Three Improved Algorithms for Multi-path Key Establishment in Sensor Networks Using Protocols for Secure Message Transmission

Jiang Wu and Douglas R. Stinson

Abstract—In this paper, we propose a security model to capture active attacks against multi-path key establishment (MPKE) in sensor networks. Our model strengthens previous models to capture more attacks and achieve essential security goals for multi-path key establishment. In this model, we can apply protocols for perfectly secure message transmission to solve the multi-path key establishment problem. We propose a simple new protocol for optimal one-round perfectly secure message transmission based on Reed-Solomon codes. Then we use this protocol to obtain two new multi-path key establishment schemes that can be applied provided that fewer than one third of the paths are controlled by the adversary. Finally, we describe another MPKE scheme that tolerates a higher fraction (less than $1/2$) of paths controlled by the adversary. This scheme is based on a new protocol for a weakened version of message transmission, which is very simple and efficient.
Our multi-path key establishment schemes achieve improved security and lower communication complexity, as compared to previous schemes.

Index Terms—sensor network, key establishment, secure message transmission

1 Introduction

Sensor networks consist of large numbers of wireless sensor nodes which have only limited memory as well as limited computational and communication capabilities. The sensor nodes are usually distributed randomly in a certain area for data acquisition and environment monitoring. After deployment, they operate unattended and without physical protection. They need to communicate with each other to accumulate data and (possibly) relay the data to a base station. In many applications, such as battle field surveillance, communications between sensor nodes have to be encrypted. At the same time, sensor nodes deployed in a hostile environment are prone to be captured and compromised.

A commonly studied key management approach for sensor networks is key predistribution, which installs cryptographic keys in sensor nodes before the nodes are deployed. Later, after the sensor nodes are deployed, they discover shared keys with their neighbouring nodes (i.e., within the wireless communication range of the nodes). If two nodes are in each other's communication range and they share a common key, then they can encrypt the messages between them using the shared key and hence establish a *secure link*. A large number of key predistribution schemes have been proposed in the literature, e.g., [8], [3], [14], [2], [15], [7], [17].

• *Jiang Wu and Douglas R. Stinson are with the David R. Cheriton School of Computer Science, University of Waterloo, Waterloo ON, N2L 3G1, Canada*
E-mail: dstinson@uwaterloo.ca

D. R. Stinson's research was supported by NSERC discovery grant 203114-06

A key predistribution scheme does not guarantee that each pair of nodes share a predistributed key. For two nodes A and B that do not share a key, they can establish a *path key* using a *secure multi-hop path* between them. On such a path, each two consecutive nodes have a secure link. A can transport a key K to B via this path. On each hop, K is transported using the secure link, encrypted using the key shared by the two consecutive nodes (see [8]).

To enhance the security of a path key, Chan, Perrig and Song [3] and Zhu *et al* [24] proposed to use multiple paths to transmit key shares. Suppose that there are m secure node-disjoint paths between A and B (for a discussion on how to find such paths, see [3]). Once n such paths are identified, A could send n key shares $s_1, \ldots, s_n$ to B, one share via each path. B recovers the key K as $K = s_1 \oplus \ldots \oplus s_n$. Note that the actual number of such paths can be estimated using the k-connectivity properties of a sensor network secured by key predistribution schemes (see, e.g., [3], [23]).

The path key establishment using $K = s_1 \oplus \ldots \oplus s_n$ is vulnerable to message dropping or altering. In [24], Zhu *et al* also proposed to use an (n, k) secret sharing scheme [20] to compute the shares and to recover the key. An (n, k) secret sharing scheme generates n shares for a secret s. With any k of these n shares, the secret s can be recovered. The secret sharing scheme enables B to recover the key when some shares are dropped.

To withstand message dropping and altering attacks, Huang and Mehdi [10] proposed a multi-path key establishment scheme (the HM scheme) based on Reed-Solomon (RS) codes. In the HM scheme, A chooses a

key and encodes it in an RS codeword which consists of multiple symbols. The symbols are sent to B via multiple paths. The RS code provides error-correction ability so that B can recover the key when some symbols are dropped or altered. However, we show in Section 3 that there are some deficiencies in the Huang and Mehdi scheme.

Deng and Han [5] proposed another RS code based multi-path key establishment scheme named JERT (Just Enough Redundancy Transmission). JERT is designed for two neighbouring nodes which have a direct communication link which provides authentication but not secrecy (e.g., a broadcast channel), over which B can send feedback to A. Unlike the HM scheme where A transmits all the symbols of a codeword at the same time, A transmits the symbols incrementally in JERT. When B has received enough symbols and recovers a key, B and A can run an authentication protocol over their direct link to verify the recovered key. If the key recovered by B is correct, then A will not send the remaining symbols. We analyze the JERT scheme in more detail in Section 3.

1.1 Our Contributions

We first define a model for multi-path key establishment (MPKE). This model enhances the model used in [10] and [5], which does not adequately address the question of secrecy of the established key. (As a result, the previous schemes are not secure.) Our model explicitly requires secrecy of the established key against active adversaries, and the schemes we construct are secure in this strengthened model.

Informally, there are two specific security objectives that need to be achieved:

reliability
: The adversary nodes should not be able to prevent B from computing the key K that was chosen by A.

secrecy
: From the point of view of the adversary nodes, the entropy of K (given the information that they observe) should be sufficiently high so that they cannot compute K.

Our initial observation is that the above objectives can be realized using a protocol for perfectly secure message transmission (PSMT). This connection has not been pointed out or utilized in most previous papers on the topic of MPKE (the sole exception being the 2004 paper by Wang [22], which considers a somewhat different problem). We note that constructions and bounds for PSMT have been studied extensively since the 1993 paper of Dolev *et al* [6], and we argue that this theory can be profitably applied in the context of multi-path key establishment.

We then propose a new optimal protocol for one-round PSMT based on Reed-Solomon codes. Our protocol is somewhat similar to a protocol found in Fitzi *et al* [9]; however, we require only a single Reed-Solomon codeword to be generated, split into pieces and transmitted over the various channels. We believe that our scheme is the simplest optimal one-round PSMT yet proposed.

