Authenticating DSR Using a Novel Multisignature Scheme Based on Cubic LFSR Sequences

Saikat Chakrabarti[⋆], Santosh Chandrasekhar, Mukesh Singhal and Kenneth L. Calvert

Laboratory for Advanced Networking, Department of Computer Science,
University of Kentucky, Lexington KY 40506
email: {schak2,schan5,singhal}@cs.uky.edu, calvert@netlab.uky.edu

[Technical Report No. TR 474-07, Department of Computer Science, University of Kentucky]

Abstract. The problem of secure routing in mobile ad hoc networks is long-standing and has been extensively studied by researchers. Recently, techniques of aggregating signatures have been applied to authenticate on demand routing protocols in mobile ad hoc networks. In this paper, we propose an efficient, single round multisignature scheme, CLFSR-M, constructed using cubic (third-order) linear feedback shift register (LFSR) sequences. The scheme, CLFSR-M is derived from a 2-party signature scheme CLFSR-S, formed using a well-known variant of the generalized ElGamal signature scheme. The multisignature has been engineered to produce an efficient technique to authenticate route discovery in the dynamic source routing (DSR) protocol. Our technique supports authentication of cached routes. Delegating special functions to nodes or assuming the existence of a trusted third party to distribute certified public keys is not practical in mobile ad hoc networks. We consider a fully distributed mechanism of public key distribution and present two variations of trust policies, based on PGP, for effective management of individual and aggregate public keys. Finally, we perform a theoretical analysis including correctness and security of CLFSR-M and also present a performance (computation and communication costs, storage overhead) comparison of the proposed scheme with existing ones.

Keywords: Secure routing, DSR, multisignatures, generalized ElGamal signatures, LFSR-based PKCs, PGP, small-world graphs.

1 Introduction

Designing secure routing protocols for mobile ad hoc networks is a challenging task. Resource constraints of nodes, limited capacity of the wireless medium,

---

[⋆] The corresponding author would like to thank Levente Buttyán for helpful comments while presenting the paper at ESAS 2007, Cambridge, UK.

node mobility and the cooperative, self-organized form of the network make it difficult to transfer techniques for securing traditional wired networks to the ad hoc networking environment. The dynamic source routing protocol (DSR) is perhaps the most popular on-demand source routing protocol designed for multi-hop wireless ad hoc networks [1]. DSR is simple and efficient in construction, offers loop-free routing guarantees and load balancing, uses only soft-state, and is robust [1]. However, the original construction of DSR does not consider an adversarial model of the underlying network. Thus, DSR is vulnerable to several forms of attack by malicious nodes such as injection of bogus routing information and formation of feedback loops by colluding adversarial nodes [2, 3].

This paper focusses on the following problem: how can a source node wanting to find a route to a destination be assured of the authenticity of the source route advertised in a received routing packet? We would like to guarantee this authenticity without imposing substantial overhead on the nodes that help in discovering routes. We propose an efficient, single round multisignature scheme (aggregate signature on the same message) to authenticate route discovery information in DSR. A number of enhancements and optimizations have been proposed for DSR so far; use of cached routes being one of the most significant ones made. Our scheme also works with *path caching* enabled [4].

Our multisignature scheme is derived from a cubic LFSR-based 2-party signature scheme [5], which uses a well-known variant of the generalized ElGamal signature scheme, EG I.4 [6]. Our scheme is efficient, requiring no prior cooperation to construct the multisignature. The efficiency of the proposed signatures schemes can be partially attributed to the use of LFSR sequence-based public key cryptosystems, which employ reduced representations of finite field elements [7]. The security of LFSR-based PKCs is based on the difficulty of solving the discrete logarithm problem in the extension field $\mathbb{F}_{q^n}$ (contains q^n elements). However, all computations involving sequence terms are performed in the base field $\mathbb{F}_q$ (contains q elements). This leads to substantial savings, both in communication and computation overhead, for a desired security level.

We first present the basic idea behind authenticating routes in DSR assuming, for simplicity, all nodes have access to certified public keys of other nodes in the route. Distributing authentic public keys among nodes in a mobile ad hoc network to bootstrap authentication protocols is a challenging task. We discuss solutions using a trusted third party (TTP) to help in distributing certified public keys. However, assuming the existence of a TTP is paradigmatically unsuitable for ad hoc networks. Using the concepts of PGP [8] and previous results of the *small-world* property [9] exhibited in trust graphs in self-organized systems [10, 11], we relax the assumption of the TTP and formulate policies for a fully distributed framework for individual and aggregate public key management.

The rest of the paper is organized as follows. We discuss related work in Section 2. In Section 3.1, we describe mathematical preliminaries of cubic LFSR sequences related PKCs and provide a short description of DSR. In Section 4, we describe techniques for authenticating route discovery in DSR based on multisignatures. We present a novel multisignature scheme in Section 5. In Section 6, we discuss a fully distributed mechanism of public key management. We provide a theoretical analysis of the proposed multisignature scheme in Section 7. Section 8 concludes the paper.

2 Related Work

The original design of DSR [1] did not incorporate any security mechanism, making it vulnerable to several attacks [2]. Papadimitratos et al. [12] and Hu et al. [13] independently proposed secure on-demand routing protocols, SRP and Ariadne, respectively, to authenticate routes using message authentication codes (MACs). In SRP, intermediate nodes in the route are not authenticated, thus exposing SRP to attacks, including addition and deletion of honest nodes from the route. In Ariadne, route request packets grow in size due to accumulation of MACs. Ariadne also requires loose time synchronization. Kim et al. [3] presented a generic DSR authentication protocol, SRDP, using MACs and aggregate signature schemes of [14–16]. SRDP does not consider authentication of routes using cached information. Moreover, the signature based variants of SRDP have performance drawbacks, discussed in Section 7.3. Acs et al. [17] proposed a mathematical framework to facilitate the analysis of secure on-demand source routing protocols in mobile ad hoc networks. Acs et al. present attacks on the well-known Ariadne protocol and also describe the construction of a protocol, endairA, which is provably secure in the proposed model of security. Bhaskar et al. [18] developed a MAC based aggregate designated verifier signature scheme for authenticating DSR. The MAC-based scheme in [18] cannot authenticate cached routes. Moreover, MAC-based authentication protocols do not offer non-repudiation. Also, in MAC-based schemes, early detection of invalid MACs by intermediate nodes requires additional key setup overhead.

The concept of a multisignature was first proposed by Itakura et al. [19]. Horster et al. [6] proposed a generalized ElGamal signature scheme [20], integrating several ElGamal variants, including Schnorr's signature [21] and the DSA. Micali et al. [14] formalized the concept of multisignatures and proposed a three round multisignature scheme based on the Schnorr variant [21]. Multisignatures are a specialized form of aggregate signatures — Boneh et al. [15] first proposed the concept of a generalized aggregate signature scheme using efficiently computable bilinear maps. Lysyanskaya et al. [16] proposed sequential constructions of aggregate signatures using families of certified trapdoor permutations.

