Chapter 4

REMOTE UPLOAD OF EVIDENCE OVER MOBILE AD HOC NETWORKS

Indrajit Ray

Abstract In this work, we report on one aspect of an autonomous robot-based digital evidence acquisition system that we are developing. When forensic investigators operate within a hostile environment they may use remotely operated unmanned devices to gather digital evidence. These systems periodically upload the evidence to a remote central server using a mobile ad hoc network. In such cases, large pieces of information need to be fragmented and transmitted in an appropriate manner. To support proper forensic analysis, certain properties must ensured for each fragment of evidence – confidentiality during communication, authenticity and integrity of the data, and, most importantly, strong evidence of membership for fragments. This paper describes a framework to provide these properties for the robot-based evidence acquisition system under development.

Keywords: Evidence collection, authenticity, mobile ad hoc networks

1. Introduction

Consider the following scenario. A team of intelligence officers is trying to penetrate a terrorist network. The officers are working in a loosely coupled manner, geographically dispersed within hostile territory and far away from their command and control center. To limit their personal exposure to hostile activities and to enable evidence gathering from remote locations the team is using a number of unmanned vehicles (both aerial and terrestrial). These vehicles are network capable and equipped with various sensors (e.g., high resolution cameras, listening devices, chemical and bio-sensors) that gather information and periodically upload them to the central server at the command and control location. Later, forensic analysis is carried out on this data at the central location. Since the instruments need to be deployed in a fairly ad hoc manner at locations

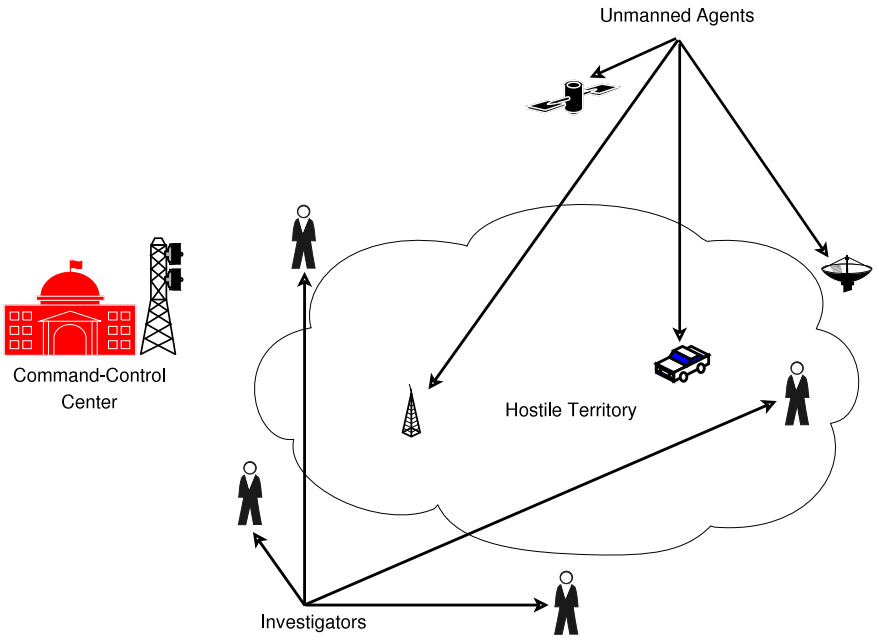

Figure 1. Target investigation environment.

that do not have access to a secure network infrastructure, they need to communicate securely with each other and with the central server over one or more mobile ad hoc networks (MANETs). A MANET can be deployed either solely over these devices or may use other mobile agents in the neighborhood. The instruments as well as the MANET(s) may be subject to active and passive attacks by adversaries. The environment is illustrated in Figure 1.

