Multi-path Key Establishment Against REM Attacks in Wireless Ad Hoc Networks

Tian Lan, Ruby Lee, and Mung Chiang
Department of Electrical Engineering, Princeton University, NJ 08544, USA
{tlan, rblee, chiangm }@princeton.edu

Abstract— **Secure communications in wireless ad hoc networks require setting up end-to-end secret keys for communicating node pairs. Due to physical limitations and scalability requirements, full key-connectivity can not be achieved by key pre-distribution. In this paper, we develop an analytical framework for the on-demand key establishment approach. We propose a novel security metric, called REM resilience vector to quantify the resilience of any key establishment schemes against Revealing, Erasure, and Modification (REM) attacks. Our analysis shows that previous key establishment schemes are vulnerable under REM attacks. Relying on the new security metric, we prove a universal bound on achievable REM resilience vectors for any on-demand key establishment scheme. This bound that characterizes the optimal security performance analytically is shown to be tight, as we propose a REM-resilient key establishment scheme which achieves any vector within this bound. In addition, we develop a class of low complexity key establishment schemes which achieve nearly-optimal REM-attack resilience.**

I. Introduction

In wireless ad hoc networks such as sensor networks, symmetric key cryptography is attractive due to its efficiency under extreme node resource constraints (e.g. computation, memory, and power). Currently, there exist three different approaches for providing pairwise secret keys: key assignment using trusted third parties, key pre-distribution before initial node deployment, and key establishment by exchanging keying messages. In particular, the key assignment schemes rely on trusted servers for key agreement among nodes [2], [3]. These schemes may not be practical for large-scale ad hoc networks, since the deployment of trusted servers or base-stations is uneconomical. The second approach, key pre-distribution, has attracted a lot of attention recently due to its efficiency in small or local networks. In key pre-distribution schemes [4], [9], [8], [5], a large amount of secret keys or keying information can be preloaded into nodes prior to deployment. Communicating nodes then discover shared keys after deployment to achieve a certain level of key-connectivity probability.

As pointed out in [13], [14], [15], key pre-distribution schemes have to struggle with the conflicts among node resource limits, desired key-connectivity probability, scalability in network size, and resilience against malicious attacks. Due to the limitation of node memory and computation ability, key pre-distribution schemes scale poorly to very large networks and the resulting pairwise key-connectivity probability is relatively low. In order to provide an end-to-end key to any communicating node pair, on-demand key establishment becomes a necessary approach. From a security perspective, most key pre-distribution schemes are designed to protect only the confidentiality of secret keys, while two other security components, integrity and availability, are not accounted for. Key pre-distribution schemes are vulnerable when various attacks occur simultaneously.

To address these issues, a key establishment approach that employs pre-distributed keys as local link keys has been proposed in [7], [9], [16]. In this approach, to set up an end-to-end secret key between two nodes, the source node generates a set of keying messages, from which a secret key can be derived. Each keying message is sent through a communication path from the source node to the destination node, which then computes the secret key locally. The transmission is protected by existing link keys at each hop. Since it is difficult to attack a large fraction of keying messages simultaneously in an ad hoc network, the key establishment approach using multi-path is able to guard against various attacks efficiently. In particular, an XOR-based key establishment scheme was proposed in [9], [16], where a secret key is derived by the XOR of all keying messages. This scheme prevents malicious attackers from deriving the secret key if not all keying messages are revealed. In [7], Huang et al proposed a Reed-Solomon code based scheme that allows node pairs to derive secret keys when both erasure and modification of keying messages occur. In a closely related problem known as secret sharing [10], it is shown that there exists a scheme to divide a secret into n messages in such a way that the key is easily reconstructable from any $r + 1$ pieces, but even complete knowledge of r pieces reveals no information about the secret. When applied to sensor networks, this technique enables the construction of a key establishment scheme that can guard against both revealing and erasure of keying messages.

However, all of these previous key establishment schemes only deal with a subset of the following three attacks, in which malicious nodes (i.e. compromised or fabricated nodes by attackers) can (a) Reveal the keying messages passing through them to make secret keys computable to the attackers; (b) Erase and not-forwarding keying messages to prevent other nodes from establishing secret keys; or (c) Modify the forwarded keying messages to prevent other nodes from deriving the correct secret keys. These attacks (defined as Reveal-Erase-Modify attacks in this paper) violate the three security properties, confidentiality, integrity, and availability of the keying messages, respectively. The problem is similar to the verifiable secret sharing [11], [12] in cryptography literature, where most existing algorithms rely on complicated algebraic operations, and thus are unsuitable for ad hoc network applications under computation constrain. The main contributions of this paper are as follows:

- We introduce a novel security metric, called REM resilience vector, to quantify the resilience of any key establishment schemes against REM attacks. The security performance of previous key establishment schemes [10], [7], [9], [16] are evaluated with respect to the proposed security metric. Our analysis and simulation show that previous key establishment schemes are vulnerable under REM attacks.
- We develop a unifying analytical framework, in which the entire set of REM resilience vectors achievable by any key establishment scheme is characterized by proving a security performance bound in a closed-form expression. The bound is tight in the sense that we propose an optimal key establishment algorithm to achieve any REM resilience vector within the bound.
- The 3-dimensional region, consisting of all feasible REM resilience vectors, models an important security tradeoff for the capability of defending against different types of attacks in the REM attack model. This tradeoff region can be used as a benchmark for the design and analysis of key establishment protocols given system parameters.
- We propose a class of low complexity key establishment algorithms with nearly-optimal REM-attack resilience. The algorithms only require XOR of keying messages and simple table lookups, as shown in our complexity analysis. We also implement the proposed algorithm in conjunction with the Zone Routing Protocol [18] and compare their performance with previous key establishment schemes. A significant security improvement is observed in large-scale simulations.

II. A New Security Metric for REM Attacks

We consider a wireless ad hoc network where nodes are not tamper resistant. Compromised or fabricated nodes may reveal all their forwarding keying messages to attackers and also try to disrupt normal key establishment in the network.