Capkun et al. [10] analyzed PGP trust graphs and showed that such graphs exhibited the small world phenomenon [9, 22]. Kleinberg [22] gave an algorithmic perspective to the small-world phenomenon. Capkun et al. [11] also proposed a PGP-like, self-organized public key management system for ad hoc networks.

Recently, new PKCs have emerged based on LFSR sequences under the Trace-DLP [7] assumption. The first PKC based on LFSR sequences was introduced by Niederreiter [23]. Gong et al. [24, 25] and Lenstra et al. [26] independently proposed the GH-PKC and the XTR-PKC, respectively, using cubic LFSR sequences. We omit a discussion on higher order LFSR sequence-based PKCs in this paper, due to space constraints.

3 Background

We present a brief discussion of the mathematics underlying cubic LFSR sequences and PKCs constructed using cubic LFSR sequences. We also provide a short note on DSR, including possible optimizations of DSR.

3.1 Cubic LFSR Sequences and Related Public Key Cryptosystems

We provide a brief discussion of the cryptographic preliminaries needed to understand the construction of the proposed signature schemes based on cubic LFSR sequences.

A sequence of elements $\{s_k\} = s_0, s_1, \ldots$ over the finite field $\mathbb{F}_q$ is called a 3rd order homogeneous linear recurring sequence in $\mathbb{F}_q$ if for all $k \geq 0$:

$$s_{k+3} = c_0 s_{k+2} + c_1 s_{k+1} + c_2 s_k \tag{1}$$

where, $c_0, c_1, c_2 \in \mathbb{F}_q$ and s_k denotes the kth term of the sequence $\{s_k\}$. Such sequences can be efficiently generated by a special kind of electronic switching circuit, called LFSR. Consider the following monic irreducible polynomial over $\mathbb{F}_q$: $f(x) = x^3 - ax^2 + bx - 1$, where $a, b \in \mathbb{F}_q$. The sequence $\{s_k\}$ is said to be a cubic-LFSR sequence generated by $f(x)$ if we have $c_0 = a$, $c_1 = b$ and $c_2 = 1$ in Equation 1, i.e., for all $k \geq 0$: $s_{k+3} = as_{k+2} - bs_{k+1} + s_k$. The polynomial $f(x)$ is called the *characteristic polynomial* of the sequence $\{s_k\}$ if, given a root α of $f(x)$, for all $k \geq 0$, we have $s_k = \alpha^k + \alpha^{kq} + \alpha^{kq^2}$, where $\alpha \in \mathbb{F}_{q^3}$. The sequence $\{s_k\}$ is called the third-order *characteristic sequence* generated by $f(x)$ (or by α). The initial state (kth state denoted as $\bar{s}_k = \{s_k, s_{k+1}, s_{k+2}\}$) of the characteristic sequence of $f(x)$ is given by $\bar{s}_0 = \{3, a, a^2 - 2b\}$ [7].

Recently, two PKCs, namely, GH-PKC [24] and XTR-PKC [26] were proposed based on cubic LFSR sequences [27]. In cubic LFSR-based PKCs [24, 26], elements in $\mathbb{F}_{q^3}$ are represented by their corresponding minimal polynomials whose coefficients are chosen from $\mathbb{F}_q$. However, the security of cubic LFSR-based PKCs is based on the difficulty of solving the discrete logarithm problem in $\mathbb{F}_{q^3}$. This leads to substantial savings, both in communication and computational overhead, for a desired security level. In particular, 170-bits of XTR-PKC gives security equivalent to 1024-bits of cryptosystems using traditional representation of finite fields [26]. The XTR cryptosystem is constructed by choosing:

1. p, a large prime of the order of 170 bits. Set $q = p^2$.
2. Q, a large prime factor of $p^2 - p + 1$ of the order of 160 bits.
3. Characteristic polynomial $f(x) = x^3 - ax^2 + a^p x - 1$ with period Q by randomly choosing $a \in \mathbb{F}_q$ and using standard irreducibility testing algorithms.

Let $f_k(x)$ denote the minimal polynomial of α^k where $\alpha \in \mathbb{F}_{q^3}$ is a root of $f(x)$. It can be shown that the polynomial $f_k(x)$ can be represented as [7, 24, 26]: $f_k(x) = x^3 - s_k x^2 + s_k^p x - 1$ in the XTR-PKC. Thus, the polynomial f_k (we drop the indeterminate x for simplicity of notation) can be represented by $s_k \in \mathbb{F}_q$ in XTR. The sequence terms are computed using the following two sequence operations [25]:

1. OP_1: given an integer k and f_e, compute the (ke)th state of the LFSR, $\bar{s}_{ke}$.
2. OP_2: given $\bar{s}_k$ and $\bar{s}_e$ (both integers k and e need not be known), compute the $(k+e)$th state of the LFSR, $\bar{s}_{k+e}$.

These sequence operations have been efficiently implemented in hardware [28]. We use the sequence operations to create/manipulate sequence terms in the proposed multisignature scheme.

In cubic-LFSR based PKCs, an entity randomly chooses a *long-term private key* $\mathsf{SK} = x$ in $\mathbb{Z}_Q^*$ and computes the *long-term public key* $\mathsf{PK} = \bar{s}_x = \{s_x, s_{x+1}, s_{x+2}\}$ using the sequence operation $\mathsf{OP}_1(x, f)$. Algorithms for sequence term computations use the following *commutative law* [24] for characteristic sequences: for all integers r and e, the r^{th} term of the characteristic sequence generated by the polynomial $f_e(x)$ equals the $(re)^{\text{th}}$ term of the characteristic sequence generated by the polynomial $f(x)$, i.e., $s_r(f_e) = s_{re}(f) = s_e(f_r)$.

Throughout the paper, we construct our signature schemes using the XTR-PKC for simplicity, although the proposed schemes can be seamlessly built using the GH-PKC and also extended to PKCs based on higher order LFSR sequences, with minor modifications.

3.2 A Short Note on DSR

DSR is composed of two central mechanisms, namely, route discovery and route maintenance. In this paper we focus on DSR's route discovery mechanism. The source initiates route discovery by generating an RREQ (route request) packet and broadcasting it to all its neighbors. The RREQ packet contains a field indicating the destination and a source route field intended to accumulate the desired route. Each node that is not the destination and has not encountered the RREQ packet previously appends its IP address to the source route contained in the packet and re-broadcasts the packet to its own neighbors. RREQ propagation continues until the destination is encountered. When the destination receives the RREQ packet, it generates the route reply (RREP) packet containing the accumulated (source) route and unicasts the RREP to the initiator of the route discovery along the reverse path of the source route.

DSR is an on-demand routing protocol and thus, attempts to discover a route to a destination node only when a source originates a data packet addressed to that node. To avoid initiating route discovery before each data packet is sent, the source needs to cache routes [4]. The RREP packet at all times contains a complete sequence of links leading to the destination. Intermediate nodes forwarding the RREP packets can (optionally) accumulate these complete paths into *path* *caches* so they can efficiently reply to route requests at a later time. Path caches are simple to implement and also guarantee that all routes are loop-free, since all source routes contained in the RREP are loop-free themselves. This mechanism of caching is one of the most important enhancements made to DSR.