At Colorado State University, we are developing such an evidence acquisition system using mobile robots. Each mobile robot runs on a rechargeable battery and is equipped with a laptop computer that has wireless connectivity. The laptop computer runs the software that controls various sensors and actuators on the robot and provides support for ad hoc networking. Each robot is also equipped with a video camera (sensor) and a gripper arm (sensor/actuator) that can be remotely operated via the laptop. All the protocols described in this paper have been implemented and tested on a platform comprising three such robots.

Ensuring that the data uploaded by the unmanned devices is suitable for forensic analysis is not trivial. There are significant challenges imposed by MANETs that need to be addressed. First and foremost, authenticity and integrity of the data must be ensured during transmission. Since active attacks are possible, cryptographically strong schemes

must be provided for this purpose. Second, secure communication channels must be established between the various unmanned devices and the central server. The problem is that of ensuring secure group communication. The group, however, dynamically mutates, with the possibility that any of the communicating devices may be captured by the enemy and reverse engineered and/or destroyed. The group needs to be re-keyed periodically, ensuring that any protocol that is used for this purpose is efficient and is not compromised even when one or more of the devices are compromised. The third important challenge arises from the fact that MANETs are, in general, extremely unreliable, low bandwidth networks that provide no guarantees on message delivery. This means that to transmit large amounts of data, the data must be fragmented and there are no assurances that all fragments will be delivered. For the data to be useful as digital evidence, it is necessary to ensure that the fragments that have been received can be reconstructed to produce the original data, and to prove that each received fragment is part of the original data that was transmitted.

In this paper, we address these three issues in the following manner. We develop a protocol that allows us to deploy secure group communication over a MANET. This protocol is based partly on a theory of compatible keys that we have developed previously [9]. Next, we propose a scheme based on Rabin's Information Dispersal Algorithm (IDA) [8] that, together with conventional cryptographic checksums, allow us to reconstruct the original data from the fragments that have been received while ensuring authenticity and integrity. Finally, we describe a scheme based on one-way accumulator functions that allow us to prove the membership of fragments that are received over the MANET. Although some aspects of this problem have been solved individually, to our knowledge, this is the first scheme that provides a complete solution to the problem. We do not address the problem of routing and forwarding for MANETs as we assume that such a protocol already exists.

The rest of the paper is organized as follows. Section 2 addresses the issue of secure group communication over MANETs. Section 3 deals with the problem of preparing the evidence for reliable and authenticated upload over MANETs. In particular, Section 3.1 describes how a digest is prepared for the data and how the digest together with the data are fragmented using IDA. It also explains how a unique witness packet can be appended to each fragment to permit proofs of membership. Section 3.2 discusses the generation of fragment witnesses. Section 4 provides the conclusions.

2. Establishing Secure Group Communication

One of the biggest challenges for secure group communication is efficient group re-keying for dynamically mutating groups. Our group key protocol is designed to make re-keying simple and efficient. We begin by establishing the theory behind our group key protocol.

DEFINITION 1 The set of *messages* $\mathcal{M}$ is the set of non negative integers m that are less than an upper bound N, i.e.

$$\mathcal{M} = \{m \mid 0 \leq m < N\}. \tag{1}$$

DEFINITION 2 Given an integer a and a positive integer N, the following relationship holds,

$$a = qN + r \text{ where } 0 \leq r < N \text{ and } q = \lfloor a/N \rfloor \tag{2}$$

where $\lfloor x \rfloor$ denotes the largest integer less than equal to x. The value q is referred to as the *quotient* and r is referred to as the *remainder*. The remainder r, denoted $a \bmod N$, is also referred to as the *least positive residue* of $a \bmod N$.

DEFINITION 3 For positive integers a, b and N, a is *equivalent* to b, modulo N, denoted by $a \equiv b \bmod n$, if $a \bmod n = b \bmod n$.