We use our new PSMT protocol to obtain two new multi-path key establishment schemes that can be applied provided that fewer than one third of the paths are controlled by the adversary. Our first PSMT scheme works in the same setting as the HM scheme, where A does not need to receive feedback from B; it is in fact just a straightforward usage of PSMT for the purposes of MPKE. Our second PSMT scheme works in the same setting as JERT, where A can receive feedback from B to reduce message transmission. For this scheme, we make use of the fact that our PSMT scheme consists of transmitting a single Reed-Solomon codeword; this allows the symbols in the codeword to be transmitted incrementally, until B has obtained a sufficient umber of symbols to decode the codeword correctly.

We optimize the parameters of both these MPKE schemes so that A uses the minimum transmission possible for B to recover a secure key.

Our third MPKE scheme tolerates a higher fraction (less than $1/2$) of paths controlled by the adversary. This scheme is based on a new protocol for a "weakened" version of message transmission, which is very simple and efficient. To be specific, we sacrifice unconditional security (which is required in PSMT but not in MPKE) to obtain higher tolerance of adversary nodes.

1.2 Organization

The remainder of the paper is organized as follows. In Section 2, we describe the proposed model and some results on secure message transmission, Reed-Solomon codes and key derivation using resilient functions. In Section 3, we present and analyze the HM and JERT schemes. In Section 4, we present our first two new schemes and their analysis. In Section 5, we present our third scheme, which tolerates a less than $1/2$ fraction of paths controlled by the adversary. In Section 6, we conclude the paper.

2 The Model and Some Preliminaries

Our model for multi-path key establishment (MPKE) is an enhancement of the model used in [10] and [5]. Our model is described as follows:

1) In a sensor network secured using key predistribution schemes, there often are multiple node-disjoint paths between a specified source node A and a specified destination node B. Every two consecutive nodes (i.e., a *link*) on such a path have a common key, and no two of these paths contain any common nodes except for A and B. These paths are identified by A before key establishment takes place (e.g., using techniques such as those

described in [3]). A sends key establishment messages over the paths. The efficiency of the scheme is measured by *communication complexity*, i.e., the total amount of information that is transmitted over all the paths.

2) We will assume that a fraction e of these paths (where $0 \leq e < 1/2$) are controlled by an adversary. We call e the *error rate*. A path P is *controlled by an adversary* if there exists an adversary that has knowledge of the key corresponding to a link on the path P[1]. We assume that an adversary controlling a path P can observe, drop or alter any messages that are transmitted from A to B using the path P.
3) The goal of a key establishment scheme is to enable A and B to establish a key with sufficient entropy. This leads to two security requirements.
   a) First, the adversary should not be able to disrupt the protocol by preventing B from computing the same key K that A holds.
   b) Second, the adversary should be prevented from determining partial information about the established key. This idea is formalized by considering the entropy of the message M received by B, from the point of view of the adversary who collects partial information about M (see Section 2.2.1 for details).

Note that we are building on the model used in [10] and [5]; in particular, the first two requirements above, as well as requirement 3(a), are similar to those used in the prior works. However, the entropy of the established key is not analyzed in [10] and [5]. Entropy is a critical requirement for the established key to be secure, because a key with low entropy can easily be determined by the adversary by exhaustive search. Key entropy should be considered when evaluating the scheme, along with efficiency. This is why we have extended the model to include an explicit requirement regarding entropy of the key.

2.1 Secure Message Transmission

Perfectly secure message transmission (PSMT) was introduced in 1993 by Dolev *et al* [6]. We define PSMT protocols and summarize some relevant results in this section.

Suppose two parties A and B are connected by p channels. An adversary controls p_a (or fewer) of these channels, but it is not known which channels are controlled by the adversary. The adversary can observe, delete, or modify the information in these p_a channels.

1. An adversary who controls a node N in one of the paths P from A to B has access to all the keys stored in N. Suppose that one of these keys, say K, is stored by a node N' in another path P' from A to B. Therefore the adversary can read information encrypted using the key K, and thus the path P' will not be secure. So the number of paths controlled by the adversary can be greater than the number of nodes controlled by the adversary. See [22] for more discussion on this issue.

An *r-round (p, p_a)-perfectly secure message transmission scheme* is an interactive protocol between A and B which takes place in r rounds (denoted as rounds $1, \ldots, r$), such that the following properties are satisfied:

1) In each odd-numbered round, A sends information to B over each of the p channels connecting them.
2) In each even-numbered round, B sends information to A over each of the p channels connecting them.
3) After the rth round, A and B both possess a common key K which is an element of a prespecified key space $\mathcal{K}$.
4) The adversary has no information on the value of K (so the entropy of K, from the point of view of the adversary, is $\log |\mathcal{K}|$).

The *overhead* of a PSMT is defined to be the ratio

$$\frac{\text{amount of information transmitted over all } p \text{ channels}}{\text{length of the key } K}.$$

There is a large literature on PSMT. For our purposes, we are most interested in one-round protocols, since these are the simplest and best suited to be applied to multipath key establishment. It was proven in [6] that a 1-round (p, p_a)-perfectly secure message transmission scheme exists if and only if $p \geq 3p_a + 1$. It is shown in [9] that the overhead of a one-round PSMT satisfies the condition

$$\text{overhead} \geq \frac{p}{p - 3p_a}. \tag{1}$$

Furthermore, for all pairs (p, p_a) with $p \geq 3p_a + 1$, schemes that meet this bound with equality (i.e., *optimal overhead schemes*) are constructed in [9].

If $3p_a \geq p \geq 2p_a + 1$, then all is not lost. It is possible to construct 2-round (p, p_a)-PSMT in these cases [19], [13]. Alternatively, one can obtain one-round schemes that are not perfectly reliable (i.e., where condition 3 in the definition of PSMT is relaxed). Such schemes are constructed in [12], [16].

2.1.1 Comparison Between PSMT and MPKE

A PSMT assumes multiple channels connecting A and B. The channels not controlled by the adversary are assumed to provide unconditional secrecy and authenticity. The security of a PSMT scheme is unconditional, provided that no computational assumptions are made in the analysis of the protocol. Almost all PSMT in the literature are studied in the setting of unconditional security.

In an MPKE, information is transmitted over links in encrypted form using conventional secret-key cryptography. Therefore we do not expect an MPKE to provide unconditional security; the security will depend on the assumption that the encryption and authentication schemes are (computationally) secure. Additional computational assumptions may be required, depending on the scheme.

In summary, any PSMT can be used for MPKE, but there are reasonable and practical MPKE schemes that are not PSMT schemes.