4 Authenticating Route Discovery in DSR

As with most routing protocols, the original construction of DSR did not consider an adversarial model of the underlying network. As a result, DSR is vulnerable to several forms of attacks by malicious nodes, including injection of bogus routing information and formation of feedback loops by colluding adversarial nodes [3]. The classical approach to mitigating such attacks is to use cryptographic tools to authenticate information exchanged during the route discovery process. In this section, we present techniques for authenticating route discovery in DSR (with and without path caching) based on multisignatures.

4.1 First Construction

A first intuition for authenticating route discovery in DSR would be to have each node sign RREQ packets as they are forwarded toward the destination, so that the destination could authenticate the accumulated source route before generating an RREP packet. However, due to flooding of RREQ packets in the DSR route discovery algorithm, several nodes would end up wasting computation and communication resources by signing, verifying and forwarding RREQ packets if these nodes are not included in the eventual route. Also, in this mechanism, if authentication is done by combining signatures on different messages, a sequential aggregate signature must be used; such signatures are usually computationally more expensive than the more specific form of multisignatures. In our technique, we authenticate the source route contained in the RREP packets using an efficient, single round multisignature scheme, requiring no prior cooperation among nodes to construct the signature.[1]

We first present the basic idea of authenticating the route discovery process in DSR without considering caching of routes. Let nodes $\{d_0, \ldots, d_k, \ldots, d_n\}$ constitute a source route. An arbitrary node and its IP address are denoted by the same notation, d_k, for simplicity. First, let us assume that an arbitrary node d_k has authentic copies of public keys $PK_{k+1}, \ldots, PK_n$ of all nodes leading to the destination. Fig. 1 shows the propagation of authenticated RREP packets from the destination d_n to the source d_0. Node d_k does the following: (1) combines the public keys to form aggregate public key $PK_{(k+1,n)}$, (2) verifies multisignature $\sigma_{(k+1,n)}$ that it receives from node d_{k+1}, (3) signs the hashed concatenation

---

[1] Authenticating source routes contained in RREP packets was also independently proposed by Acs et al. [17]. However, the primary goal of Acs et al. was to present a framework of security for analysing secure routing protocols in mobile ad hoc networks, and not propose a cryptographic construction of a suitable multisignature, which is the essential goal of this paper.

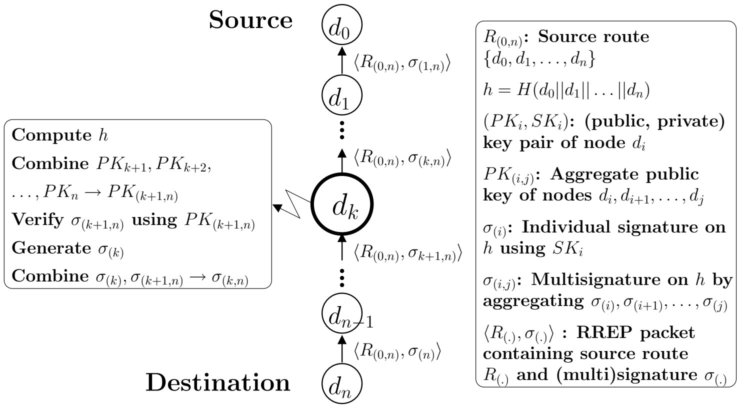
Source
d_0
$\langle R_{(0,n)}, \sigma_{(1,n)} \rangle$
d_1
$\langle R_{(0,n)}, \sigma_{(k,n)} \rangle$
d_k
$\langle R_{(0,n)}, \sigma_{k+1,n)} \rangle$
d_{n-1}
$\langle R_{(0,n)}, \sigma_{(n)} \rangle$
Destination
d_n
Compute h
Combine $PK_{k+1}, PK_{k+2}, \ldots, PK_n \rightarrow PK_{(k+1,n)}$
Verify $\sigma_{(k+1,n)}$ using $PK_{(k+1,n)}$
Generate $\sigma_{(k)}$
Combine $\sigma_{(k)}, \sigma_{(k+1,n)} \rightarrow \sigma_{(k,n)}$
$R_{(0,n)}$: Source route $\{d_0, d_1, \ldots, d_n\}$
$h = H(d_0||d_1|| \ldots ||d_n)$
(PK_i, SK_i): (public, private) key pair of node d_i
$PK_{(i,j)}$: Aggregate public key of nodes $d_i, d_{i+1}, \ldots, d_j$
$\sigma_{(i)}$: Individual signature on h using SK_i
$\sigma_{(i,j)}$: Multisignature on h by aggregating $\sigma_{(i)}, \sigma_{(i+1)}, \ldots, \sigma_{(j)}$
$\langle R_{(.)}, \sigma_{(.)} \rangle$: RREP packet containing source route $R_{(.)}$ and (multi)signature $\sigma_{(.)}$

Fig. 1. Propagation and authentication of RREP

of the IP addresses contained in source route (all nodes sign this message) to create $\sigma_{(k)}$ if the verification in Step (2) is successful, (4) combines $\sigma_{(k+1,n)}$ and $\sigma_{(k)}$ to form multisignature $\sigma_{(k,n)}$, (5) removes $\sigma_{(k+1,n)}$ from and appends $\sigma_{(k,n)}$ to the RREP packet and (6) sends the RREP packet to the next node d_{k-1}. At the source d_0, successful verification of multisignature $\sigma_{(1,n)}$ under the aggregate public key $PK_{(1,n)}$ establishes the authenticity of all signatures on the source route. Note that signature verification by intermediate nodes facilitates early detection of bogus routes injected by an adversary. The procedures for combining public keys, generation, verification and aggregation of signatures are presented in Section 5.

4.2 Incorporating Path Caching

Now, we extend the above technique to incorporate path caching[2]. Consider the case where source d_0 has already established a route to destination d_n as shown in Fig. 2. All nodes $\{d_0, d_1, \ldots, d_{n-1}, d_n\}$ cache the source route, $R_{(0,n)}$, along with their respective multisignatures $\{\sigma_{(0,n)}, \sigma_{(1,n)}, \ldots, \sigma_{(n-1,n)}, \sigma_{(n)}\}$ (destination caches its own signature $\sigma_{(n)}$). Suppose node d'_0 (a new source) now attempts to discover a route to the same destination d_n and the RREQ packet generated by the new source d'_0 containing the accumulated route $\{d'_0, d'_1, \ldots, d'_m\}$ reaches node d_l as shown in Fig. 2.

Node d_l prepares the RREP packet containing: (1) cached information $\langle R_{(0,n)}, \sigma_{(l,n)} \rangle$ and (2) accumulated route, signature pair $\langle R'_{(0,l)} = \{d'_0, \ldots, d'_k, \ldots d'_m, d_l\}, \sigma'_{(l)} \rangle$, where $\sigma'_{(l)}$ is node d_l's own signature on the hashed concatenation of the IP address in the accumulated route $h' = H(d'_0|| \ldots ||d'_m||d_l)$. Node d_l sends the RREP packet to node d'_m. Now, consider an arbitrary node d'_k en route to the

---

[2] We use multisignatures in authenticating cached routes and thus, do not consider using link caches [4].