DEFINITION 4 For positive integers a, x, n and $n > 1$, if $\gcd(a, n) = 1$ and $a.x \equiv 1 \bmod n$, then x is referred to as the *multiplicative inverse of a modulo n*. Two integers a, b are said to be *relatively prime* if their only common divisor is 1, that is, $\gcd(a, b) = 1$. The integers n_1,n_2,...,n_k are said to be *pairwise relatively prime*, if $\gcd(n_i, n_j) = 1$ for $i \neq j$.

DEFINITION 5 Euler's totient function $\phi(N)$ is defined as the number of integers that are less than N and relatively prime to N. Below we provide some relevant properties of totient functions.

1. $\phi(N) = N - 1$ if N is prime.
2. $\phi(N) = \phi(N_1)\phi(N_2)\ldots\phi(N_k)$ if $N = N_1N_2\ldots N_k$ and N_1, ..., N_k are pairwise relatively prime.

THEOREM 1 Euler's theorem states that for every a and N that are relatively prime, $a^{\phi(N)} \equiv 1 \bmod N$, where $\phi(N)$ is Euler's totient function.

Proof: We omit the proof of Euler's theorem and refer interested readers to a book on number theory (see, e.g., [6]).

COROLLARY 1 If $0 < m < N$ and $N = N_1 N_2 \ldots N_k$ and N_1,N_2,...,N_k are primes, then $m^{x\phi(N)+1} \equiv m \bmod N$ where x is an integer.

DEFINITION 6 A *key* K is defined to be the ordered pair $\langle e, N\rangle$, where N is a product of distinct primes, $N \geq M$ and e is relatively prime to $\phi(N)$; e is the *exponent* and N is the *base* of the key K.

DEFINITION 7 The *encryption* of a message m with the key $K = \langle e, N\rangle$, denoted as $[m,K]$, is defined as

$$[m, \langle e, N\rangle] = m^e \bmod N. \tag{3}$$

DEFINITION 8 The *inverse* of a key $K = \langle e, N\rangle$, denoted by K^{-1}, is an ordered pair $\langle d, N\rangle$, satisfying $ed \equiv 1 \bmod \phi(N)$, where $\phi(N)$ is Euler's totient function.

THEOREM 2 For any message m.

$$\left[\left[m, K\right], K^{-1}\right] = \left[\left[m, K^{-1}\right], K\right] = m \tag{4}$$

where $K = \langle e, N\rangle$ and $K^{-1} = \langle d, N\rangle$.

Proof: We first show that

$$\left[\left[m, K\right], K^{-1}\right] = m$$

$$\begin{aligned} L.H.S. &= \left[\left[m, K\right], K^{-1}\right] \\ &= \left[m^e \bmod N, K^{-1}\right] \\ &= (m^e \bmod N)^d \bmod N \\ &= m^{ed} \bmod N \\ &= m^{(x\phi(N)+1)} \bmod N \\ &= m \bmod N \\ &= m \\ &= R.H.S. \end{aligned}$$

By symmetry $\left[\left[m, K^{-1}\right], K\right] = m$.

COROLLARY 2 An encryption, $[m, K]$, is *one-to-one* if it satisfies the relation

$$\left[\left[m, K\right], K^{-1}\right] = \left[\left[m, K^{-1}\right], K\right] = m.$$

DEFINITION 9 Two keys $K_1 = \langle e_1, N_1\rangle$ and $K_2 = \langle e_2, N_2\rangle$ are said to be *compatible* if $e_1 = e_2$ and N_1 and N_2 are relatively prime.

DEFINITION 10 If two keys $K_1 = \langle e, N_1\rangle$ and $K_2 = \langle e, N_2\rangle$ are compatible, then the *product* key, $K_1 \times K_2$, is defined as $\langle e, N_1 N_2\rangle$.