2.2 Reed-Solomon Codes

Many PSMT protocols are based on Reed-Solomon codes, which we introduce now. There are different ways to construct RS codes. Each has its encoding and decoding algorithms. Here we only describe the general functionalities of the encoding/decoding algorithms. For details of the algorithms, see, e.g., [18], [11]. Simply speaking, the input of the RS encoding algorithm is a *message* $\mathbf{m} = (m_0, \ldots, m_{k-1}) \in F_q^k$ where F_q is a finite field of order q. The output of an RS encoding algorithm is $\mathbf{c} = (c_0, \ldots, c_{n-1}) \in F_q^n$ where $k \leq n \leq q$. $\mathbf{c}$ is called a *codeword*. Each element in $\mathbf{m}$ or $\mathbf{c}$ is called a *symbol*.

If it always happens that $c_i = m_i$ for $0 \leq i \leq k-1$, then the encoding is *systematic*. In this case, $m_0, \ldots, m_{k-1}$ may be called *information symbols* and $c_k, \ldots, c_{n-1}$ may be called *parity check symbols*. Not all RS encoding schemes are systematic, however.

The above-described RS code has *length* n and *dimension* k. Its *distance* is $d = n - k + 1$ (i.e., any two distinct codewords differ in at least $n - k + 1$ symbols).

A Reed-Solomon code is a *linear code*, which means that the codewords form a k-dimensional subspace of the vector space F_q^n. A commonly-used method of encoding a linear code is to construct a *generator matrix*, denoted G, whose rows form a basis for the code. Then, to encode a message $\mathbf{m}$, we compute $\mathbf{c} = \mathbf{m}G$.

During transmission, some symbols in a codeword $\mathbf{c}$ may be deleted or altered. Suppose that δ of the symbols in $\mathbf{c}$ are deleted, and ϵ other symbols in $\mathbf{c}$ are altered. Let $\mathbf{r}$ be the resulting *received vector*. The input of the decoding algorithm is $\mathbf{r}$. The output of the decoding algorithm is the codeword whose distance from $\mathbf{r}$ is minimized. It is a standard result in coding theory that this decoding algorithm will output $\mathbf{c}$ provided that

$$\delta + 2\epsilon < d. \tag{2}$$

In the case of an RS code, we have $d = n - k + 1$ and the condition (2) becomes

$$\delta + 2\epsilon \leq n - k. \tag{3}$$

Given any codeword $\mathbf{c}$, it is a simple matter to obtain the corresponding message $\mathbf{m}$, regardless of whether or not the code is systematic.

In the above-described RS code, a message consists of k symbols and a codeword consists of n ($n \geq k$) symbols. The code is termed an (n, k) RS code.

2.2.1 Evaluating the Secrecy of a Message

When RS codes are used to encode and transmit a secret message $\mathbf{m} = (m_0, \ldots, m_{k-1})$, we need to consider the *entropy* of $\mathbf{m}$ from an adversary's point of view. The original entropy of $\mathbf{m}$ is $k \log_2 q$ bits. Suppose that i symbols are received by the adversary. If $i \geq k$, then the adversary can recover $\mathbf{m}$, and the entropy of $\mathbf{m}$ is 0. If $i < k$, then the adversary can randomly guess $k - i$ additional symbols and recover a (possibly incorrect) message. In this case the adversary recovers the correct message with probability

$$\frac{1}{2^{(k-i)\log_2 q}}.$$

Therefore, when the adversary knows i symbols in $\mathbf{m}$, the entropy of $\mathbf{m}$ is

$$\max\{(k-i)\log_2 q, 0\}$$

bits.

2.3 Key Derivation and Resilient Functions

In the protocols we will be describing, the key K, which is derived from a k-tuple $\mathbf{m}$, should have sufficient entropy. We will use the number of symbols, instead of number of bits, to indicate the entropy. In this terminology, the entropy of a message $\mathbf{m}$ is k symbols.

To ensure that K is secure, we desire that $\mathbf{m}$ should have entropy at least ℓ symbols, for some prespecified value of ℓ. There are several ways to derive K from $\mathbf{m}$ while preserving its entropy. For example, we can use a cryptographic hash function $hash$ to compute $K = hash(\mathbf{m})$. If it holds that

1) the hash function is modelled as a random oracle,
2) the input of the hash function has entropy at least ℓ symbols, and
3) the output of the hash function has a length of at least ℓ symbols,

then the entropy of the derived key K is at least ℓ symbols (so we say that K is *ℓ-secure*).

The above approach only provides computational security of the key. An alternative is to use resilient functions [1], [4] to derive the key. This approach would provide unconditional security of the key.

Suppose q is a prime power. Let k, ℓ, t be positive integers such that $k \geq \ell + t$. A *(k, ℓ, t, q)-resilient function*, or *(k, ℓ, t, q)-RF*, is a function $f : F_q^k \rightarrow F_q^\ell$ such that $f(\mathbf{m})$ is uniformly distributed in F_q^ℓ whenever any t inputs are fixed and the remaining $k-t$ inputs are chosen independently and uniformly at random from F_q, e.g., by an adversary (here we are regarding f as a function with k inputs from F_q).

There is a large body of literature on resilient functions. For our purposes, we need a well-known class of resilient functions that is derived from Reed-Solomon codes. In fact, any linear code gives rise to a linear function. The following was proven for binary codes in [1], [4]. It was observed in [21] that the same result holds for codes over an arbitrary finite field.

Theorem 2.1: Suppose q is a prime power, and suppose there exists a linear code over F_q having length n, dimension k and distance d. Then there exists a $(n, k, d-1, q)$-RF.

Using Reed-Solomon codes, the following is an immediate corollary.

Corollary 2.2: Suppose q is a prime power such that $q \geq k > \ell$, where k and ℓ are positive integers. Then there exists a $(k, \ell, k - \ell, q)$-RF.

The construction of a $(k, \ell, k-\ell, q)$-RF is easy. Let G be the generator matrix of a Reed-Solomon code of dimension ℓ and length k over F_q. Then f is defined as $f(\mathbf{m}) = \mathbf{m}G^T$, where G^T denotes the transpose of G.

Remark. The above-described usage of resilient functions has previously been employed in the literature on PSMT; see, for example, the function **EXTRAND** in [16, §4.2]. However, surprisingly, the connection to resilient functions is not made in [16] or in other papers on PSMT.