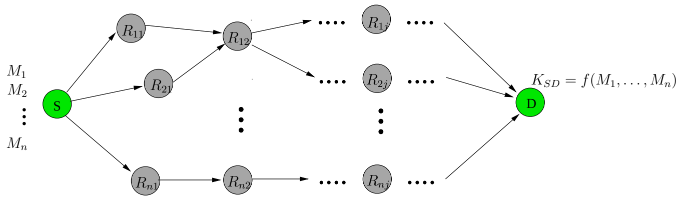

Fig. 1. This figure illustrates a general key establishment by sending n messages from the source node S to the destination node D.

In Fig 1, an end-to-end secret key is provided for nodes S and D, who do not share a common key from key pre-distribution. The procedure is described as follows: After receiving a request message, source node S first employs a network routing protocol and finds m paths (which can be non-disjoint) to the destination node D. n keying messages, denoted by $M_1, \dots, M_n$, are generated by the source node and then sent to the destination node, each via a different path, i.e. message M_i is send via path $(S, R_{i,1}, R_{i,2}, \dots, D)$. To secure keying messages during transmission, encryptions by existing link keys are performed at each intermediate node before forwarding keying messages, and nodes at the next hop decrypt the messages by the same link keys. More precisely, the following message is sent from node $R_{i,j}$ to node $R_{i,j+1}$:

$$R_{i,j} \to R_{i,j+1} : E[M_i, K_{i,j}^{i,j+1}]$$

where $E[\cdot]$ denotes the encryption function and $K_{i,j}^{i,j+1}$ is a link key from key pre-distribution. Upon receiving the keying messages, node D employs a function $f(\cdot)$ to reconstruct the secret key $K_{SD} = f(M_1, \dots, M_n)$ for future communication with node S. Since secret keys are set up on demand, the key establishment approach allows rekeying or key refreshing to be easily implemented in wireless ad hoc networks.

In this paper, we define a REM attack as any arbitrary combination of revealing, erasure, and modification attacks. Each type of attack targets at a different security property:

- *Revealing attacks on keying message confidentiality*: Compromised or fabricated nodes reveal to attackers the content of keying messages traveling through them. To quantify the resilience against this attack, we define a threshold value $r \geq 0$, such that if no more than r messages are revealed to attackers, resulting secret keys remain *completely unknown* even if all attackers collude.

  Definition 1: A secret key generated by a key establishment scheme with function $f(\cdot)$ is *completely unknown* under r revealed messages if

  $$\begin{aligned}&\text{Prob}\left\{f(M_1, \dots, M_n) = \hat{K} \middle| M_{i_1}, \dots, M_{i_r}\right\} \\ &= \text{Prob}\left\{f(M_1, \dots, M_n) = \hat{K}\right\}. \qquad (1)\end{aligned}$$

  for any $i_1, \dots, i_r$ and any choice of key $\hat{K}$.

  Definition 1 implies that revealing any set of no more than r keying messages does not change the original probability distribution of $\text{Prob}\{f(M_1, \dots, M_n)\}$. Thus, attacks obtain zero information by knowing r out of n keying messages.
- *Erasure attacks on keying message availability*: In an attempt to prevent the end-to-end secret key from being established, compromised or fabricated nodes make keying messages unavailable to the destination, by not-forwarding keying messages or jamming the forwarding link. We define $e \geq 0$ to be a threshold such that the secret key can be recovered at the destination node if no more than e messages are erased or dropped.
- *Modification attacks on keying message integrity*: Since complicated authentication methods (e.g. digital signature using public-key cryptography) are impractical in ad hoc networks, keying messages are subject to modification attacks, in which compromised or fabricated nodes forward modified keying messages to cause confusion. A threshold value $m \geq 0$ is chosen to denote the maximum

number of modified messages that can be corrected by a key establishment scheme.

Definition 2: An REM attack in wireless ad hoc networks is defined as any arbitrary combination of the revealing, erasure, and modification attacks, defined above.

Although erasure and modification attacks can also be regarded as transmission erasures and errors from a classical error control coding perspective, our REM attack model in this paper is different, because providing confidentiality (which is irrelevant to error control coding applications) jointly with integrity and availability is a must for establishing secret keys. In the following, we will provide a unifying framework and analysis for resilience of any key establishment scheme under REM attacks.

Given that n keying messages are used for establishing a secret key in a key establishment scheme, we quantify its REM-attack resilience by introducing a new security metric $(r, e, m)_n$ denoted as a REM resilience vector.

Definition 3: A key establishment scheme achieves REM resilience $(r, e, m)_n$, if a secret key can be successfully established under no more than e erasure attacks and m modification attacks, and at the same time, the key is completely unknown to attackers for up to r revealed keying messages.

For a key establishment scheme using n keying messages, the set of achievable REM resilience vectors lies in a 3-dimensional region, which illustrates security of the particular scheme along three axis: confidentiality, availability, and integrity (see Figure 2).

We can analyze the security of previous key establishment schemes under our REM framework. In [9], [16], secret keys of length k are derived at destination nodes by the bitwise XOR of all keying messages, each being exactly k bits, i.e. $K_{SD} = M_1 \oplus \ldots \oplus M_n$. It is easy to verified that a secret key remains completely unknown if not all keying messages are revealed to attackers. Thus, this scheme achieves REM resilience $(r = n - 1, e = 0, m = 0)_n$. In another scheme based on secret sharing [10], a secret key is regarded as an integer coefficient of a degree t random polynomial in GF_{2^k}, such that it can be recovered from any $t + 1$ evaluations of the polynomial and remains completely unknown if only t evaluations are given. Thus, it achieves $(r = t, e = n - t - 1, m = 0)_n$. By varying the degree t, we denote the set of achievable REM resilience vectors by $(r+e = n-1, m = 0)_n$.