LEMMA 1 For positive integers a, N_1 and N_2,

$$(a \mod N_1 N_2) \equiv a \mod N_1.$$

Proof: Let $a = N_1 N_2 x + N_1 y + z$, where x, y and z are integers.

$$\begin{aligned} L.H.S. &= (a \bmod N_1 N_2) \bmod N_1 \\ &= \left(N_1 N_2 x + N_1 y + z - \left\lfloor \frac{N_1 N_2 x + N_1 y + z}{N_1 N_2} \right\rfloor N_1 N_2 \right) \bmod N_1 \\ &= (N_1 y + z) \mod N_1 \\ &= z \end{aligned}$$

$$\begin{aligned} R.H.S. &= (a \mod N_1) \\ &= (N_1 N_2 x + N_1 y + z) \mod N_1 \\ &= z \end{aligned}$$

Hence the proof.

THEOREM 3 For any two messages m and $\hat{m}$, such that $m, \hat{m} < N_1, N_2$,

$$[m, K_1 \times K_2] \equiv [\hat{m}, K_1] \mod N_1 \text{ if and only if } m = \hat{m} \quad (5)$$

$$[m, K_1 \times K_2] \equiv [\hat{m}, K_2] \mod N_2 \text{ if and only if } m = \hat{m} \quad (6)$$

where K_1 is the key $\langle e, N_1 \rangle$, K_2 is the key $\langle e, N_2 \rangle$ and $K_1 \times K_2$ is the product key $\langle e, N_1 N_2 \rangle$.

Proof: The proof for (6) is similar to that for (5). Therefore, we only provide the proof for (5).
[If part] Given $m = \hat{m}$, prove that $[m, K_1 \times K_2] \equiv [\hat{m}, K_1] \bmod N_1$, that is,

$$[m, K_1 \times K_2] \mod N_1 = [\hat{m}, K_1] \mod N_1$$

$$\begin{aligned} L.H.S. &= [m, K_1 \times K_2] \mod N_1 \\ &= (m^e \mod N_1 N_2) \mod N_1 \\ &= m^e \mod N_1 \end{aligned}$$

$$\begin{aligned} R.H.S. &= [\hat{m}^e \mod N_1] \mod N_1 \\ &= \hat{m}^e \mod N_1 \\ &= m^e \bmod N_1 \end{aligned}$$

[Only If part] Given $[m, K_1 \times K_2] \equiv [\hat{m}, K_1] \mod N_1$, we have to prove $m = \hat{m}$

$$\begin{array}{lrcl} & [m, K_1 \times K_2] & \equiv & [\hat{m}, K_1] \mod N_1 \\ \text{or} & [m, K_1 \times K_2] \mod N_1 & = & [\hat{m}, K_1] \mod N_1 \\ \text{or} & (m^e \mod N_1 N_2) & = & (\hat{m}^e \mod N_1) \\ \text{or} & m^e \mod N_1 & = & (\hat{m}^e \mod N_1) \mod N_1 \\ \text{or} & [m, \langle e, N_1 \rangle] & = & [\hat{m}, \langle e, N_1 \rangle] \\ \text{or} & m & = & \hat{m} \end{array}$$

The more general case for the above theorem is as follows.

$$[m, K_1 \times K_2 \ldots K_p] \equiv [\hat{m}, K_1] \mod N_1 \text{ if and only if } m = \hat{m} \quad (7)$$

$$[m, K_1 \times K_2 \ldots K_p] \equiv [\hat{m}, K_1] \mod N_2 \text{ if and only if } m = \hat{m} \quad (8)$$

$$\vdots$$

$$[m, K_1 \times K_2 \ldots K_p] \equiv [\hat{m}, K_1] \mod N_p \text{ if and only if } m = \hat{m} \quad (9)$$

where $K_1 = \langle e, N_1 \rangle$, $K_2 = \langle e, N_2 \rangle$, etc. and $K_1 \times K_2 \ldots K_p$ is the product key $\langle e, N_1\, N_2\, \ldots, N_p \rangle$.