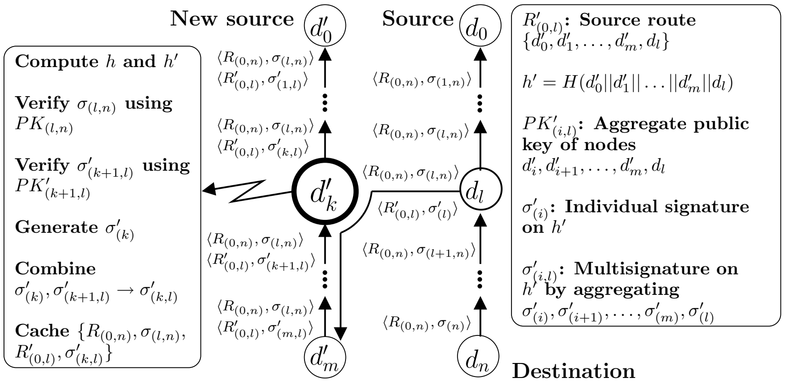
New source
d'_0
Source
d_0
$R'_{(0,l)}$: Source route $\{d'_0, d'_1, \ldots, d'_m, d_l\}$
$h' = H(d'_0||d'_1||\ldots||d'_m||d_l)$
$PK'_{(i,l)}$: Aggregate public key of nodes $d'_i, d'_{i+1}, \ldots, d'_m, d_l$
$\sigma'_{(i)}$: Individual signature on h'
$\sigma'_{(i,l)}$: Multisignature on h' by aggregating $\sigma'_{(i)}, \sigma'_{(i+1)}, \ldots, \sigma'_{(m)}, \sigma'_{(l)}$
Compute h and h'
Verify $\sigma_{(l,n)}$ using $PK_{(l,n)}$
Verify $\sigma'_{(k+1,l)}$ using $PK'_{(k+1,l)}$
Generate $\sigma'_{(k)}$
Combine $\sigma'_{(k)}, \sigma'_{(k+1,l)} \rightarrow \sigma'_{(k,l)}$
Cache $\{R_{(0,n)}, \sigma_{(l,n)}, R'_{(0,l)}, \sigma'_{(k,l)}\}$
$\langle R_{(0,n)}, \sigma_{(l,n)}\rangle$
$\langle R'_{(0,l)}, \sigma'_{(1,l)}\rangle$
$\langle R_{(0,n)}, \sigma_{(l,n)}\rangle$
$\langle R'_{(0,l)}, \sigma'_{(k,l)}\rangle$
d'_k
$\langle R_{(0,n)}, \sigma_{(l,n)}\rangle$
$\langle R'_{(0,l)}, \sigma'_{(k+1,l)}\rangle$
$\langle R_{(0,n)}, \sigma_{(l,n)}\rangle$
$\langle R'_{(0,l)}, \sigma'_{(m,l)}\rangle$
d'_m
$\langle R_{(0,n)}, \sigma_{(1,n)}\rangle$
$\langle R_{(0,n)}, \sigma_{(l,n)}\rangle$
$\langle R_{(0,n)}, \sigma_{(l,n)}\rangle$
d_l
$\langle R'_{(0,l)}, \sigma'_{(l)}\rangle$
$\langle R_{(0,n)}, \sigma_{(l+1,n)}\rangle$
$\langle R_{(0,n)}, \sigma_{(n)}\rangle$
d_n
Destination

Fig. 2. Propagation and authentication of cached RREP

new source d'_0. Node d'_k does the following: (1) Verifies multisignatures $\sigma_{(l,n)}$ and $\sigma'_{(k+1,l)}$ that it receives from its previous node d'_{k+1} using aggregate public keys $PK_{(l,n)}$ and $PK'_{(k+1,l)}$ respectively; (2) generates it's own signature $\sigma'_{(k)}$ on h'; (3) combines $\sigma'_{(k)}$ and $\sigma'_{(k+1,l)}$ to form multisignature $\sigma'_{(k,l)}$ on h'; (4) caches $\{R_{(0,n)}, \sigma_{(l,n)}, R'_{(0,l)}, \sigma'_{(k,l)}\}$ under the cached entry for route to d_n and (5) sends the RREP packet containing $\langle R_{(0,n)}, \sigma_{(l,n)}\rangle$ and $\langle R'_{(0,l)}, \sigma'_{(k,l)}\rangle$ to node d'_{k-1}. In this fashion, the RREP packet propagates to the source d'_0, which performs the same operations as node d'_k. Successful verification of multisignatures $\sigma_{(l,n)}$, $\sigma'_{(1,l)}$ under the aggregate public keys $PK_{(l,n)}$, $PK'_{(1,l)}$ establishes the authenticity of the route $R'_{(0,l)}$ and the partial route $\{d_{l+1}, \ldots, d_n\}$ contained in $R_{(0,n)}$. Note that the remaining part of $R_{(0,n)}$, i.e., IP addresses $\{d_0, \ldots, d_{l-1}\}$, are not authenticated by nodes. Finally, source d'_0 extracts $\{d_{l+1}, \ldots, d_n\}$ from $R_{(0,n)}$, and appends the extracted route to $R'_{(0,l)}$ to obtain the desired route $\{d'_0, \ldots, d'_m, d_l, \ldots, d_n\}$. Similarly, nodes in the route $R'_{(0,l)}$ may use cached information to reply to future RREQs encountered for destination d_n.

5 Construction of an Efficient and Scalable Multisignature Scheme

In this section, we construct an efficient, single round, multisignature scheme based on cubic LFSR sequences, suitable for authenticating route discovery in DSR.

5.1 A Variant of a Generalized ElGamal Signature Scheme based on LFSR Sequences

We present the cubic LFSR-based individual signature scheme, `CLFSR-S` [5], with a unique construction that uses the EG I.4 [6] variant of the generalized

ElGamal signature scheme. Generation of individual signatures and verification of (multi)signatures in multisignature scheme, CLFSR-M, follow the procedures in CLFSR-S.

Signature Generation	**Signature Verification**
1. Randomly choose ephemeral private key $k \in_R \mathbb{Z}_Q^*$ and compute ephemeral public key $\bar{s}_k \leftarrow \mathsf{OP}_1(k, f)$. Denote $r = s_k \mod Q$ as an integer. 2. Compute hash of message $h = H(m)$; Solve for t in the following equation: $t \equiv kr - xh \mod Q$. 3. Compute $\bar{s}_{kr} \leftarrow \mathsf{OP}_1(\bar{s}_k, r)$. 4. Send the signature $\sigma = \big(\bar{s}_{kr}, t\big)$ and the message m to verifier.	1. Compute $h = H(m)$. 2. Compute $A = f_{(th^{-1}+x)} \leftarrow \mathsf{OP}_2(th^{-1}, \bar{s}_x)$. 3. Compute $B = f_{(rh^{-1}k)} \leftarrow \mathsf{OP}_1(h^{-1}, f_{kr})$. f_{kr} can be directly derived from $\bar{s}_{kr}$. 4. Accept signature if $A = B$, else reject signature.