3 Analysis of the HM and JERT Schemes

In this section, we analyze the HM and JERT schemes. The HM scheme does not achieve its stated objectives, and, moreover, it is insecure. The main problem with the JERT scheme is that the security is not proved under a strong enough adversarial model.

3.1 The HM Scheme

The HM scheme [10] is as follows. Let $n - k = 2t$. The number of paths controlled by the adversary is assumed to be at most t. Suppose that there are p node-disjoint paths between A and B, where $2t < p \leq k$. A chooses a message $\mathbf{m} = (m_0, \ldots, m_{k-1})$ and uses a systematic (n, k) RS encoding algorithm to generate a codeword $\mathbf{c} = (m_0, \ldots, m_{k-1}, b_0, \ldots, b_{2t-1})$. $m_0, \ldots, m_{k-1}$ are k information symbols and $b_0, \ldots, b_{2t-1}$ are $2t$ parity check symbols. Let $\mathbf{b} = (b_0, \ldots, b_{2t-1})$. Then A creates k $(2t+1)$-tuples, each of the form $m_i \parallel \mathbf{b}$, and sends at most t of the $(2t+1)$-tuples on each of the p node-disjoint paths (note that this requires that $k \leq pt$, which is not stated as a necessary condition in [10]).

Since there are at most t paths that are controlled by the adversary and $p > 2t$, B can use majority rule to find the correct $\mathbf{b}$. It is then claimed in [10] that B can then recover $\mathbf{m}$, but ability to recover $\mathbf{m}$ also depends on how many information symbols have been altered by the adversary. In fact, we show that B may not be able to recover $\mathbf{m}$ at all in many situations. Suppose $k > p > 2t$ (note that it is assumed that $2t < p \leq k$, so we are just saying that $k \neq p$). Suppose that each message symbol is transmitted by one path. Then there is at least one of the p paths, say P_0, that is used to transmit at least two message symbols. If P_0 is one of the t paths controlled by the adversary, then the adversary can alter at least $t+1$ message symbols. However, an RS code can only correct t errors, so the message cannot be recovered by B.

Another problem with the scheme in [10] is that adversary can obtain information about the message. Recall that the scheme is supposed to tolerate up to t compromised paths. However, if t paths are controlled by the adversary, then the adversary receives $2t$ correct parity check symbols and at least t correct information symbols, and hence by (3) the adversaries collectively are able to recover $\mathbf{m}$ when $3t \geq k$ (equivalently, when $n \leq 5t$). Even when there is only one adversary node, it will receive $2t$ parity check symbols and at least one information symbol. Then the entropy of the key is $\max\{k - 2t - 1, 0\}$ symbols, which could be very low.

We regard it as a weakness in the scheme for A to send all the $2t$ parity check symbols on every path, because for decoding of RS codes, a parity check symbol yields the same amount of information about the message as an information symbol does. Another problem (as noted in [5]) is that the scheme is quite inefficient due to the amount of repeated information that is transmitted.

3.2 The JERT Scheme

JERT [5] is designed for two neighbouring nodes that have a direct channel that provides message integrity but not secrecy, e.g., a broadcast channel. In this case, A and B can run a challenge-response authentication protocol over this insecure channel to verify if they share a common secret key. The communication overhead over the direct link is neglected in the analysis of the efficiency of the scheme.

Here are the details of the scheme. A chooses $\mathbf{m} = (m_0, \ldots, m_{k-1})$, encodes it into a codeword $\mathbf{c} = (c_0, \ldots, c_{n-1})$, and derives a key K from $\mathbf{m}$. Then A selects p node-disjoint paths between A and B. A divides the n symbols into R groups. Group j contains r_j symbols. It holds that

$$\sum_{j=1}^{R} r_j = n.$$

A sends the n symbols in R rounds. In round j, the r_j symbols in group j are sent over the p paths. For each path i, A computes *fraction parameters* q_i $(0 \leq q_i \leq 1)$, where

$$\sum_{i=1}^{p} q_i = 1.$$

Then A sends $r_j q_i$ symbols over path i in round j. In each round, if B can recover an $\mathbf{m}'$ using all the received symbols, then B derives a key K' from $\mathbf{m}'$, and runs an authentication protocol with A over their direct link to verify if $K = K'$. A keeps on sending the codeword symbols until the authentication protocol indicates that $K' = K$ or until all n symbols are sent.

The main purpose JERT is for A to send *just enough* symbols for B to recover K, instead of transmitting all symbols as in the HM scheme. JERT may be thought of as an *adaptive* algorithm, whereas HM is *non-adaptive*.

The security analysis in [5] discusses three attack scenarios:

1) All adversary nodes are passive. In this case, the probability that a given fraction of the symbols are received by the adversary is computed.
2) All adversary nodes are active. In this case, the number of symbols that must be sent so that B can recover the key is computed.

3) Some adversary nodes are active and some are passive. This case is not analyzed in [5], where it is stated that an analysis of this case "would be quite complex".

We believe that an adequate security model should consider any combination of active and passive adversaries.

4 Two New Schemes for MPKE Based on Reed-Solomon Codes

In our schemes, we consider all possible attack scenarios. First, the adversary nodes cannot make B accept an incorrect key, even if all the adversary nodes are active. Therefore, since B learns the correct key, K, we only need to consider how much information the adversary nodes can derive about K. For this analysis, we allow adversary nodes to be active or passive.

In this section, we propose two multi-path key establishment schemes, **Protocol 1** and **Protocol 2**. These protocols are obtained from a new and extremely simple PSMT protocol based on (n, k) RS codes. In **Protocol 1**, as in the HM scheme, A does not receive feedback from B. In **Protocol 2**, as in the JERT scheme, A receives feedback from B. For both schemes, we are interested in finding the optimal choice of (n, k) values such that B can recover a key with the desired entropy while A only transmits the minimum possible number of bits (i.e., the transmission overhead is optimized).

As mentioned above, our schemes are based on RS codes over finite field F_q. We assume that q is fixed and $q \geq n$ in the chosen (n, k) RS code.

4.1 Protocol 1

Here are the details of our first MPKE protocol, which is in fact a 1-round PSMT scheme if the parameters are chosen appropriately. We refer to this protocol as **Protocol 1**.