DEFINITION 11 The *group key* K_g for a group of k members $p_1, \ldots, p_k$, with a member p_i having the public/private key pair $K_i = \langle e, N_i \rangle / K_i^{-1} = \langle d_i, N_i \rangle$, is defined to be the product key $K_1 \times K_2 \times \ldots \times K_k$.

COROLLARY 3 A message $m < N_1 \times N_2 \ldots N_k$ encrypted with a group key $K_g = K_1 \times K_2 \times K_k$ can be decrypted with any one of the private keys K_i^{-1}.

Proof: The proof follows directly from Lemma 1 and Theorem 3.

2.1 Group Key Generation and Revocation

Group key management is usually done at the central server. This involves initial group key generation and periodic re-keying of the group. Re-keying to exclude a group member can also be performed at the local level. This can be done when one or more agents identify that one of their neighbors have been compromised. We employ re-keying by a local agent as one of the many ways to signal the command and control center that an agent has been compromised. In the following sections we discuss key generation and revocation in more detail.

Initial Group Key Generation: The group key is initially generated at the command and control center.

1. The center chooses a server key pair $\langle K_s, K_s^{-1} \rangle$ for itself and key pairs $\langle K_i, K_i^{-1} \rangle$ for the unmanned devices that will be deployed. These keys are compatible to each other according to Definition 9.
2. The group key is generated as $K_{gr} = K_s \times K_1 \times \ldots \times K_n$. Each unmanned device stores the key pair $\langle K_{gr}, K_i^{-1} \rangle$ The keys $K_s, K_s^{-1}, K_1 \ldots K_n, K_{gr}$ are stored at the central server.
3. All entities use the group key K_{gr} to encrypt messages. To decrypt a message, an unmanned device uses the key K_i^{-1} while the server uses K_s^{-1}. An agent uses K_i^{-1} to sign a message, while the server uses K_s^{-1}.

Group Re-Keying: The group key must be changed every time the group changes and it should also be changed periodically to limit exposure of the key. The group changes most often when an agent is compromised. An agent compromise is suspected by the central server if it does not receive an "alive" message from the agent for a period of time. At this stage, it takes one of two steps. It either determines unilaterally that the agent is compromised (it may have other information that confirms the suspicion) or it instructs other agents in the suspected agent's neighborhood to confirm the suspicion. The neighboring agents then execute a consensus algorithm among themselves to validate the central server's suspicion. Re-keying for the different scenarios takes place as follows.

1. To exclude an agent, let the agent which needs to be excluded from the group be agent i. The group key $K_{gr} = K_s \times K_1 \times \ldots K_i \times \ldots \times K_n$ is divided by K_i to generate the new group key as $K'_{gr} = K_s \times K_1 \times \ldots K_{i-1} \times K_{i+1} \times \ldots \times K_n$. The new group key is digitally signed by the server and sent to the remaining agents encrypted under itself. Thus, all the agents except the compromised one are able to obtain the new encryption key. (This will be true for every agent j as long as K_j is a factor of the group key.)
2. To include a new agent $n+1$ within the group, the central server creates the new group key $K'_{gr} = K_s \times \ldots K_n \times K_{n+1}$. This key is digitally signed by the server, encrypted under the new group key itself and forwarded to the remaining agents.

3. To perform periodic re-keying, the new group is generated according to the group key generation process. For each agent i a new key pair $\langle Q_i, Q_i^{-1} \rangle$ is generated. The relevant information is sent encrypted under the agent's previous key K_i.

2.2 Security and Implementation Issues

The group key system is as secure as the RSA cryptosystem. The most damaging attack on this scheme would be for an attacker to discover the private key K_i^{-1} corresponding to the key K_i involved in the group key K_g. This will enable the attacker to decrypt all messages encrypted with the group key. The obvious way to do this is to factor the modulus N_i of K_i into its two prime factors. This is a well known hard problem.