Fig. 3. The CLFSR-S Signature Scheme

The CLFSR-S scheme consists of four phases: initialization, key generation, signature generation and signature verification. During the initialization phase, both entities, i.e., the signer and the verifier, choose and agree on the system public parameters: $\mathsf{params} = \langle p, Q, f(x), H \rangle$, where p, Q and $f(x)$ are as described in Section 3.1 and $H : \{0,1\}^* \mapsto \mathbb{Z}_Q$ is a cryptographic hash function. The signer generates its long-term private and public key pair, $(SK, PK) = (x, \bar{s}_x)$. Fig. 3 describes the signature generation and signature verification phases of CLFSR-S scheme. Note that a naive cubic LFSR variant of EG I.4 will generate the signature $\sigma = (f_k, t)$. We perform an additional computation in Step 3 (Fig. 3) of the signature process to compute the term $\bar{s}_{kr}$. The specific format of the individual signature that CLFSR-S generates enables us to efficiently construct the multisignature in a single round, i.e., without any prior cooperation among the nodes participating in the RREQ phase of DSR.

Next, we present an efficient, single round multisignature scheme that uses the individual CLFSR-S signatures to generate a multisignature.

5.2 The Proposed Multisignature Scheme CLFSR-M

The multisignature scheme, CLFSR-M, consists of five phases: initialization, key generation (MS.K), signature generation (MS.G), multisignature verification (MS.V) and multisignature generation (MS.A). During the initialization phase, all nodes choose and agree upon the system public parameters $\mathsf{params} = \langle p, Q, f(x), H \rangle$. The process of key generation consists of: (1) generation of individual long term private public key pair $(SK_l, PK_l) = (x_l, \bar{s}_{x_l})$ of node d_l and (2) the generation of aggregate public key $PK_{(l,n)} = \bar{s}_{x_{(l,n)}} \leftarrow \mathsf{OP}_2(\bar{s}_{x_l}, \bar{s}_{x_{(l+1,n)}})$ of nodes $d_l, d_{l+1}, \ldots, d_n$, where $x_{(l,n)} = \sum_{i=l}^{n}(x_i)$.

The signature generation, multisignature verification and multisignature generation phases of CLFSR-M work as follows:

1. Signature generation ($\mathsf{MS.G}(\mathsf{params}, SK_l, m = d_0||\ldots||d_n) \rightarrow \sigma_{(l)}$): Each node, d_l, participating in the RREP propagation generates an individual signature $\sigma_{(l)} = (\bar{s}_{k_l r_l}, t_l)$ on the hashed concatenation of the IP address in the source route $h = H(m)$ following the $\mathsf{CLFSR\text{-}S}$ signature generation.
2. Multisignature Verification ($\mathsf{MS.V}(\mathsf{params}, PK_{(l+1,n)}, \sigma_{(l+1,n)}, m) \rightarrow (Valid, Invalid)$): Each intermediate node (other than the destination), d_l, receives a signed RREP packet containing the multisignature $\sigma_{(l+1,n)} = (t_{(l+1,n)}, \bar{s}_{k_{(l+1,n)}})$, where $t_{(l+1,n)} = \sum_{i=l+1}^{n}(t_i)$ and $k_{(l+1,n)} = \sum_{i=l+1}^{n}(k_i r_i)$. Node d_l verifies $\sigma_{(l+1,n)}$ following the $\mathsf{CLFSR\text{-}S}$ signature verification procedure, using the aggregate public key $PK_{(l+1,n)} = \bar{s}_{x_{(l+1,n)}}$, where $x_{(l+1,n)} = \sum_{i=l+1}^{n}(x_i)$. Note that for the node d_{n-1} (the last hop before the destination d_n) the signature $\sigma_{(l+1,n)}$ denotes σ_n.
3. Multisignature Generation ($\mathsf{MS.A}(\mathsf{params}, \sigma_{(l+1,n)}, \sigma_{(l)}) \rightarrow \sigma_{(l,n)}$): If the signature $\sigma_{(l+1,n)}$ passes the verification procedure, $\mathsf{MS.V}$, node d_l, generates the multisignature $\sigma_{(l,n)}$ by computing $t_{(l,n)} = t_{(l+1,n)} + t_l$ and $\bar{s}_{k_{(l,n)}} = \bar{s}_{k_{(l+1,n)}+k_l r_l} \leftarrow \mathsf{OP}_2(\bar{s}_{k_l r_l}, \bar{s}_{k_{(l+1,n)}})$. Node d_l finally removes the multisignature $\sigma_{(l+1,n)}$ from and adds the multisignature $\sigma_{(l,n)} = (t_{(l,n)}, \bar{s}_{k_{(l,n)}})$ to the RREP packet before forwarding the RREP to the next hop node d_{l-1}.

The wave of signature generation, multisignature verification and multisignature aggregation continues until the RREP packet containing the multisignature $\sigma_{(1,n)} = (t_{(1,n)}, \bar{s}_{k_{(1,n)}})$, is delivered to the source. If the multisignature $\sigma_{(1,n)}$ passes the verification procedure, $\mathsf{MS.V}$, under the aggregate public key $PK_{(1,n)}$, then individual signatures $\sigma_{(1)}, \ldots, \sigma_{(n)}$ of corresponding nodes $d_1, \ldots, d_n$ in the discovered source route (to the destination d_n) are verified collectively. In the following section, we present a discussion on policy aspects of bootstrapping authentication protocols in ad hoc networks.

6 A Discussion on Distributing Public keys

An authentication protocol is typically composed of two distinct phases — the bootstrapping phase and the authentication phase. In the realm of public key cryptography, entities need to use authentic channels (need not be confidential) to exchange public keys constituting the "bootstrapping material" [29]. Once this exchange has taken place in the bootstrapping phase, entities can authenticate each other by proving the possession of their corresponding private keys.

6.1 Using a Trusted Third Party

A trusted third party (TTP) can be used to distribute certified public keys (the bootstrapping material) and also provide a way to check the validity of certificates via publishing certificate revocation lists. An online TTP works as follows: an arbitrary node d_k wanting to authenticate the source route can request and receive certified copies of public keys $PK_{k+1}, \ldots, PK_n$ of nodes leading to the destination from the TTP. However, an online TTP in an ad hoc network introduces circular dependency between the need for a TTP to perform secure routing and the need to find a secure route to the TTP. However, in such a case, public keys have to be redistributed when network membership changes, i.e., when nodes join or leave the network. To avoid this, an offline TTP can distribute all certified public keys to all nodes when the network is set up. Such an offline TTP may not be viable, since nodes would require to store all certified public keys. Various such solutions of bootstrapping authentication have been proposed for securing ad hoc networks, each having its own disadvantages [30]. In essence, the assumption of a TTP-based public key management policy in an ad hoc networking paradigm is not practical. Delegating specialized functions to a single node or a small subset of nodes [31, 32] does not suit the ad hoc networking paradigm. These restrictions motivate us towards a fully distributed public key management policy.

