A new collision-free pseudonym scheme in mobile ad hoc networks

Ji Won Yoon
Department of Engineering Science,
University of Oxford, UK.
Email: jwyoon@robots.ox.ac.uk

Hyoungshick Kim
Computing Laboratory,
University of Cambridge, UK.
Email: hk331@cam.ac.uk

Abstract—A mobile ad hoc network (MANET) is a decentralized network of mobile nodes. Due to the broadcast nature of radio transmissions, communication in MANETs is more susceptible to malicious traffic analysis. An interesting problem is how to thwart malicious traffic analysis. Most anonymous communication protocols are based on the pseudonyms of mobile nodes. However, conventional pseudonym schemes have some limitations such as collisions of pseudonyms and high computational complexity due to the use of cryptographic hash functions. Collisions of identities are not desirable since they are the main causes for reduced effective bandwidth, increased energy consumption and non-deterministic data delivery. In this paper, we propose a new collision-free pseudonym scheme to enable anonymous communication. In our approach, each node generates pseudonyms by using a permutation matrix without collisions. The challenging issue is how to store the overall permutation matrix. It is practically hard to assume that mobile nodes maintain the permutation matrix due to the limitation of resources. Therefore we design the online computation of each node's own pseudonym without loading the overall matrix.

I. INTRODUCTION

As the availability of mobile devices and wireless networking increases ever more rapidly, so does the requirement to mobile ad-hoc networks (MANETs) in wireless networks become more pressing. While conventional wireless mobile communications are supported by a fixed wire or wireless infrastructure, the MANETs do not have any fixed infrastructure. In addition, MANETs are highly dynamic in time and space. The nodes in MANETs intercommunicate each other in a peer-to-peer way. On connection, the intermediate nodes act as a routers and the nodes operate both as hosts and routers. There exist several operations including node adding and deleting in the network (trans-dimensional environment) and the ad hoc network is mobile over time. Thus, the topology of the network is randomly changeable, dynamic and unexpected. These characteristics of the MANETs result in both increasing popularity and increasing security issues [1], [2].

One of the most subtle and unsolved security attacks against MANETs is traffic analysis which is an attack such that an adversary observes network traffic and infers sensitive information of the applications and system. Sensitive information includes the identities of communication parties, the network traffic patterns and their changes [3].

The potentially passive eavesdropping on data communications are introduced due to the share of wireless medium of MANETs. There have been several approaches against adversaries which withstand eavesdropping and other types of traffic analysis [4], [5].

The anonymity, which is one of the most important factors in secure ad hoc network, is guaranteed in case that the intruders do not know the communication IDs of other nodes. Moreover, the traffic analysis attack can be efficiently defeated if the requirement of anonymity is satisfied in a particular network.

There have been several studies in protocols to build anonymity in MANETs. Basically, previous schemes provide anonymous communication with each node's pseudonym. These pseudonyms are generated by using a hash function, named statistically unique and cryptographically verifiable (SUCV) Identifiers and Addresses [6]. Although this scheme provides statistically unique identities, the hash algorithm employed should have an output of at least reasonable long length because of a square-root attack called the *birthday attack* [7].

Zhang et al. [8] proposed an anonymous communication protocol, named MASK, with the centralized pseudonym assignment. In MASK, the system administrator generates a sufficiently large set of pseudo IDs (pseudonyms) and then distributes them for every node. Each node should have to keep an extremely large enough number of its own pseudo IDs set. Rahman et al [9] proposed RINOMO which is an extension of MASK and computes each node's pseudo ID in a distributed manner. RINOMO solved the storage problem for maintaining a set of pseudo IDs. However, it cannot resolve the inherent problems since it still has cryptographic hash functions.

In this paper, we propose a perfectly collision free algorithm which has reasonably practical space complexity. In our proposed algorithm the system administrator generates only one pseudo IDs as done in RINOMO. After receiving the pseudonym from the system administrator, each node generates its own pseudo IDs by accessing an element of the permutation matrix in each round. In addition, conventional schemes including MASK and RINOMO are mainly based on collision-resistant cryptographic hash functions (e.g. SHA-1) and paring based operations. These operations are relatively more expensive than permutation based operations. In particular, paring operations may be not desirable in mobile applications since pairings are often the most costly operation.