Another way to break the group key encryption is to find a technique to compute the e^{th} roots mod N. Since the encrypted message is $c = m^e \bmod N_1 \times N_2 \times \ldots \times N_k$ and $N = N_1 \times N_2 \times \ldots N_k$ is publicly known, the e^{th} root of c mod $N_1 \times N_2 \times \ldots \times N_k$ is the message m. This attack is not known to be equivalent to factoring. However, as pointed out in [10], there are no known techniques for performing this attack.

Care must be taken when choosing the exponent e. If e is small and $m^e < N_1 \times N_2 \times \ldots \times N_k$, then $c = m^e \bmod N_1 \times N_2 \times \ldots \times N_k = m^e$. Then, by simply extracting the e^{th} root of c an attacker can obtain the message m. Choosing a large e makes the encryption process computationally intensive. On the other hand, choosing an appropriate algorithm for the computation, alleviates the problem considerably. For example, the best time bound for multiplying two n-bit integers is $O(n \log n \log \log n)$ [3]. Finding $m^e \bmod p$, where p is an n-bit integer, is $O(\log e \; n \log n \log \log n)$.

3. Evidence Upload

Since a mobile ad hoc network is highly unreliable, an agent intending to upload a piece of evidence to the main server has to transmit the message in a redundant manner. To do this, the agent can transmit the entire evidence multiple times. However, this is expensive in terms of bandwidth and power consumption. Rabin's Information Dispersal Algorithm (IDA) [8] can be used to fragment data and transmit it efficiently. The fragmentation is performed so that if the original evidence has n fragments, the receipt of any m fragments at the central server ensures that all the evidence is received. Note that this scheme is cost effective as well as bandwidth efficient.

Data fragmentation raises an important issue: a strong proof of membership is required for all the fragments. In particular, it is necessary to

prove that every fragment that is received correctly belongs to the original message. The proof should be performed independently for each fragment. Consequently, this precludes the use of cryptographic techniques, e.g., Merkle hashes [5], where the hash (or signature) of a fragment is dependent on other fragments. In the following, we propose a novel scheme based on one-way accumulator functions to generate "witnesses" for each fragment. A witness serves as a proof of membership for a fragment. The algorithm for generating witnesses is described below.

3.1 Evidence Fragmentation

Let M be the message that has to be sent to the central server over the ad hoc network. In the following, h(X) denotes a digest of a message X created by, say, SHA-1 or MD5; $S_{kprv}(X)$ denotes a signature on message X using a private key $kprv$; and $E_{kpub}(X)$ denotes an encryption of message X with a public key $kpub$. The signature is verified using an encryption algorithm. The "witness" corresponding to a particular message i is denoted by w_i. We assume that the computations are done in GF(2^8). An agent intending to upload a message proceeds as follows.

1. The agent creates a message digest of the original message and digitally signs it with its private key $kprv$. The signed message digest is appended to the original message to get the new message $M' = M \| S_{kprv}(h(M))$. Let the length of the new message be N.

2. The agent breaks M' into $\frac{N}{m}$ fixed sized fragments of length m such as $M' = (b_1 \ldots b_m), (b_{m+1} \ldots b_{2m}), \ldots (b_{N-m+1} \ldots b_N)$ where b_i are integers taken from a certain range $[0 \ldots (2^r - 1)]$. That is, $M' = B_1 \| B_2 \| \ldots \| B_{\frac{N}{m}}$ where $B_i = (b_{i-1m+1} \ldots b_{im}), 1 \leq i \leq \frac{N}{m}$. The size of each fragment needs to be smaller than the MTU of the ad hoc network.

3. The agent chooses a set A of n vectors $A = (\vec{A_1}, \vec{A_2}, \ldots, \vec{A_n})$ such that each $\vec{A_i} = (a_{i1}, a_{i2}, \ldots, a_{im})$ and every subset of m different vectors in A are linearly independent.