6.2 Towards Fully Distributed Self-organized Bootstrapping

PGP [8] is a policy-based mechanism for public key management and can be used to distribute certified copies of public keys in the absence of a centralized TTP. In PGP, each node generates its own (public, private) key pair and certifies its own public key as well as public keys of other nodes based on certain trust policies. Similarly, in an ad hoc network, when two nodes come within radio-range of each other, they can certify each other's public keys, based on policies. This process of certification creates a *certificate graph* $G = (V, E)$, where $V = \{d_0, d_1, \ldots, d_N\}$ and $E = \{(d_i, d_j) : \forall i, j : 0 \leq i, j \leq N, \exists\, \sigma_{SK_i}(d_j, PK_j)\}$, where N is the total number of nodes in the network and $\sigma_{SK_i}(d_j, PK_j)$ denotes node d_i's signature on node d_j's public key. When a node d_i wants to verify the authenticity of public key PK_j of node d_j, node d_i tries to find a simple path $d_i \rightsquigarrow d_j = d_i \rightarrow d_{i_0} \rightarrow \ldots d_{i_n} \rightarrow d_j$ in the certificate graph, where $d_{i_k} \rightarrow d_{i_l} \implies (d_{i_k}, d_{i_l}) \in E$. Capkun et al. [10, 11] studied PGP certificate graphs and observed that trust graphs in self-organized systems, for example mobile ad hoc networks, naturally exhibit the *small-world* phenomenon. Informally, a graph is said to exhibit the *small-world* property if any two nodes in the network are likely to be connected through a short sequence of intermediate acquaintances. Since the first experimental study by Milgram [9], several network models [22, 33] have been proposed to study the problem analytically.

In our public key management model, individual nodes store, manage and distribute certificates themselves in a such a way that the size of the certificate repository at each node is small compared to the total number of certificates in the network, while still maintaining a high probability of finding a trust path from one node to another. We assume routing initiates after convergence of the certificate graph.

Policy Variants

In Policy I, the trust is based on the following. Node d_i completely[3] trusts node d_j implies: (1) node d_i believes that node d_j's public key PK_j is valid

---

[3] For simplicity, we assign trust either a true or false value. We do not model marginal or partial trust.

and authentic, and (2) node d_i trusts node d_j's decision on signing any other public key PK_k of node d_k, i.e., d_j would be careful not to sign any bogus public key. Thus, the following condition should hold for authenticating the route discovery process: $\forall i, \exists\, d_i \rightsquigarrow d_j, i < j \leq n$. Informally, this condition means that any node d_i wanting to authenticate the route from itself to the destination $\{d_i, d_{i+1}, \ldots, d_n\}$ needs to find a way to verify the authenticity of all corresponding public keys $\{PK_i, PK_{i+1}, \ldots, PK_n\}$.

In Policy II, the trust policy has an added condition. Node d_i completely trusts node d_j implies: (1), (2) and (3) node d_i trusts node d_j to honestly aggregate and sign other public keys $PK_{j+1}, \ldots, PK_n$ of nodes $d_{j+1}, \ldots, d_n$. Note that in Policy II, the following condition should hold for authenticating the route discovery process: $\forall i, \exists\, d_i \rightsquigarrow d_{i+1}, 0 \leq i \leq (n-1)$. This means that any node d_i wanting to authenticate the route from itself to the destination $\{d_i, d_{i+1}, \ldots, d_n\}$ needs to look-up a single node in the certificate graph, i.e., to verify the authenticity of one public key PK_{i+1}. Node d_{i+1} would sign the aggregate public public key $PK_{(i+1,n)}$ and deliver $(PK_{(i+1,n)}, \texttt{Cert}_{(i+1,n)} = \sigma_{SK_{i+1}}(d_{i+1}, \ldots, d_n, PK_{(i+1,n)}))$ to node d_i, where $\texttt{Cert}_{(i+1,n)}$ denotes the certificate on the aggregate public key $PK_{(i+1,n)}$.

7 Theoretical Analysis

We present a concise theoretical analysis of correctness, security and performance of the proposed multisignature CLFSR-M.

7.1 Correctness

A multisignature scheme constructed following the procedures described in Section 5.2 is correct if an arbitrary multisignature, $\sigma_{(l+1,n)}$, received by node $d_l \in \{d_0, \ldots, d_{n-1}\}$ from node, d_{l+1}, passes the verification procedure MS.V at node d_l under the aggregate public key $PK_{(l+1,n)}$ provided: (1) Each node $d_i \in d_{l+1}, \ldots, d_n$ chooses and agrees upon the system public parameters $\mathsf{params} = \langle p, Q, f(x), H \rangle$ and, honestly executes the key generation algorithm, $\mathsf{MS.K}(\mathsf{params}) \rightarrow (PK_i, SK_i)$ and the signature generation algorithm, $\mathsf{MS.G}(\mathsf{params}, SK_i, m) \rightarrow \sigma_{(i)}$, where $m = d_0 || \ldots || d_n$; (2) each node $d_i \in \{d_{l+1}, \ldots, d_{n-1}\}$, honestly executes the multisignature generation algorithm, $\mathsf{MS.A}(\mathsf{params}, \sigma_{(i+1,n)}, \sigma_{(i)}) \rightarrow \sigma_{(i,n)}$.

Proposition 1. *The multisignature scheme* CLFSR-M *follows the correctness property.*

Proof. Consider any arbitrary node $d_l \in \{d_0, \ldots, d_{n-1}\}$. We show that the multisignature, $\sigma_{(l+1,n)}$, of node d_{l+1} passes the verification procedure $\mathsf{MS.V}(\mathsf{params}, PK_{(l+1,n)}, \sigma_{(l+1,n)}, m) \rightarrow (Valid, Invalid)$ executed at d_l under the aggregate public key $PK_{(l+1,n)} = \bar{s}_{x_{(l+1,n)}}$ provided the above mentioned conditions hold.

In the verification of the multisignature $\sigma_{(l+1,n)}$ using the algorithm MS.V, we observe: $A_{(l+1,n)} = f_{v+x_{(l+1,n)}} \leftarrow \mathsf{OP}_2(v, \bar{s}_{x_{(l+1,n)}})$, where, $v = h^{-1} \sum_{i=l+1}^{n} (t_i)$

and $x_{(l+1,n)} = \sum_{i=l+1}^{n}(x_i)$. All nodes use the signing equation: $t_i \equiv k_i r_i - x_i h \mod Q$, where $(l+1) \leq i \leq n$ and $k_{(l+1,n)} = \sum_{i=l+1}^{n}(k_i r_i)$. Thus,

$$\begin{aligned} A_{(l+1,n)} &= f_{\sum_{i=l+1}^{n}(h^{-1}t_i + x_i)} = f_{\sum_{i=l+1}^{n}(h^{-1}k_i r_i)} \\ &= f_{h^{-1}k_{(l+1,n)}} = \mathsf{OP}_1(h^{-1}, f_{k_{(l+1,n)}}) = B_{(l+1,n)} \end{aligned}$$

Thus, the multisignature, $\sigma_{(l+1,n)}$, is valid under $PK_{(l+1,n)}$.

Now, we need to show that it is hard for an adversary to deviate from the key pair and signature generation algorithms and still generate a correct signature. However, this is precisely the issue of forgery which we discuss in the following section.