Another scheme in [7] employs Reed-Solomon (RS) codes (a special class of error control codes) to deal with keying message erasures and modifications. Using a secret key as an input, keying messages are constructed by dividing the output codeword into n pieces. For an RS-code with block distance s, the key can be recovered if no more than e and m keying messages are erased and modified respectively, given that $2m + e \leq s - 1$, since each keying message is a linear combination of the secret key, revealing any keying message makes some choices of keys impossible. Consider a simple scheme with 3-bit secret keys $K_{SD} = [b_1 b_2 b_3]$ and a (7,4,3) binary code. If an attack obtains just one bit of the codeword $b_1 \oplus b_2 = 1$, it immediately derives that the secret key can not be $[00b_3]$ or $[11b_3]$. According to Definition 1, the secret key is not completely unknown to the attacker, and he can remove four possible keys from his entire search space. Thus, we have $r = 0$ for the RS code based scheme. Further, by extending this scheme to general non-binary error control codes, a REM resilience of $(r = 0, e+2m = n-1)_n$ can be achieved. Table I summarizes the security analysis of previous key establishment schemes, whose vulnerabilities under REM attacks (i.e. entries with zero resilience) are marked by * in the table.

Previous Schemes	Resilience vector $(r, e, m)_n$		
	r	e	m
XOR [9], [16]	$r = n - 1$	$e = 0^*$	$m = 0^*$
Polynomial [10]	$r + e = n - 1$		$m = 0^*$
RS code [7]	$r = 0^*$	$2m + e = n - 1$	

TABLE I

SECURITY ANALYSIS FOR PREVIOUS KEY ESTABLISHMENT SCHEMES UNDER REM ATTACKS. THIS SHOWS THAT THESE SCHEMES ARE DESIGNED TO DEAL WITH ONLY A SUBSET OF POSSIBLE ATTACKS.

III. PROVING OPTIMAL REM RESILIENCE

In this section, we analyze the optimal REM resilience for arbitrary key establishment schemes. For n paths and n keying messages, we show that no matter what keying-message construction and function $f(\cdot)$ are used, it is impossible to achieve any REM resilience vector with $r+e+2m > n-1$. This result states that $r + e + 2m \leq n - 1$ is a universal upper bound on achievable REM resilience vectors. The upper bound is also tight, as we find an optimal key establishment scheme which can achieve any REM resilience vector within this bound.

At first glance, it may appear that both optimality and achievability of bound $r + e + 2m \leq n - 1$ can be readily proved by encoding secret keys using an $(n, r + 1, s)$ linear error control code, since the keys are undecodable from r pieces of output codewords, and a direct application of the Hamming distance gives $2m + e \leq n - r - 1$. However, result in this section is much stronger and requires more interesting proofs.. First, our definition of security for key establishment requires secret keys to be completely unknown, not even partially decodable. Any piece of output codeword from a simple $(n, r + 1, s)$-encoding reveals certain linear constraints of the secret keys, and thus violates the desired security. Second, our upper bound $r + e + 2m \leq n - 1$ is applicable to any key establishment schemes with arbitrary keying-message construction and function $f(\cdot)$, while linear error control code is just one possible approach. The following analysis provides a fundamental limit for the security performance of key establishment, quantified by the proposed REM resilience vector. We state the first result in the following theorem, whose proof is given in Appendix A.

Theorem 1: Let each keying message be the same length as the secret key. For n paths and n keying messages, a REM resilience vector $(r, e, m)_n$ can be achieved if and only if $r + e + 2m \leq n - 1$.

Theorem 1 shows that for $n > 1$, the set of all achievable REM resilience vectors $(r, e, m)_n$ form a 3-dimensional tetrahedron $r + e + 2m \leq n - 1$ as shown in Fig 2, while previous key establishment schemes only explored certain 2-dimensional sub-planes in the tetrahedron: the polynomial based approach based on [10] achieves $\{r + e \leq n - 1, m = 0\}$, the Reed-Solomon code based approach in [7] achieves $\{r = 0, e + 2m \leq n - 1\}$, and the XOR based approach in [9] only achieves a single line $\{r \leq n - 1, e = 0, m = 0\}$. Theorem 1 for key establishment includes all previous results as lower-dimensional special cases.

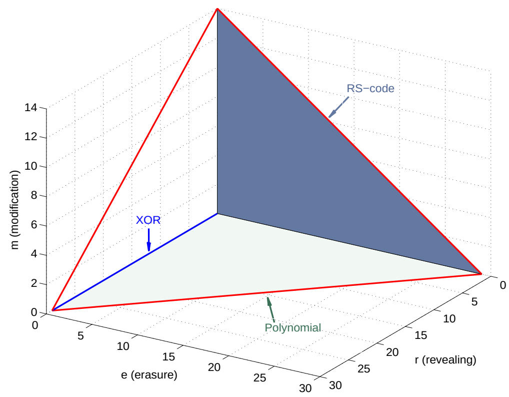

Fig. 2. For $n = 30$, this figure plots the 3-D optimal REM resilience region (i.e. the tetrahedron defined by $r + e + 2m \leq n - 1$) and 2-D sub-planes achieved by previous schemes.

Remark 1: When the length of keying messages is less than that of the secret key (i.e. length$(M_i) < k$, $\forall i$), it can be proven that a REM resilience $(r, e, m)_n$ can be achieved if and only if $r + e + 2m \leq n - \lceil \frac{k}{\text{length}(M_i)} \rceil$. This is a more general result than Theorem 1 and applies to all key establishment algorithms. Its proof is omitted due to space limitations.