4. M' is next processed and divided into n pieces $M'_1, M'_2, \ldots, M'_n$ such that $M'_i = (\vec{A_i} \bullet B_1, \ldots, \vec{A_i} \bullet B_{\frac{N}{m}}), i = 1 \ldots n$, where $\vec{A_i} \bullet B_k = (a_{i1} \cdot b_{(k-1)m+1} + \ldots + a_{im} \cdot b_{km})$.

5. For each $M_i\prime$ the agent prepares a witness w_i as described in Section 3.2. Each witness w_i is appended to the corresponding $M_i\prime$. The agent also generates a single accumulated witness W from the

individual witness w_i. Then, following a process similar to one described in Steps 2–4, the agent disperses the accumulated witness W over the M_i' fragments. Let $\bar{W}_i$ denote the portion of W that is dispersed over $M_i\prime$. After this step each fragment F_i is given by: $F_i = M_i' \| w_i \| \bar{W}_i$.

6. A digest of each F_i is prepared, digitally signed by the agent's private key $kprv$, and appended to fragment $F_i' = F_i \| S_{kprv}(h(F_i))$.

7. Finally, each F_i' is encrypted with the group key established earlier and transmitted over the MANET to the destination.

We can easily show that if any m fragments F_i' are received correctly by the central server (i.e., F_i is received correctly) then

1. the original message M' can be reconstructed, and

2. the accumulated witness W can be reconstructed.

We omit the proof for lack of space. Interested readers are directed to Rabin's work [8] for details. The ratio n/m is dependent on the estimated loss rate of the MANET. Since each fragment is accompanied by a strong cryptographic checksum that is digitally signed, the authenticity of the fragment is ensured in transit. Additionally, since the message M' can be reconstructed, it follows that the signed digest $S_{kprv}(h(M))$ is available to verify authenticity of message M.

3.2 Witness Generation

Let M_r' be the reconstructed message from the previous section. Each witness w_i together with the accumulated witness W are used to prove that each fragment M_i' is part of the same data from which the message M' has been reconstructed. We discuss how the different witnesses are created.

Witnesses are generated using one-way accumulator functions [2]. Briefly, a one-way accumulator is a special type of one-way hash function that satisfies the *quasi-commutative* property. A function $f : X \times Y \rightarrow X$ is said to be quasi-commutative if for all $x \in X$ and for all $y_1, y_2 \in Y$, $f(f(x, y_1), y_2) = f(f(x, y_2), y_1)$. An interesting consequence of this property is that if one starts with an initial value x and a set of values y_1, y_2, ..., y_n, then the accumulated hash $z = f(f(f(\ldots f(f(f(x, y_1), y_2), y_3), \ldots y_{n-2}), y_{n-1}), y_n)$ would be unchanged if the order of the y_i's were permuted.

Let us assume that such a one-way accumulator function f is available. Each M_i' that was obtained in the previous section after IDA was applied

to the message M' is processed as follows to generate the individual witness w_i and the accumulated witness W.

1. A unique seed value x_0 is defined globally and is available to every agent.

2. Using the one-way accumulator function f the agent calculates $x_i = f(x_{i-1}, M'_i)$ for the different M'_i. The final x_n is digitally signed by the agent and forms the accumulated witness W, i.e., $W = x_n \| S_{kprv}(x_n)$.

3. For each M'_i the agent computes the witness w_i as

$$w_i = f(f(\ldots f(f(x_0, M'_1), \ldots, M'_{i-1}), M'_{i+1}) \ldots, M'_n)$$

We can easily show that $x_n = f(w_i, M'_i)$. This proves that each received M'_i is a member of the original message from which M' was computed.

Benaloh and Mare were the first to propose the notion of one-way accumulator functions [2]. Subsequently, a number of researchers have proposed modifications to the original Benaloh-Mare proposal [1, 4, 7, 11, 12]. For the function f, we adopt an efficient implementation of the Benaloh-Mare scheme based on *universal-2* hash functions that has been proposed by Goodrich and co-workers [4].