1) A chooses a random message $\mathbf{m} = (m_0, \ldots, m_{k-1}) \in F_q^k$ and encodes it into an RS codeword $\mathbf{c} = (c_0, \ldots, c_{n-1}) \in F_q^n$.
2) A sends the n codeword symbols over p pre-specified node-disjoint paths. Note that the number of symbols sent over any path is either $\lceil \frac{n}{p} \rceil$ or $\lfloor \frac{n}{p} \rfloor$.
3) B decodes the received symbols to a codeword $\mathbf{c}'$. Then a message $\mathbf{m}'$ is derived from $\mathbf{c}'$. Finally, a key K' is derived from $\mathbf{m}'$ using a pre-specified key derivation function.

We discuss feasible and optimal choices of (n, k) values in Section 4.1.1.

Remark. When B receives a symbol c_i, B also needs to know its index i for decoding purposes (this applies to the HM scheme and JERT as well). This objective could be accomplished, for example, if A and B have some synchronization mechanism. In any event, we assume that B has some reliable means of knowing the index of any received symbol.

4.1.1 Analysis and Optimization

Our goal is that the key K (derived from $\mathbf{m}$) has entropy ℓ symbols if $\mathbf{m}$ has entropy at least ℓ symbols. We will derive conditions to ensure that $\mathbf{m}$ has entropy at least ℓ symbols. Then the key derivation function is just a $(k, \ell, k-\ell, q)$-RF, obtained from Corollary 2.2. This would provide A and B with an unconditionally secure key in F_q^ℓ.

Suppose there are p node-disjoint paths from A to B and p_a of these paths are controlled by the adversary. Therefore the error rate is $e = p_a/p$. For simplicity, assume that n/p is an integer. Then in **Protocol 1**, $n/p \times p_a = ne$ symbols will be received by adversary nodes. These ne symbols may be altered or deleted.

First, we derive a condition to ensure *reliability* (i.e., so that B can correctly compute the key K). Since altering symbols makes it most difficult for B to recover $\mathbf{m}$, we assume that these ne symbols are all altered. For B to be able to correctly recover $\mathbf{m}$, the condition (3) becomes

$$ne \leq \frac{n-k}{2}. \tag{4}$$

Observe that (4) implies that $ne < n/2$, so $e < 1/2$.

Remark. If the scheme satisfies (4), then it is already a *perfectly reliable message transmission* scheme (for a definition, see [16]).

Now we consider *secrecy* of the transmitted message. In order for $\mathbf{m}$ to have entropy at least ℓ symbols, we require that

$$k - ne \geq \ell. \tag{5}$$

Using the fact that the desired entropy $\ell > 0$, it can be seen that (4) and (5) together imply that

$$\frac{k}{1-2e} \leq n < \frac{k}{e},$$

which yields $e < 1/3$.

So hereinafter we assume that $e < 1/3$. Under this assumption, (4) and (5) are equivalent to

$$\frac{k-\ell}{e} \geq n \geq \frac{k}{1-2e}. \tag{6}$$

The inequalities in (6) provide the conditions under which B can compute a key with entropy at least ℓ symbols. Note that (6) can equivalently be expressed as follows:

$$n(1-2e) \geq k \geq \ell + ne. \tag{7}$$

Suppose a value ℓ is fixed. Then we define an ordered pair (n, k) to be *e-feasible* if (6) (equivalently, (7)) is satisfied.

Given ℓ and e, the set of all e-feasible ordered pairs form a region, a typical example of which is indicated by the shadowed area in Figure 1. The optimal solution will be an ordered pair of integers (n, k) that is close to the ordered pair (n_{min}, k_{min}), which denotes the intersection

of the two lines $k = n(1-2e)$and $k = \ell + ne$. It is easy to compute

$$n_{min} = \frac{\ell}{1-3e} \qquad \text{and} \qquad k_{min} = \frac{\ell(1-2e)}{1-3e}. \quad (8)$$

Clearly $n_{min} > 0$ and $k_{min} > 0$ because $e < 1/3$. n_{min} represents the optimal transmission size in the protocol.

Theorem 4.1: Suppose ℓ is a positive integer and $0 \leq e < 1/3$. Suppose (n, k) is e-feasible. Finally, suppose that there are p disjoint paths from A to B, where $p_a = pe$ of these paths are controlled by the adversary. Suppose that n/p is an integer. Then **Protocol 1** yields an ℓ-secure secret key. The total transmission of **Protocol 1** consists of n symbols.

If we apply **Protocol 1** with $(n, k) = (n_{min}, k_{min})$, then the *transmission overhead* is

$$\frac{n}{\ell} = \frac{1}{1-3e} = \frac{p}{p-3p_a},$$

which is optimal, by (1).

Protocol 1 is analyzed in terms of the error rate e. In general, the error rate will not be known. In practice, Alice and Bob would choose a value $e^* < 1/3$ which they hope is an upper bound on e. They would then execute **Protocol 1** with an e^*-feasible ordered pair (n, k). It is easy to see that an e^*-feasible ordered pair is also e-feasible provided that $0 \leq e \leq e^*$, so **Protocol 1** will still work correctly in these circumstances.

Example. Suppose that $e = 1/5$, $p = 5$ and $\ell = 40$. Then we can take $n = 100$ and $k = 40$ in Theorem 4.1. That is, we obtain a 40-secure key using a $(100, 40)$ RS code under the assumption that at most one of five node-disjoint paths joining A and B is controlled by the adversary.

4.2 Protocol 2

In **Protocol 1**, if the actual error rate $e < e^*$, then the protocol might transmit more information than is actually necessary, i.e., the efficiency might not be optimal. To reduce the number of transmitted symbols, A can use feedback from B. This idea was first proposed in JERT [5]. JERT is designed for two neighbouring nodes that can communicate directly. It is assumed that the channel connecting A and B is a broadcast channel. Therefore, it provides data integrity, but no confidentiality or data origin authentication. The lack of confidentiality or authentication is not a problem, as this channel is used only for message authentication. We assume a similar channel in our protocol.

Next we define **Protocol 2**, where B will send feedback to A using the broadcast channel. Let e^* be the maximum error rate that the protocol is designed for (i.e., an e^*-feasible ordered pair (n, k) is chosen for use in the protocol). As before, assume that there are p node-disjoint paths from A to B and assume for convenience that $p \mid n$. In **Protocol 2**, MAC denotes a secure message authentication protocol.