The REM resilience vector's optimal bound proved in Theorem 1 provides a fundamental benchmark, from which many important security performance metrics can be derived directly. In ad hoc networks, a path is malicious if it contains at least one compromised or fabricated node. As the simplest case, if there is exactly one attack on each malicious path and the number of each type of attack is equal (i.e. $r = e = m$), then from $2m + e + r \leq n - 1$, it is easy to verify that a secret key can be established if less than three quarters of the paths are malicious, i.e. $r + e + m \leq \lfloor \frac{3(n-1)}{4} \rfloor$. However, the assumption of single attack on each path is impractical, because malicious nodes can collude to perform multiple attacks on one path and choose attack types to cause maximum damage. For such smart REM attacks, we derive the maximum resilience in the number of malicious paths as follows. We refer readers to [17] due to space limitation.

Corollary 1: Under smart REM attacks, a secret key can be established if and only if less than one third of paths are malicious, i.e. $\lfloor \frac{n-1}{3} \rfloor$.

IV. Low-Complexity Key Establishment Scheme

Theorem 1 characterizes the optimal REM resilience. However, according to the proof of Theorem 1, it requires multiplications of large integers in GF_p with $p > 2^k$ for constructing keying messages and a complicated sphere decoder to achieve REM resilience vectors on the optimal bound. The complexity is prohibitive for wireless ad hoc networks. In this section, we derive a class of low-complexity key establishment schemes that only requires bitwise XOR operations and simple table lookups. The new algorithm, generalized from linear binary error control codes, can achieve a nearly-optimal REM resilience. We first describe the proposed algorithm and then provide a security and complexity analysis.

A. Syndrome Decoding for Linear Binary Codes

A linear binary code $\mathcal{C}$ is a linear subspace of the field of binary vectors. If $\mathcal{C}$ is an (n, t, s)-code, then it encodes vectors of length t into codewords of length n, whose minimum Hamming distance is s. Let G of size $n \times t$ be a generating matrix for this linear code. Codewords are obtained by linear combinations of the rows of G, i.e. if $\vec{x}$ is a vector of length t, then $\vec{y} = G\vec{x}$ has length n and is the codeword for $\vec{x}$.

To correct both erasures and modifications in a received codeword, the following syndrome decoding procedure for binary linear codes can be employed: Let H be a parity check matrix for code $\mathcal{C}$. We first replace the erased coordinates by all zeros (denoted by $\vec{y^0}$) or all ones (denoted by $\vec{y^1}$) and compute two different syndromes (i.e. $\vec{r^0} = H^T\vec{y^0}$ and $\vec{r^1} = H^T\vec{y^1}$) respectively. By looking up $\vec{r^0}$ and $\vec{r^1}$ in the syndrome table to obtain two different error vectors $\vec{t^0}$ and $\vec{t^1}$, the one that contains fewer number of errors on non-erased coordinates gives us the correct syndrome that should be chosen. More precisely, if $\vec{r^0}$ (or $\vec{r^1}$ instead) gives fewer errors, then the original codeword can be recovered by inserting zeros (or ones) on the erased coordinates and then subtracting the error vector $\vec{t^0}$ (or $\vec{t^1}$) i.e. $\vec{y} = \vec{y^0} - \vec{t^0}$ (or $\vec{y} = \vec{y^1} - \vec{t^1}$). In classical coding theory, it has been proven that an (n, t, s)-code is able to correct any e erasures and m modifications at the same time, given that $2m + e \leq s - 1$ [1].

The following illustrative example contains a generating matrix and a parity check matrix for an $(8, 2, 5)$ linear binary code

$$G = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}^T,$$

$$H = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 \end{bmatrix}^T.$$

For an input vector $\vec{x} = [1\ 1]^T$, the corresponding codeword is given by $\vec{y} = G\vec{x} = [1\ 1\ 1\ 0\ 0\ 1\ 1\ 1]^T$. Now, suppose that the first two bits of $\vec{y}$ are erased and the third bit is flipped, i.e. the received vector becomes $\vec{y} = [* \ * \ 0 \ 0 \ 0 \ 1 \ 1 \ 1]^T$. We replace erased coordinates in $\vec{y}$ with ones and zeros respectively and compute two syndromes $\vec{r^0} = [0 \ 0 \ 0 \ 0 \ 0 \ 1]^T$ and $\vec{r^1} = [0 \ 1 \ 0 \ 0 \ 0 \ 1]^T$. By looking up the syndrome table for this $(8, 2, 5)$-code, we get $\vec{t^0} = [0 \ 0 \ 0 \ 1 \ 1 \ 0 \ 0 \ 0]$ and $\vec{t^1} = [0 \ 0 \ 1 \ 0 \ 0 \ 0 \ 0 \ 0]$. Since $\vec{t^0}$ contains two errors on non-erased coordinates, while $\vec{t^1}$ contains only one error, we choose all ones on the erased bits in $\vec{y}$ and subtract $\vec{t^1}$ from it. This gives us the correct codeword $\vec{y}$. In the next section, we generalize this syndrome decoding method and derive an algorithm for secure key establishment. The proposed algorithm not only corrects modifications and erasures, but also makes secret keys completely unknown to attackers.

B. Low-Complexity Key Establishment Algorithm

In this section, we propose a complete protocol for key establishment in wireless ad hoc networks. Our low-complexity algorithm relying on linear binary codes only requires bitwise XOR operations and simple table lookups for constructing keying messages and deriving the secret key. The protocol is divided into four phases: *(1) Request and Path-discovery*, *(2) Sending Keying Messages*, *(3) Recovering Key*, and *(4) Verification*. Packets transmitted in the protocol have the structure

ID_1	ID_2	Payload	CmdType

where ID_1 and ID_2 are the IDs of the source node and the destination node, respectively. In Phase 1, any standard ad hoc network routing, such as the Zone Routing Protocol [18], is employed to discover n paths, after receiving a request for key establishment. In Phase 2, a $(n+1, t, s)$ error control code is used to generate n keying messages. Let G be a generating matrix for the code

$$G = \begin{bmatrix} g_{01} & g_{02} & \cdots & g_{0t} \\ g_{11} & g_{12} & \cdots & g_{1t} \\ \vdots & \vdots & \ddots & \vdots \\ g_{n1} & g_{n2} & \cdots & g_{nt} \end{bmatrix}_{(n+1)\times t} \quad (2)$$

In order to add freshness to the algorithm, source node constructs t length-k pseudo-random vectors $X_1, \ldots, X_t$, and encodes each column of matrix $[X_1, \ldots, X_t]$ using G:

$$[K_{SD}, M_1, \ldots, M_n]^T = G \cdot [X_1, \ldots, X_t]^T \quad (3)$$

where the first row of the output codeword is chosen as a secret key and the $(i+1)$'th row as keying message $\mathbb{M}_i$ for $i = 1, \ldots, n$. Since linear binary codes are used, all operations required in this phase are simply binary XORs, denoted by $\oplus$.