7.2 Security

The security of CLFSR-M is based on the difficulty of solving the trace discrete logarithm (Tr-DL) problem in $\mathbb{F}_q$ [7, 24–26]. Informally, the trace function $Tr : \mathbb{F}_{q^3} \mapsto \mathbb{F}_q$ is given as $Tr(\alpha) = \alpha + \alpha^q + \alpha^{q^2}$. The Tr-DL problem and assumption can be defined as follows:

Definition 1 (Tr-DL Problem/Assumption). *Let α be a generator of the multiplicative group $(\mathbb{F}_{q^3})^*$, where q is a large prime or a power of a large prime. The Tr-DL Problem in $\mathbb{F}_q$ can be defined as follows: Given $(q, \alpha \in (\mathbb{F}_{q^3})^*, \beta \in \mathbb{F}_q)$, find an index k such that $\beta = Tr(\alpha^k)$ or determine that there is no such index. Let $\mathcal{A}$ be a probabilistic polynomial time (PPT) algorithm that runs in time t and solves the Tr-DL problem with probability at least ϵ. Define the advantage of the (t, ϵ) Tr-DL solver $\mathcal{A}$ as: $\mathsf{Adv}_{\mathcal{A}}^{TrDL} = Pr[\mathcal{A}(q, \alpha, \beta) = k \mid \alpha \in_R \mathbb{F}_{q^3}, k \in_R \mathbb{Z}_Q, \beta = Tr(\alpha^k)]$. The probability is over the random choices of α, k and the random bits of $\mathcal{A}$.*

Tr-DL Assumption*: The finite field $\mathbb{F}_q$ satisfies the Tr-DL Assumption if $\mathsf{Adv}_{\mathcal{A}}^{TrDL}(\lambda)$ is a negligible function.*

Lemma 1 (Giuliani et al. [7]). *The Tr-DL Problem is equivalent to the DL problem.*

A total break of CLFSR-M occurs if, given a public key $\mathsf{PK}_i = \bar{s}_{x_i}$ of an arbitrary node d_i, the adversary is able to compute the corresponding private key $\mathsf{SK}_i = x_i$. In such a case, any node's signature can be forged. However, given $\bar{s}_x$, finding x is equivalent to solving the DL problem in the extension field $\mathbb{F}_{q^3}$ [25]. Using the following lemmas we show that, assuming a total break has not occurred, if an adversary can successfully forge a CLFSR-M multisignature, then he can successfully forge a signature in the EG I.4 variant of the generalized ElGamal scheme.

Lemma 2 (Chakrabarti et al. [5]). *The 2-party signature scheme* CLFSR-S *is equivalent to the well-known EG I.4 variant of the Generalized ElGamal scheme.*

Lemma 3. *The 2-party signature scheme* CLFSR-S *reduces to the proposed multisignature scheme* CLFSR-M*.*

Proof (Sketch). Suppose there exists a PPT forger $\mathcal{F}$, which given system parameters $\mathsf{params} = \langle p, Q, f(x), H\rangle$, public keys $\bar{s}_{x_0}, \ldots, \bar{s}_{x_n}$ and message m, outputs a forged multigsignature $\sigma^F_{(0,n)} = (t^F_{(0,n)}, \bar{s}^F_{k_{(0,n)}})$ on $h = H(m)$ with non-negligible probability, i.e., $\sigma^F_{(0,n)}$ passes the verification procedure, MS.V, under the aggregate public key $\bar{s}_{x_{(0,n)}}$.

We show that given access to the PPT forger $\mathcal{F}$, system parameters params, public key $PK = \bar{s}_x$ and message m, an adversary can output a forged signature $\sigma^F = (\bar{s}^F_{kr}, t^F)$ on $h = H(m)$ that passes the verification procedure of CLFSR-S under public key PK. The adversary generates σ^F as follows: (1) picks $x_0, \ldots, x_{n-1} \in_R \mathbb{Z}^*_Q$ and computes $\bar{s}_{x_0}, \ldots, \bar{s}_{x_{n-1}}$, (2) computes $\bar{s}_{x_n} \leftarrow \mathsf{OP}_2(-\sum_{i=0}^{n-1} x_i, \bar{s}_x)$, (3) calls $\mathcal{F}$ with inputs params, $\bar{s}_{x_0}, \bar{s}_{x_1}, \ldots, \bar{s}_{x_n}$ and m. $\mathcal{F}$ outputs forged multisignature $(t^F_{(0,n)}, \bar{s}^F_{k_{(0,n)}})$ on $h = H(m)$; and (4) sets $\bar{s}^F_{kr} = \bar{s}^F_{k_{(0,n)}}$ and $t^F = t^F_{(0,n)}$. The signature $\sigma^F = (\bar{s}^F_{kr}, t^F)$ is a forged signature on $h = H(m)$ under public key PK.

Theorem 1. *The well-known EG I.4 variant of the Generalized ElGamal signature scheme reduces to the proposed multisignature scheme* CLFSR-M*.*

Proof. The proof of the theorem is immediate from Lemmas 2 and 3.

Note that CLFSR-M, though not provably secure, is engineered to be an efficient means to authenticate routes in DSR. In contrast, Micali et al.'s multisignature scheme [14] uses the Schnorr's variant [21] (the only known provably secure variant) and takes three communication rounds. We omit a thorough discussion on provable security; the reader is requested to refer to [34] for an exemplary discussion on the subject. Next, we present a performance comparison of CLFSR-M with existing schemes.

7.3 Performance

Table 1 shows a performance comparison of the proposed multisignature scheme, CLFSR-M with three signature aggregation techniques used to instantiate SRDP [3], namely the multisignature by Micali et al. (ASM) [14], the generalized aggregate signature by Boneh et al. (MBLS) [15] and the sequential aggregate signature by Lysyanskaya et al. (SAS) [16].

The original construction of Micali et al.'s multisignature scheme [14] takes three communication rounds; ASM in SRDP requires two rounds for completion, with prior cooperation (though small: one exponentiation and one modular multiplication) among nodes during the RREQ phase, which might be wasteful if the node is not included in the final route. The proposed multisignature, CLFSR-M, uses extremely fast LFSR sequence operations [26, 28] and achieves the best computational efficiency. The public key sizes equivalent to 1024-bit RSA (excluding shared components of the public key) are highest in SAS and ASM, followed by MBLS. CLFSR-M offers the least public key size. Note that in ASM, nodes need to additionally propagate the accumulated ephemeral public

	SAS	ASM	MBLS	CLFSR-M
Rounds	2	1	1	1
Generation cost	$e + h$	$e + 2m + h$	$s + h$	$2\mathsf{OP}_1 + h + 2m$
Verification cost	$n(h + e)$	$2e + m + h$	$2p + h$	$\mathsf{OP}_1 + \mathsf{OP}_2 + h + m$
Aggregation cost	–	–	m	OP_2
Signature size (bits)	1024	$320 + (160^*)$	160	500
PK size (bits)	2048	2048	766	680

Table 1. Performance Comparison. e : modular exponentiation, m : modular multiplication, h : hash operation, p : pairing computation, s : scalar multiplication, n : number of signers, $*$: ephemeral public key propagated during RREQ phase

keys (160-bits) during the RREQ phase, wasting bandwidth. Signature sizes are lowest for MBLS, followed by ASM and CLFSR-M, while SAS incurs the highest sizes.