1) A chooses a random message $\mathbf{m} = (m_0, \ldots, m_{k-1}) \in F_q^k$ and encodes it into an RS codeword $\mathbf{c} = (c_0, \ldots, c_{n-1}) \in F_q^n$.
2) In each of n/p *rounds*, A sends one codeword symbol over each of the p pre-specified node-disjoint paths.
3) After each round, B attempts to decode the symbols he has received in the current and all previous rounds to a codeword $\mathbf{c}'$. If he is successful, then a message $\mathbf{m}'$ is derived from $\mathbf{c}'$ and a key K' is derived from $\mathbf{m}'$ using the key derivation function.
4) If B is able to compute a (possible) key K', then B initiates a conventional message authentication code (MAC) based mutual authentication protocol with A over the broadcast channel. In the protocol, A and B both verify if they hold the same key. If the authentication succeeds, then both A and B stop. Since we assumed that the broadcast channel provides integrity, the adversary is not able to change the messages between A and B in the authentication protocol. If A and B have the same key, then the adversary is not able to prevent the authentication from succeeding.

4.2.1 Analysis

First, we consider the properties of security and reliability. We claim that **Protocol 2** is computationally secure whenever the mutual authentication protocol used in step 4 is computationally secure. It is certainly possible that B computes an incorrect key, but he will not accept a wrong key (except with very small probability) due to the mutual authentication protocol used in step 4 to prove possession of the new key. Eventually, after some number of rounds, B will be able to compute the correct key provided that $e \leq e^*$ (for details, see below). Therefore **Protocol 2** achieves reliability. Secrecy follows from the same analysis as for **Protocol 1**.

Protocol 2 is also secure against mobile adversaries. (A *mobile adversary* is allowed to compromise different nodes in different rounds, subject to the constraint that the error rate is at most e^* in any given round.)

Next, we analyze the efficiency of **Protocol 2** by determining the number of rounds required for B to be able to compute the correct key K. After r rounds, B has received rp symbols, at most rpe of which have been altered. The number of symbols which have not yet been transmitted to B is $n - pr$. Thus B has a received vector in which $\epsilon \leq per$ and $\delta = n - pr$. Referring to (3), B can correctly decode this received vector if

$$2rpe + n - pr \leq n - k,$$

which is equivalent to

$$r \geq \frac{k}{p(1-2e)}. \quad (9)$$

Therefore the correct key is computed by B after at most $\lceil k/(p(1-2e)) \rceil$ rounds. It follows that the *speedup factor*

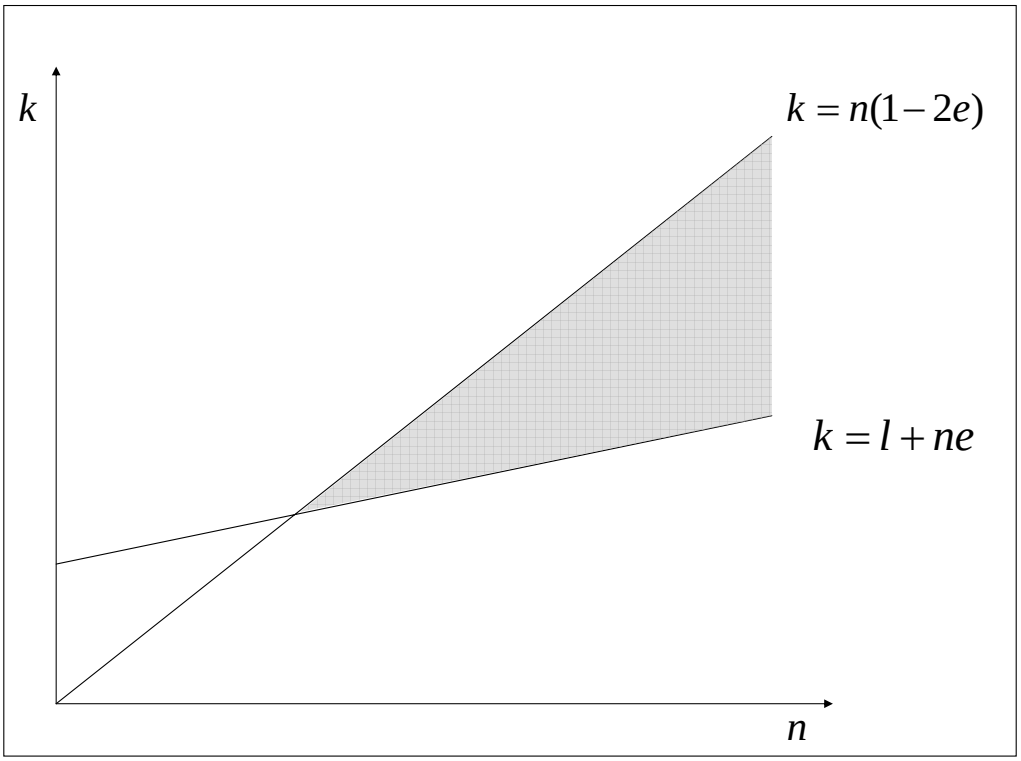

Fig. 1. e-feasible ordered pairs (n, k) for given error rate e and desired key entropy ℓ.

of **Protocol 2** as compared to **Protocol 1** is

$$\frac{1-2e^*}{1-2e}.$$

If $e = e^*$, then the number of rounds required is $\lceil k/(p(1-2e^*))\rceil$. If $e = 0$, then the number of rounds required is $\lceil k/p\rceil$.

Summarizing the above discussion, we have the following theorem.

Theorem 4.2: Suppose ℓ is a positive integer and $0 < e \leq e^* < 1/3$ and suppose (n, k) is e^*-feasible. Suppose that there are p disjoint paths from A to B, where $p_a = pe$ of these paths are controlled by the adversary. Then **Protocol 2** yields an ℓ-secure secret key. The total transmission of **Protocol 2** consists of (roughly) $n(1-2e^*)/(1-2e)$ symbols.

Remark. In practice, B would not attempt to decode the received vector after every round. The exact error rate $e = p_a/p$, where $p_a \leq pe^*$ is an integer. Using (9), we see that it is sufficient for B to decode a received vector only when a round r has the form

$$r = \left\lceil \frac{k}{p\left(1-\frac{2i}{p}\right)} \right\rceil = \left\lceil \frac{k}{p-2i} \right\rceil$$

for some integer $i \geq 0$.