Without loss of generality, we assume that the last e keying messages are unavailable to the destination node due to erasure attacks and the remaining $n - e$ keying messages contain m faulty ones due to modification attacks. Let H be a parity check matrix of size $(n+1) \times (n+1-t)$ for the generating matrix in (2). In Phase 3, the destination node implements a key-recovery algorithm based on the syndrome decoding for linear binary codes, as described in Section IV.A. Since the secret key K_{SD} is just the first row of the codeword in (3), the algorithm only needs to restore the first row of the codeword, rather than to decode all random vectors $X_1, \ldots, X_t$. In Phase 4, the secret key is verified between the source and destination node. Our protocol for establishing a secret key between two nodes S and D is summarized as follows:

Phase 1: *Request and Path-discovery*

1) Node D broadcasts a request for key establishment:

D:

D	S	Void	ReqKey

2) Node S responses to the request and starts a routing query for node D using the standard Zone Routing Protocol [18].
3) Node S recodes the first n replies to its routing query and prepares n paths to D:

$$(S, R_{i,1}, R_{i,2}, R_{i,3}, \ldots, D), \text{ for } i = 1, \ldots, n.$$

Phase 2: *Sending Keying Messages*

1) Node S constructs t length-k pseudo-random vectors $\mathbb{X}_1, \ldots, \mathbb{X}_t$.
2) The secret key is derived by

$$K_{SD} = (g_{01}X_1) \oplus (g_{02}X_2) \oplus \ldots \oplus (g_{0t}X_t).$$

3) Initialize $i = 1$.
4) Node S generates keying message $\mathbb{M}_i$:

$$\mathbb{M}_i = (g_{i1}X_1) \oplus (g_{i2}X_2) \oplus \ldots \oplus (g_{it}X_t).$$

5) Node S sends $\mathbb{M}_i$ to node $R_{i,1}$ and erases $\mathbb{M}_i$ locally

$S \rightarrow R_{i,1}$:

D	$R_{i,1}$	$E[M_i, K_s^{i,1}]$	EstKey

6) If $i < n$, let $i = i + 1$ and go to step 4.
7) Node S erases $X_1, \ldots, X_t$ from his memory.
8) Messages are forwarded to node D, for $i = 1, \ldots, n$:

$R_{i,1} \rightarrow R_{i,2}$:	$R_{i,1}$	$R_{i,2}$	$E[M_i, K_{i,1}^{i,2}]$	EstKey
$R_{i,2} \rightarrow R_{i,3}$:	$R_{i,2}$	$R_{i,3}$	$E[M_i, K_{i,2}^{i,3}]$	EstKey
⋮				
$R_{i,j} \rightarrow D$:	$R_{i,j}$	D	$E[M_i, K_{i,j}^{D}]$	EstKey

Phase 3: *Recovering Key*

1) Node D receives at least $n - e$ keying messages $\hat{M}_1, \ldots, \hat{M}_{n-e}$.
2) Define a mask vector A according to the indices of received keying messages: $A_1 = 0$ and

$$A_{i+1} = \begin{cases} 1, & \text{if } \hat{M}_i \text{ is received} \\ 0, & \text{otherwise} \end{cases} \quad \forall i = 1, \ldots, n.$$

3) Node D computes a submatrix $\tilde{H}$, consisting of the $n-e$ non-erased rows of H:

$$\tilde{H}_i = H_{i+1}, \text{ for } i = 1, \ldots, n-e.$$

4) Node D computes a syndrome perturbation vector $\tilde{r}$ as the XOR of the $e+1$ erased rows of H:

$$\tilde{r} = H_1 \oplus H_{n-e+2} \ldots \oplus H_{n+1}.$$

5) Node D computes $R^0 = \tilde{H}^T \cdot \left[\hat{\mathbb{M}}_1, \ldots, \hat{\mathbb{M}}_{n-e}\right]^T$.

6) Initialize $i = 1$. Let $ADDR$ be the base address of the syndrome table stored at the destination node.

7) Retrieve $\vec{t^0}$ from address $ADDR + R_i^0$.

8) Retrieve $\vec{t^1}$ from address $ADDR + (R_i^0 \oplus \tilde{r})$.

9) The i'th bit of K_{SD} is given by

$$K_{SD,i} = \begin{cases} \vec{t_1^0}, & \text{if } popcnt(\vec{t^0} \bigwedge A) < popcnt(\vec{t^1} \bigwedge A) \\ 1 \oplus \vec{t_1^1}, & \text{otherwise} \end{cases}$$

10) If $i < k$, let $i = i + 1$ and go to step 5.

Phase 4: *Verifying Key*

1) Node D generate a random message R and computes its hash value $h(R)$.

2) Node D broadcasts a challenge using secret key K_{SD}:

D:	D	S	$E[(R, h(R)), K_{SD}]$	GotKey

3) Node S decrypts $E[(R, h(R)), K_{SD}]]$ using its version of secret key K_{SD} and obtains $\hat{R}$.

4) Node S broadcasts an acknowledgement

S:	S	D	$\hat{R}$	ACK

5) Node D accepts K_{SD} if it receives $\hat{R} = R$.

