a fair coin flip, Eve cannot influence Alice's choice. Moreover, since we assume that Alice and Bob record the same bit, then by Claim 20, Bob transmits at least k acknowledgment

[5] For example, an active Eve can cause her desired party to be elected by blocking all iterations in which CASS results in that party not being elected. Since Eve can successfully decipher the sender of a message in a ρ-bounded channel with probability at most ρ, this is her probability of succeeding in influencing the winner.

tuples that are all identical to Alice's tuple or Eve is discovered with a probability lower bounded by $1 - \pi^k$. Since the result holds trivially in the latter case, we assume that Bob's acknowledgment tuples are identical to Alice's tuples from step 4. That is, in every repetition of step 4 in an iteration, Alice transmits the identical tuple (this follows from the definition of the protocol), irrespective of any interference from Eve. In every repetition of step 5, regardless of Eve's interference, Bob either does not transmit (as in the case in which Alice's tuple is not received due to loss or blocking) or transmits an acknowledgment tuple which agrees with Alice's tuple. Thus, forging messages either has no effect on either Alice's or Bob's actions in steps 4 and 5 or it causes Eve to be discovered with a probability lower bounded by $1 - \pi^k$. Eve can, at best, cause Bob not to transmit in a repetition of step 5. Hence, at the conclusion of the iteration, either Eve knows the bit with probability ρ' or she is detected with a probability lower bounded by $1 - \pi^k$.

□

Theorem 14 *Assuming a sufficiently large γ such that the probability that $r_A = r_B$ is negligible, if Alice wants Bob to invalidate his key, then one of the following must be true:*

(a) at the conclusion of SKV, Bob will invalidate his key with a probability lower bounded by $1 - \max(\rho^w, \pi^w)$, or

(b) Eve conducts a denial-of-service attack.

Proof. We first show that, at best, Eve achieves a denial-of-service attack if she forges or blocks $\{\texttt{Done}\}$ messages. If during an iteration of SKV, Bob has not invalidate his key and Eve forges a $\{\texttt{Done}\}$ message, then either Alice will not receive the message (in which case the transmission has no effect) or Alice receives the message and stops participating in the protocol. The latter results in a denial-of-service attack, since Bob continues to engage in the protocol until either he invalidates his key or receives w r_B messages in step 2. If Eve blocks a $\{\texttt{Done}\}$ messages, she only succeeds in delaying progress since this only emulates (artificially high) loss.

Since Alice's goal is to cause Bob to cancel his key, she will transmit `0` in step 2, regardless of any action performed by Eve in step 1. If Bob receives `0`, he will invalidate his key. To prevent this from occurring, Eve must block `0`. If Eve does not transmit during step 2, then Bob receives no message and the iteration is discarded. If Eve sends anything other than r_A or r_B in step 2, then she needlessly increases the probability that Bob will invalidate his key. Thus, there are two cases to consider:

- *Eve guesses r_B and transmits her guess.* As previously shown, Eve's ability to identify r_B is ρ. If her transmission is not received (due to loss), then the iteration is wasted and no progress is made. With probability $1 - \rho$, Eve will incorrectly identify r_B (and instead transmit r_A), and Bob will invalidate his key.
- *Eve transmits both r_A and r_B.* Bob cancels his key if he receives either both messages or only r_B. The probability that this occurs is $(1 - \pi)^2 + \pi(1 - \pi) = (1 - \pi)$.

The result then holds, noting that Bob continues the protocol until either he invalidates the key or there have been w iterations in which Bob receives r_B in step 2.

□

Lemma 15 *In the absence of an active adversary, SKV will terminate.*

Proof. An iteration results in progress if Bob receives a message in step 2 of SKV. If Alice does not want Bob to invalidate his key, she transmits r_B in step 2. The probability that Bob receives r_B in step 2 is $(1 - \pi)^2$, since r_B must be received by Alice in step 1 and Alice's broadcast in step 2 must be received by Bob. If Alice wants Bob to invalidate his key, Alice will broadcast `0` in step 2. The probability that Bob receives this message is $(1 - \pi)$. Since $\pi < 1$, progress is made in either case with probability greater than

0. The result then holds, since Bob will continue the protocol until he has either canceled his key or received $w \geq 1$ responses in step 2. $\square$