5 A Scheme Tolerating Error Rate $< 1/2$

Both of our protocols described in Section 4 assume that the number of paths controlled by the adversary is less than a $1/3$ fraction of the number of paths connecting A and B. A higher fraction (less than $1/2$) of paths controlled by the adversary could be tolerated by using appropriate message transmission schemes mentioned in Section 2.1. These schemes either require additional rounds of communication or they are not perfectly reliable. They are also somewhat complicated and/or inefficient. In this section, we present a new 2-round protocol for a weakened version of message transmission which is very simple and efficient, and well-suited for application as a MPKE scheme. Our protocol will be computationally secure provided that certain specified ingredients exist.

Our scheme has the following properties:

- We assume that A and B are joined by p node-disjoint paths, at most p_a of which are controlled the adversary, where $p \geq 2p_a + 1$.
- We require a mapping $h : \mathcal{K} \to \mathcal{T}$, where $\mathcal{K}$ is the *key component space* and $\mathcal{T}$ is the *tag space*. We will take $\mathcal{K} = F_q$ for some prime power $q \geq p$.
- The scheme will be perfectly reliable if h is injective (in this case, the scheme enables A and B to estab-

lish a shared key $K \in \mathcal{K}$ with probability equal to 1 independent of any computational assumptions).
- The scheme will be (computationally) reliable if h is second-preimage resistant.
- Under the assumption that h is a random function, the scheme provides secrecy of the established key. (For a random function, a computationally-bounded adversary is unable to compute any non-negligible information about a secret value L, when the adversary is given only the value $h(L)$.)
- The scheme is a two-round scheme.

Here is the protocol, which we term **Protocol 3**.

1) For $1 \leq i \leq p$, A chooses a *key component* $L_i \in \mathcal{K}$ independently and uniformly at random. Then A computes $h_i = h(L_i)$, $i = 1, \ldots, p$.
2) For $1 \leq i \leq p$, A sends L_i over the ith path. Also, for $1 \leq i, j \leq p$, $i \neq j$, A defines $h_{i,j} = h_i$ and sends $h_{i,j}$ over the jth path.
3) a) For $1 \leq i \leq p$, B computes
$$\mathbf{check}(i) = \{j : h_{i,j} = h(L_i)\}.$$
b) B *accepts* L_i if and only if $|\mathbf{check}(i)| \geq p - p_a - 1$.
c) B defines
$$\mathbf{accept} = \{i : B \text{ accepts } L_i\}$$
and $n = |\mathbf{accept}|$.
d) B defines $\mathbf{m} = (L_i : i \in \mathbf{accept})$.
e) B computes $K = f(\mathbf{m})$, where f is an $(n, p - p_a, n - (p - p_a), q)$-resilient function.
4) B transmits **accept** to A over every one of the p paths.
5) a) A determines **accept**, as it will be correctly received over at least $p - p_a$ paths (i.e., a majority of the paths).
b) A computes $\mathbf{m}$ and K exactly as B did.

Remark. As described above, **Protocol 3** is not a message transmission scheme due to the fact that the value of the derived key, K, is not specified *a priori*; its value depends on possible actions of the adversary. This is sufficient for the goals of an MPKE scheme. However, if desired, it is easy to use a standard trick to make a minor alteration to our scheme in order to transmit a predetermined key K^* from A to B. Namely, the protocol would be initiated by B (instead of A), and in the second round, A would send $K^* + K$ to B along with **accept**.

Remark. In practice, we could take h to be a second preimage-resistant and one-way hash function. Another alternative is to let h be a semantically secure public-key cryptosystem with randomly chosen public key, in which case h would be injective.

5.1 Analysis

First, we show that if B accepts a key component L_i, then it was not altered by the adversary. Suppose that the adversary replaces L_i by a different value L_i'. Assuming that h is injective, we have that $h(L_i') \neq h(L_i)$. In order for the adversary to make B accept L_i' (in step 3(b)), he would have to change at least $p - p_a - 1$ of the $p-1$ values $h_{i,j}$ ($j \neq i$). But if the adversary controls p_i. then the adversary controls at most $p_a - 1$ of the other $p-1$ paths. We have $p_a - 1 < p - p_a - 1$ because $p \geq 2p_a + 1$. Therefore, in this situation, the scheme is perfectly reliable.

If h is not injective but it is second-preimage resistant, then a computationally-bounded adversary is unable to find L_i' such that $h(L_i') = h(L_i)$, even if such L_i' exist. The scheme is (computationally) reliable in this case.

It remains to evaluate the secrecy of the derived key K. First, we observe that if the adversary does not control the ith path, then he cannot determine any information about the key component L_i. This is because we are assuming that h is a random mapping. Now, let r denote the number of rejected key components; $r = p - a$. Any rejected key component lies on a path controlled by the adversary. Therefore the number of accepted key components that lie on paths controlled by the adversary is at most $p_a - r = n - (p - p_a)$. Now, the n-tuple $\mathbf{m}$ contains at most $n - (p - p_a)$ components that are known to the adversary. Hence, application of a $(n, p - p_a, n - (p - p_a), q)$-resilient function will yield a key whose entropy is $p - p_a$ symbols. This resilient function exists by Corollary 2.2.

Finally, A is also able to compute K because A is able to correctly determine the set **accept** after receiving p copies of it from B (at most p_a of these copies are altered by the adversary, so the correct **accept** can be determined by majority rule).

Let's next analyze the transmission overhead of the scheme. For the purpose of this analysis, assume that $|\mathcal{K}|$ is $\Theta(|\mathcal{T}|)$. Then the derived key has entropy $p - p_a$ symbols and the total transmission from A to B is $\Theta(p^2)$ symbols. The total transmission from B to A is p^2 bits (the set **accept** can be represented as a bitstring of length p). Since $q \geq p$, this is at most p symbols. So the total transmission is $\Theta(p^2)$ symbols, and the transmission overhead is at most
$$\Theta\left(\frac{p^2}{p - p_a}\right).$$
Since $p - p_a > p/2$, the transmission overhead is $\Theta(p)$.

Theorem 5.1: Suppose ℓ is a positive integer and $0 \leq e < 1/2$. Suppose that there are p disjoint paths from A to B, where $p_a = pe$ of these paths are controlled by the adversary. Then **Protocol 3** yields a $(p - p_a)$-secure secret key. The total transmission of **Protocol 3** consists of $\Theta(p^2)$ symbols, and the transmission overhead is $\Theta(p)$.