4. Conclusions

The protocol presented in this paper permits digital evidence to be uploaded over mobile ad hoc networks in a secure and reliable manner. The data is transmitted in a manner that ensures its confidentiality, integrity and authenticity. The low bandwidth characteristics of a MANET necessitate data fragmentation. This, in turn, imposes the need for proofs of membership for the data fragments. A new group key protocol is devised to ensure secure communication over a MANET. Rabin's Information Dispersal Algorithm (IDA) is used to fragment data so that any m of n fragments allow the complete reconstruction of original data. One-way accumulator functions are used to generate proofs of membership for fragments. One component of the proof is dispersed over fragments to ensure that it will be always available when needed.

To our knowledge, this is the first integrated scheme that provides all the desired security and reliability properties. The protocol has been implemented as part of an on-going project focused on developing a robot-based digital evidence acquisition system. We hope to publish

other results from this project, including performance evaluation, in the near future.

References

[1] N. Baric and B. Pfitzmann, Collision-free accumulators and fail-stop signatures without trees, *Proceedings of the International Conference on the Theory and Application of Cryptographic Techniques (Lecture Notes in Computer Science, Volume 1233)*, Springer, Heidelberg, Germany, pp. 480-494, 1997.

[2] J. Benaloh and M. de Mare, One-way accumulators: A decentralized alternative to digital signatures, *Proceedings of the Workshop on the Theory and Application of Cryptographic Techniques (Lecture Notes in Computer Science, Volume 765)*, Springer, Heidelberg, Germany, pp. 274-285, 1993.

[3] T. Cormen, C. Leiserson, R. Rivest and C. Stein, *Introduction to Algorithms*, McGraw-Hill, Boston, Massachusetts, 2001.

[4] R. Gennaro, S. Halevi and T. Rabin, Secure hash-and-sign signatures without random oracle, *Proceedings of the International Conference on the Theory and Application of Cryptographic Techniques (Lecture Notes in Computer Science, Volume 1592)*, Springer, Heidelberg, Germany, pp. 123-139, 1999.

[5] R. Merkle, A certified digital signature, *Proceedings of the Ninth Annual International Cryptology Conference (Lecture Notes in Computer Science, Volume 435)*, Springer, Heidelberg, Germany, pp. 234-246, 1989.

[6] I. Niven and H. Zuckerman, *An Introduction to the Theory of Numbers*, John Wiley, New York, 1980.

[7] R. Nyberg, Fast accumulated hashing, *Proceedings of the Third International Workshop on Fast Software Encryption (Lecture Notes in Computer Science, Volume 1039)*, Springer, Heidelberg, Germany, pp. 83-87, 1996.

[8] M. Rabin, Efficient dispersal of information for security, load balancing and fault tolerance, *Journal of the ACM*, vol. 36(2), pp.335-348, 1989.

[9] I. Ray, E. Kim, R. McConnell and D. Massey, Reliably, securely and efficiently distributing electronic content using multicasting, *Proceedings of the Sixth International Conference on E-Commerce and Web Technologies (Lecture Notes in Computer Science, Volume 3590)*, Springer, Heidelberg, Germany, pp. 327-336, 2005.

[10] RSA Laboratories, RSA Laboratories' frequently asked questions about today's cryptography (v. 4.1) (www.rsasecurity.com/rsalabs), 2004.

[11] T. Sander, Efficient accumulators without trapdoors, *Proceedings of the Second International Conference on Information and Communications Security (Lecture Notes in Computer Science, Volume 1726)*, Springer, Heidelberg, Germany, pp. 252-262, 1999.

[12] T. Sander, A. T-Shma, and M. Yung, Blind auditable membership proofs, *Proceedings of the Fourth International Conference on Financial Cryptography (Lecture Notes in Computer Science, Volume 1962)*, Springer, Heidelberg, Germany, pp. 53-71, 2001.