8 Conclusions

In this paper, we presented the first LFSR sequence based multisignature scheme CLFSR-M geared toward authenticating routes in DSR. Our scheme also works with cached routing information. The scheme CLFSR-M scheme is derived from a cubic LFSR sequence-based, 2-party signature scheme, CLFSR-S [5], and uses extremely fast LFSR operations, small public keys (smallest among schemes in [3]) and generates a reasonably small multisignature (500 bits). The security of the scheme, CLFSR-M, is based on the Tr-DL(DL) Problem in $\mathbb{F}_q(\mathbb{F}_{q^3})$. CLFSR-M was constructed using the XTR-PKC for simplicity, although it can be seamlessly constructed using the GH-PKC and can also be extended to PKCs based on higher order LFSR sequences, with minor modifications, depending on the desired security level.

Distributing authentic public keys among nodes in a mobile ad hoc network to bootstrap authentication protocols is a challenging task. Delegating special functions to nodes or assuming the existence of a TTP to distribute certified public keys is paradigmatically unsuitable for ad hoc networks. We consider a fully distributed mechanism of public key distribution and present two variations of trust-policies, based on PGP, for effective management of individual and aggregate public keys.

References

1. Johnson, D.B., Maltz, D.A., Hu, Y.C.: The Dynamic Souce Routing Protocol for Mobile Ad Hoc Networks (DSR). Internet draft draft-ietf-manet-dsr-10, IETF MANET Working Group (July 2004)
2. Hu, Y.C., Perrig, A.: A survey of secure wireless ad hoc routing. IEEE Security & Privacy **2**(3) (2004) 28–39
3. Kim, J., Tsudik, G.: SRDP: Securing route discovery in DSR. In: Proceedings of MobiQuitous, IEEE Computer Society (2005) 247–260

4. Hu, Y.C., Johnson, D.B.: Caching strategies in on-demand routing protocols for wireless ad hoc networks. In: Proceedings of MOBICOM. (2000) 231–242
5. Chakrabarti, S., Chandrasekhar, S., Singhal, M., Calvert, K.L.: Authenticating feedback in multicast applications using a novel multisignature scheme based on cubic LFSR sequences. To appear in Proceedings of SSNDS (2007)
6. Horster, P., Petersen, H., Michels, M.: Meta-ElGamal signature schemes. In: ACM Conference on Computer and Communications Security. (1994) 96–107
7. Giuliani, K.J., Gong, G.: New LFSR-based cryptosystems and the trace discrete log problem (trace-DLP). In: Proceedings of SETA. (2004) 298–312
8. Zimmermann, P.: The official PGP user's guide (1995)
9. Milgram, S.: The small world problem. Psychology Today **61**(2) (1967) 60–67
10. Čapkun, S., Buttyán, L., Hubaux, J.P.: Small worlds in security systems: an analysis of the PGP certificate graph. In: Proceedings of NSPW. (2002)
11. Čapkun, S., Buttyán, L., Hubaux, J.P.: Self-organized public-key management for mobile ad hoc networks. IEEE Transactions on Mobile Computing **2**(1) (2003) 52–64
12. Papadimitratos, P., Haas, Z.J.: Secure routing for mobile ad hoc networks. In: Proceedings of CNDS. (2002)
13. Hu, Y.C., Perrig, A., Johnson, D.B.: Ariadne: A secure on-demand routing protocol for ad hoc networks. Wireless Networks **11**(1-2) (2005) 21–38
14. Micali, S., Ohta, K., Reyzin, L.: Accountable-subgroup multisignatures: extended abstract. In: Proceedings of CCS. (2001) 245–254
15. Boneh, D., Gentry, C., Lynn, B., Shacham, H.: Aggregate and verifiably encrypted signatures from bilinear maps. In: Proceedings of EUROCRYPT. Volume 2656 of LNCS., Springer (2003) 416–432
16. Lysyanskaya, A., Micali, S., Reyzin, L., Shacham, H.: Sequential aggregate signatures from trapdoor permutations. In: Proceedings of EUROCRYPT. Volume 3027 of LNCS., Springer (2004) 74–90
17. Acs, G., Buttyan, L., Vajda, I.: Provably Secure On-Demand Source Routing in Mobile Ad Hoc Networks. IEEE Transactions on Mobile Computing **5**(11) (2006) 1533–1546
18. Bhaskar, R., Herranz, J., Laguillaumie, F.: Efficient authentication for reactive routing protocols. In: Proceedings of AINA, IEEE Computer Society (2006) 57–61
19. Itakura, K., Nakamura, H., Nakazawa, K.: A public-key cryptosystem suitable for digital multisignatures. NEC Research and Development (1983)
20. ElGamal, T.: A public key cryptosystem and a signature scheme based on discrete logarithms. In: Proceedings of CRYPTO. Volume 196 of LNCS. (1984) 10–18
21. Schnorr, C.P.: Efficient signature generation by smart cards. Journal of Cryptology **4**(3) (1991) 161–174
22. Kleinberg, J.M.: The small-world phenomenon: an algorithm perspective. In: Proceedings of STOC. (2000) 163–170
23. Niederreiter, H.: A public-key cryptosystem based on shift register sequences. In: Proceedings of EUROCRYPT, Springer-Verlag (1986) 35–39
24. Gong, G., Harn, L.: Public-key cryptosystems based on cubic finite field extensions. IEEE Transactions on Information Theory **45**(7) (1999) 2601–2605
25. Gong, G., Harn, L., Wu, H.: The GH public-key cryptosystem. In: Proceedings of SAC, Springer-Verlag (2001) 284–300
26. Lenstra, A.K., Verheul, E.R.: The XTR Public Key System. LNCS **1880** (2000) 1–19
27. Golomb, S.W.: Shift Register Sequences. Holden-Day (1967)

28. Peeters, E., Neve, M., Ciet, M.: XTR implementation on reconfigurable hardware. In: Proceedings of CHES. Volume 3156 of LNCS., Springer (2004) 386–399
29. Chakrabarti, S., Giruka, V.C., Singhal, M.: Security in Distributed, Grid, and Pervasive Computing, Edited by Prof. Yang Xiao. Auerbach Publications, CRC Press (2006)
30. Čapkun, S., Hubaux, J.P.: BISS: building secure routing out of an incomplete set of security associations. In: Workshop on Wireless Security, ACM (2003) 21–29
31. Zhou, L., Haas, Z.J.: Securing ad hoc networks. IEEE Network **13**(6) (1999) 24–30
32. Kong, J., Zerfos, P., Luo, H., Lu, S., Zhang, L.: Providing robust and ubiquitous security support for mobile ad hoc networks. In: Proceedings of ICNP, IEEE Computer Society (2001) 251–260
33. Watts, D.J.: Small Worlds: The Dynamics of Networks Between Order and Randomness. Princeton University Press (1999)
34. Koblitz, N., Menezes, A.: Another Look at "Provable Security". II. In: Proceedings of INDOCRYPT. (2006) 148–175