6 Conclusion

We proposed an enhanced security model to capture attacks against multi-path key establishment schemes in sensor networks. We identified two security objectives, which we term reliability and secrecy, that should be achieved. We observed that these objectives could be

realized using perfectly secure message transmission schemes.

Next, we proposed a new, optimal one-round PSMT scheme using Reed-Solomon codes, and we constructed two new multi-path key establishment schemes based on it. Both MPKE schemes achieve the desired objectives in an efficient manner. The second protocol potentially reduces the communication complexity in some cases by using feedback involving a message authentication code. Both of these protocols assume that the number of adversary-controlled paths is less than a $1/3$ fraction of the number of paths connecting A and B.

Finally, we described another MPKE scheme that tolerates a higher fraction (less than $1/2$) of paths controlled by the adversary. This scheme is based on a new protocol for a weakened version of message transmission, which is very simple and efficient.

REFERENCES

[1] C. H. Bennett, G. Brassard, and J.-M. Robert, "Privacy Amplification by Public Discussion," SIAM J. Comput., vol. 17, pp. 210–229, 1988.

[2] S. Çamtepe and B. Yener, "Combinatorial Design of Key Distribution Mechanisms for Wireless Sensor Networks," IEEE/ACM Trans. Netw., vol. 15, pp. 346–358, 2007.

[3] H. Chan, A. Perrig, and D. Song, "Random Key Predistribution Schemes for Sensor Networks," Proceedings of the 2003 IEEE Symposium on Security and Privacy (SP '03), pp. 197–213.

[4] B. Chor, O. Goldreich, J. Hasted, J. Friedman, S. Rudich, and R. Smolensky, "The Bit Extraction Problem or t-resilient Functions," Proceedings of the 26th Annual Symposium on Foundations of Computer Science (FOCS '84), pp. 396–407.

[5] J. Deng and Y. S. Han, "Multipath Key Establishment for Wireless Sensor Networks Using Just-enough Redundancy Transmission," IEEE Transactions on Dependable and Secure Computing, vol. 5, pp. 177–190, 2008.

[6] D. Dolev, C. Dwork, O. Waarts, and M. Yung, "Perfectly Secure Message Transmission," Journal of the ACM, vol. 40, pp. 17–47, 1993.

[7] W. Du, J. Deng, Y. Han, P. Varshney, J. Katz, and A. Khalili, "A Pairwise Key Pre-distribution Scheme for Wireless Sensor Networks," ACM Trans. Inf. Syst. Secur., vol. 8, pp. 228–258, 2005.

[8] L. Eschenauer and V. D. Gligor, "A Key-management Scheme for Distributed Sensor Networks," Proceedings of the 9th ACM Conference on Computer and Communications Security (CCS '02), pp. 41–47.

[9] M. Fitzi, M. Franklin, J. Garay, and S. H. Vardhan, "Towards Optimal and Efficient Perfectly Secure Message Transmission," Fourth Theory of Cryptography Conference (TCC 2007), Lecture Notes in Computer Science, vol. 4392, pp. 311–322, 2007.

[10] D. Huang and D. Medhi, "A Byzantine Resilient Multi-path Key Establishment Scheme and its Robustness Analysis for Sensor Networks," Proc. of 19th IEEE International Parallel and Distributed Processing Symposium (IPDPS '05), 2005.

[11] W. C. Huffman and V. Pless, *Fundamentals of Error-Correcting Codes*, Cambridge University Press, 2003.

[12] K. Kurosawa and K. Suzuki, "Almost Secure (1-round, n-channel) Message Transmission Scheme," Cryptology ePrint Archive, Report 2007/076, 2007.

[13] K. Kurosawa and K. Suzuki, "Truly Efficient 2-round Perfectly Secure Message Transmission Scheme," EUROCRYPT '08, Lecture Notes in Computer Science, vol. 4965, pp. 324–340, 2008.

[14] J. Lee and D. R. Stinson, "On the Construction of Practical Key Predistribution Schemes for Distributed Sensor Metworks Using Combinatorial Designs," ACM Trans. Inf. Syst. Secur., vol. 11, pp. 1–35, 2008.

[15] D. Liu, P. Ning, and R. Li, "Establishing Pairwise Keys in Distributed Sensor Networks," ACM Trans. Inf. Syst. Secur., vol. 8, pp. 41–77, 2005.

[16] A. Patra, A. Choudhary, K. Srinathan, and C. P. Rangan, "Unconditionally Reliable and Secure Message Transmission in Undirected Synchronous Networks: Possibility, Feasibility and Optimality," Cryptology ePrint Archive, Report 2008/141, 2008.

[17] R. Di Pietro, L. V. Mancini, and A. Mei, "Efficient and Resilient Key Discovery Based on Pseudo-Random Key Pre-Deployment," 18th International Parallel and Distributed Processing Symposium (IPDPS'04), pp. 217–224.

[18] I. S. Reed and X. Chen, *Error-Control Coding for Data Networks*, Kluwer Academic Publishers, Norwell, MA, USA, 1999.

[19] H. M. Sayeeda and H. Abu-Amara, "Efficient Perfectly Secure Message Transmission in Synchronous Networks," Information and Computation, vol. 126, pp. 53–61, 1996.

[20] A. Shamir, "How to Share a Secret," Commun. ACM, vol. 22, pp. 612–613, 1979.

[21] D. R. Stinson and J. L. Massey, "An Infinite Class of Counterexamples to a Conjecture Concerning Nonlinear Resilient Functions," Journal of Cryptology, vol. 8, pp. 167–173, 1995.

[22] Y. Wang, "Robust Key Establishment in Sensor Networks," SIGMOD Record, vol. 33, pp. 14–19, 2004.

[23] J. Wu and D. Stinson, "Minimum Node Degree and k-connectivity for Key Predistribution Schemes and Distributed Sensor Networks," ACM Conference on Wireless Network Security (WiSec '08), pp. 119–124, 2008.

[24] S. Zhu, S. Xu, S. Setia, and S. Jajodia, "Establishing Pairwise Keys for Secure Communication in Ad Hoc Networks: A Probabilistic Approach," IEEE International Conference on Network Protocols (ICNP '03), pp. 326–335, 2003.