In Step 5 of Phase 3 above, each row of $\left[\hat{M}_1, \ldots, \hat{M}_{n-e}\right]$ is a valid codeword generated by (2) with $e+1$ erasures and m modifications. According to the syndrome decoding procedure described in Section IV.A, if we assume that the erased keying messages are all zero vectors, we can compute a syndrome matrix $R^0 = \tilde{H}^T \cdot \left[\hat{M}_1, \ldots, \hat{M}_{n-e}\right]^T$, where each column of R^0 is a syndrome vector. On the other hand, if we assume that the erased keying messages are all one vectors, it is easy to show that the syndrome for the i'th row of $\left[\hat{M}_1, \ldots, \hat{M}_{n-e}\right]$ becomes $\tilde{r} \oplus R_i^0$, with $\tilde{r}$ as a perturbation vector defined in Step 4. Therefore, by looking up the syndrome table and comparing resulting error vectors, we can recover the first bit of the secret key, and thereafter bit by bit. In Step 9 of Phase *Recovering Key*, *popcnt* is a population count instruction which counts the number of "1" bits in a word.

C. Security Analysis

The proposed low-complexity key establishment algorithm is able to achieve nearly-optimal REM resilience vectors $(r, e, m)_n$ by choosing different linear error control codes. For n paths and n keying messages, we characterize the security performance of the algorithm in Theorem 2 as follows. We refer readers to [17] due to space limitation.

Theorem 2: For a linear binary error control code $(n + 1, t, s)$ with dual code $(n + 1, n + 1 - t, s')$, the proposed low-complexity key establishment algorithm in this section achieves a REM resilience vector $(r, e, m)_n$ for $r = s' - 2$ and $2m + e = s - 2$. In particular, when both codes are maximum distance separable (MDS), the proposed algorithm achieves an optimal REM resilience of $2m + e + r = s + s' - 4 = n - 3$.

D. Complexity Analysis

We analyze the complexity of the proposed key establishment algorithm in terms of computation overhead and storage space. For computation overhead, since we are restricted to linear binary codes in this paper, all operations are performed in Gf_2. We observe that the algorithm consists of four basic operations: binary XOR, table lookup, pseudo-random vectors, and assembly instruction (i.e. *popcnt* and comparison). For storage space, a syndrome table, generating and parity check matrices, and auxiliary vectors have to be stored at each node.

Complexity Metrics		Generating	Recovering
Computation	Bitwise XOR	$o(kn^2)$	$o(kn^2)$
	Random Vector	$o(n)$	-
	Table Lookup	-	$o(k)$
	popcnt and comparison	-	$o(k)$
Total Computation		$o(kn^2)$	$o(kn^2)$
Storage (bits)	Syndrome Table	-	$o(n2^{n-t})$
	Coding Matrices	$o(n^2)$	$o(n^2)$
	Auxiliary Vectors	$o(kn)$	$o(kn)$
Total Storage		$o(kn + n^2)$	$o(kn + n2^{n-t})$

TABLE II

COMPLEXITY ANALYSIS SUMARY FOR THE PROPOSED KEY ESTABLISHMENT ALGORITHM.

Table II summarizes the complexity of our proposed key establishment algorithm. As a numerical example, for a network using $n = 30$ keying messages (based on a (31,14,8) code) and an AES encryption with key size $k = 128$, the complexity is on the order of 200K operations and 4M bits of storage for generating keying messages and recovering a secret key. Our proposed algorithm, which is able to guard against all three attacks in the REM attack model, is much less complex than previous schemes [10][7], which require more than 20K 32-bit integer multiplications.

V. Numerical Examples

Consider a wireless ad hoc network with $Z = 1000$ nodes, uniformly distributed in a square area of size $L = 100$. We assume that nodes in the neighborhood of communication range $R = 15$ share pre-installed keys with probability p. These pre-installed link keys are used to secure keying messages during transmission. To discover n paths for each node pair, we implement the standard Zone Routing Protocol (ZRP) [18] with a zone radius of $\rho = 2$ hops. In all simulations, compromised nodes are randomly selected from the Z nodes such that the locations of compromised nodes are also uniformly distributed in the area.

We define the probability of secure and successful key establishment as the average probability that two nodes can successfully establish a secret key, and at the same time, the secret key remains completely unknown to attackers. For $p = 0.5$ and the optimal key establishment algorithm in Appendix A, Fig. 3 plots the probability of secure and successful key establishment for the use of $n = 1, 5, 10, 20, 30, 40$ keying

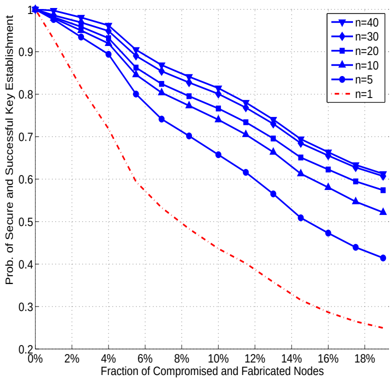

Fig. 3. Probability of secure and successful key establishment v.s. number of compromised nodes for $n = 1, 5, 10, 20, 30, 40$ keying messages.

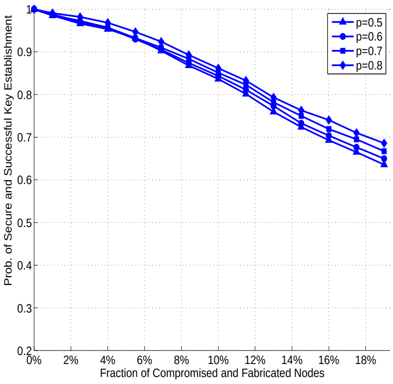

Fig. 4. Probability of secure and successful key establishment v.s. number of compromised nodes for pre-installed key-sharing probability $p = 0.5, 0.6, 0.7, 0.8$.

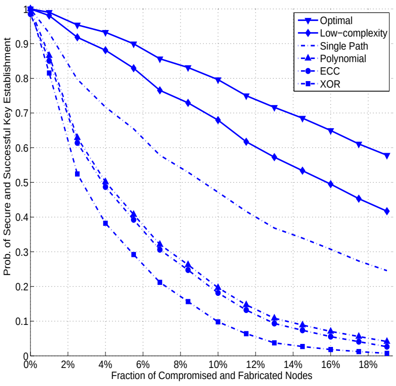

Fig. 5. Compare the security performance of key establishment schemes under our REM attacks.

messages, under REM attacks with equal probability of each type of attack. It can be observed that the optimal key establishment algorithm with $n \geq 20$ can safeguard secret keys with a probability of over 80% for as many as 80 (i.e. 8%) malicious nodes, and its security performance benefits from the increase of keying messages as more path diversity is exploited. In another simulation with $n = 30$, Fig. 4 shows that the probability of secure and successful key establishment remains almost the same for different pre-installed key-sharing probabilities $p = 0.5, 0.6, 0.7, 0.8$. This observation implies that no matter what key pre-distribution algorithm is used, the security performance achieved by the key establishment algorithm can be guaranteed. Thus, complicated key pre-distribution algorithms that are intended to provide high pre-installed key-sharing probability may not be necessary, since full key-connectivity can be achieved by our on-demand key establishment algorithm.

For the same network model with $p = 0.5$ and $n = 30$, we compare the security performance of different schemes: the optimal key establishment algorithm in Appendix A, the low-complexity key establishment algorithm in section IV, key establishment using single path, and the three previous multi-path key establishment schemes. Our low-complexity algorithm proposed in Section IV has a performance that is close to the optimal one, and is more suitable for practical implementations. This comparison highlights the importance of defending against multiple attacks simultaneously: Under REM attacks, the overall security performance of a key establishment algorithm is largely determined by the worst individual-attack resilience (i.e. $\min(r, e, m)$). It also demonstrates the excellent security-complexity properties of our proposed key establishment protocol.

VI. Conclusion

This paper proposes a unifying framework for analyzing the security of any key establishment scheme, quantified by a new metric we call a REM resilience vector. A universal bound on achievable REM resilience vectors is derived in closed-form and is shown to be attained by an optimal key establishment algorithm. For practical implementations, we also develop a low-complexity XOR-based key establishment protocol that achieves nearly-optimal REM resilience. Our analysis and simulation show that the capability of simultaneously defending against multiple attack classes, critical for the security of wireless ad hoc networks, can indeed be achieved with provable REM resilience and low complexity.

VII. Appendix

A. Proof of Theorem 1

Proof: The theorem states that the bound $r + e + 2m = n - 1$ is both optimal and tight. In the following, we start by showing the optimality and then propose a new key establishment scheme to prove the achievability.

To show $r + e + 2m = n - 1$ is optimal. If $e = m = 0$, then we immediately have $r \leq n - 1$, since the secret key becomes deterministic given all n keying messages. For $e + m > 0$, we denote $[M_1, \ldots, M_n]$ as a *feasible message vector*, in which $M_1, \ldots, M_n$ are a set of allowable keying messages that can be used to establish a secret key $K_{SD} = f(M_1, \ldots, M_n)$. Without loss of generality, we assume that the first r keying messages $M_1, \ldots, M_r$ are revealed to attackers who are able to collude. Then, with this information, the attackers can rule out any feasible message vector whose first r keying messages are not equal to $M_1, \ldots, M_r$. To guarantee that the secret key remains completely unknown, it is necessary that the number of remaining feasible message vectors with the first r messages in common must be no less than 2^k, i.e. the number of all possible secret keys of length k. Formally, if $H(\cdot)$ denotes the entropy function and feasible message vectors are random, we derive

$$\begin{aligned} & H([M_1, \ldots, M_n] | M_1, \ldots, M_r) \\ & \quad \geq H(f(M_1, \ldots, M_m) | M_1, \ldots, M_r) \\ & \quad = H(K_{SD} | M_1, \ldots, M_r) \\ & \quad = H(K_{SD}) = k \end{aligned} \tag{4}$$

where K_{SD} is the secret key. The second step is from the information processing inequality and the last step holds because all keys are equally likely due to the definition of completely unknown (1). Equation (4) implies that with the first r messages fixed, there exists at least 2^k feasible message vectors. These 2^k feasible message vectors are different only in the last $m-r$ messages, each of length k. Thus, the minimum Hamming distance of these feasible message vectors (i.e. the minimum number of different messages in any two feasible message vectors) can be no more than $m-r$. According to error control coding theory, given e erasures and m modifications, two feasible message vectors with a Hamming distance of $m-r$ remain distinct and separable only if

$$2m+e+1 \leq n-r \quad \Leftrightarrow \quad r+e+2m \leq n-1 \tag{5}$$

This gives the optimality of bound $r+e+2m \leq n-1$.

For achievability of the bound, we propose a new key establishment scheme that achieves any REM resilience vector $(r,e,m)_n$ satisfying the upper bound $r+e+2m+1=n$. The proposed algorithm for generating n keying messages is similar to the polynomial evaluation used in [10]. However, we employ a different decoding strategy and show that the algorithm can deal with revealing, erasure, and modification attacks at the same time. Let $p > 2^k$ be a prime number. Thus the desired secret key can be regarded as an integer in the field GF_p, i.e. $K_{SD} \in [0, 2^k-1]$. We generate a random degree r polynomial in GF_p as follows:

$$q(z) = K_{SD} + A_1 z + \ldots + A_r z^{rv} \tag{6}$$

where $A_i \in GF_p$ for $i = 1, \ldots, r$ are randomly chosen integers. Then n keying messages are computed by evaluating $q(x)$ at n distinct points for $z = 1, \ldots, n$, i.e.

$$[M_1, M_2, \ldots, M_n] = [q(1), q(2), \ldots, q(n)] \tag{7}$$

Since the polynomial has degree r, it has been shown in [10] that revealing no more than r keying messages would leave the secret key K_{SD} completely unknown. So we only need to show that the destination node can recover key K_{SD} under e erasures and m modifications, given that $2m+e = m-r-1$. Toward this end, we re-write equation (7) using a matrix representation:

$$\begin{bmatrix} M_1 \\ M_2 \\ \vdots \\ M_n \end{bmatrix} = \begin{bmatrix} 1 & 1^1 & 1^2 & \ldots & 1^r \\ 1 & 2^1 & 2^2 & \ldots & 2^r \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & m^1 & m^2 & \ldots & m^r \end{bmatrix} \cdot \begin{bmatrix} K_{SD} \\ A_1 \\ \vdots \\ A_r \end{bmatrix}$$

It is easy to verify that the $n \times (r+1)$ coefficient matrix (denoted by G) on the right hand side is a Vandermonde matrix, whose any $r+1$ rows are full rank. Thus, any non-zero vector $\vec{x}$ in $GF_p^{(r+1)}$ of size $1 \times (r+1)$ can be orthogonal to at most r rows of matrix G. We have

$$\forall \vec{x} \neq \vec{0}, \ \mathrm{Hamming}(G\vec{x}, \vec{0}) \geq n-r \tag{8}$$

where $\vec{0}$ is a zero vector and $\mathrm{Hamming}(\cdot)$ is the Hamming distance function. This implies that matrix G is a generating matrix for a $(n, r+1, s)$ linear error control code in GF_p with a minimum Hamming distance of at least $n-r$. According to error control coding theory, given that $2m+e+1 \leq n-r$, any m modifications and e erasures of the keying messages can be corrected at the destination node using a sphere decoding algorithm which finds the closest feasible message vector to the received one [1]. We summarize the optimal key establishment algorithm as follows:

Optimal Key Establishment Algorithm

1) Source node generates a random key K_{SD} and r random integers $A_1, \ldots, A_r$.
2) Source node generates $\mathbb{M}_i = K_{SD} + A_1 i + \ldots + A_r i^r$ and sends it to destination node, for $i = 1, \ldots, n$.
3) Destination node employs sphere decoding to derive K_{SD} upon receiving the keying messages.

This complete the proof of Theorem 1. ■

References

[1] S. Lin and D.J. Costello, "Error Control Coding: Fundamentals and Applications", *Prentice Hall*, 1983.

[2] B.C. Neuman and T. Tso, "Kerberos: An Authentication Service for Computer Networks," *IEEE Comm. Magazine*, vol. 32, no. 9, pp. 33-38, 1994.

[3] A. Perrig, R. Szewczyk, J.D. Tygar, V. Wen, and D.E. Culler, "SPINS: Security Protocols for Sensor Networks", *Wireless Networks*, vol. 8, no. 5, pp. 521-534, 2002.

[4] L. Eschenauer and D. Gligor, "A key-management scheme for distributed sensor networks", *Conference on Computer and Communications Security*, Washington, 2002.

[5] L. Zhou, J. Ni, and C.V. Ravishankar, Efficient Key Establishment for Group-Based Wireless Sensor Deployments, *in Proc. Fourth ACM Workshop Wireless Security*, pp. 1-10, 2005.

[6] W. Du , J. Deng , Y. S. Han , and P. K. Varshney, "A pairwise key pre-distribution scheme for wireless sensor networks", *Proceedings of the 10th ACM conference on Computer and communications security*, October 27-30, 2003, Washington D.C., USA

[7] D. Huang and D. Medhi, "A Byzantine resilient multi-path key establishment scheme and its robustness analysis for sensor networks", *19th IEEE International Parallel and Distributed Processing Symposium* , IPDPS 2005, Apr 4-8 2005. Vol 2005. (2005), Apr 7-9 2005. (2005) p. 177-183.

[8] D. Liu, P. Ning, and W. Du, "Group-Based Key Pre-Distribution in Wireless Sensor Networks", *in Proceedings of 2005 ACM Workshop on Wireless Security (WiSe 2005)*, September 2005.

[9] H. Chan, A. Perrig and D. Song, "Random Key Pre-distribution Schemes for Sensor Networks", *In 2003 IEEE Symposium on Research in Security and Privacy. pp197-213*, 2003.

[10] A. Shamir, "How to Share a Secret", *Communications of the ACM*, 1979.

[11] P. Feldman, "A practical scheme for non-interactive verifiable secret sharing", *IEEE Symposium on Foundations of Computer Science*, pp. 427-437, 1987.

[12] M. Fitzi, J. Garay, S. Gollakota, C. P. Rangan, and K. Srinathan, "Round-Optimal and Efficient Verifiable Secret Sharing". *Third Theory of Cryptography Conference*, 2006,

[13] E. Shi and A. Perrig, "Designing Secure Sensor Networks", *Wireless Comm. Magazine*, vol. 11, no. 6, pp. 38-43, Dec. 2004.

[14] W. Du, R. Wang, and P. Ning, "An Efficient Scheme for Authenticating Public Keys in Sensor Networks", *in Proc. ACM MobiHoc '05*, pp. 58-67, 2005.

[15] D. Xu, J. Huang, J. Dwoskin, M. Chiang, and R. Lee, "Re-examining Probablistic Versus Dtermininistic Key Management," *ISIT*, June 2007.

[16] S. Zhu, S. Xu, S. Setia, and S. Jajodia. "Establishing pair-wise keys for secure communication in ad hoc networks: A probabilistic approach", *In Proceedings of 11th IEEE International Conference on Network Protocols*, November 2003.

[17] T. Lan, R. Lee, and M. Chiang "Multi-path key establishment Against Byzantine Attacks in Wireless Ad hoc Networks", *Technical Report, Princeton University*, March, 2009.

[18] Z.J. Haas and M.R. Pearlman, "The Performance of Query Control Schemes for the Zone Routing Protocol", *IEEE/ACM Transactions on Networking*, vol. 9, no. 4, August, 2001.