II. Motivation

Assigning pseudo IDs without collision can be developed by introducing permutation array/matrix. The elements of permutation matrix act as pseudo IDs. A simple way to make collision free protocol using a permutation matrix is as follows. A sufficiently large permutation arrays are generated by a trusted system administrator in wireless network and the pseudo IDs are transmitted to all connected nodes. Afterwards, nodes works with the transmitted IDs. Suppose that we need a secure network which any node cannot know the mapping between pseudo IDs and actual node identifiers of other nodes. The system administrator can generate n pseudo IDs for n nodes every rounds. The system administrator sends the pseudo ID to an associated node every times. In this case, there are however several practical issues if this protocol works in ad-hoc networks : extremely large traffic usage (in space complexity), difficulty in pseudo IDs assignment in trans-dimensional environment. Since the pseudo IDs are sent to all nodes every round, the network may be occupied by the packets including the pseudo IDs and the packets use network resource in traffic a lot. In addition, this approach may have difficulty in dealing with IDs in case the number of nodes changes over time. For example, there are n nodes at the $t-1$th round but a new node is added to the ad-hoc system at the tth round. Because the system administrator has already made n randomly permuted IDs, the system administrator can give a pseudo ID to the new node after the system administrator regenerates $n+1$ randomly permuted IDs. Lastly, a lot of permutation operations are run only in a system administrator so we require efficient resource management in the system administrator.

All these problems come from the fact that only system administrator is in charge of generating pseudo IDs. Therefore, we need to develop a new system in which nodes generate the pseudo IDs. However, this is not so straightforward since collision of node IDs may exist if the system administrator does not regulate them and all clients do not have the same permutation matrix. Although such a permutation matrix is shared in all nodes to avoid collisions, there are two serious risks: weak anonymity and costly spaced clients due to the large matrix. Therefore, we need a modified system where only nodes can generate the pseudo IDs without such a shared large permutation matrix.

In this point of view, we suggest mixing two different concepts. In principle, the nodes can generate their own IDs at each round. In this model, we need private and public keys which are generated and sent by a system administrator. A private key is a unique pseudo ID which acts as a initial pseudo ID and the large permutation matrix is implicitly constructed with the public keys by each node. To maximize the advantage of both schemes, the delivered key from the administrator to nodes size should be small since the network usage and the system administrator's task amount depend on it.

III. Framework

We assume that there are a system administrator B and n nodes A where $A = \{A_i\}_{i=1}^n$ in our model which is the same as in MASK [8]. The system administrator B distributes private and public keys to connected nodes in a pre-processing step. Afterwards, the system administrator disappears in the network and nodes can only communicate each other with their dynamically changeable pseudo IDs. Therefore, the system administrator involves only in key distribution at the pre-processing time in order to design the decentralized communication.

The anonymity of pseudo IDs is achieved with private keys (an initial pseudo ID) and public keys (a small finite number of matrices) which are distributed at the pre-processing step. To save network traffic usage, the size of private and public keys should be minimized. In this paper, the public key is regarded as a matrix which has m IDs in each column where $m >> n$. To guarantee the anonymity, m should be large enough such that m can be 2^{64} or 2^{128}. Suppose that M is the matrix with size $m \times m$. Given a simple example with $m = 6$, we have the similar format as the following:

$$M = \begin{bmatrix} 0 & 1 & 2 & 3 & 4 & 5 \\ 1 & 2 & 0 & 4 & 5 & 3 \\ 2 & 0 & 1 & 5 & 3 & 4 \\ 3 & 4 & 5 & 0 & 1 & 2 \\ 4 & 5 & 3 & 1 & 2 & 0 \\ 5 & 3 & 4 & 2 & 0 & 1 \end{bmatrix}. \tag{1}$$

Any column in the matrix is a permuted sequence between 1 to m. The ith node can generate a pseudo ID from this matrix using private keys: an initial pseudo ID, κ_i and a random seed α. A_i's ID at round t is located at $M(\kappa_i, \beta)$ where β is a random number. β is obtained by a random generator $\beta = g(\alpha, t)$ for $\beta \in \{1, \cdots, m\}$ with the round order t.

It is impractical that a row of M is distributed to each node if m is extremely large for the anonymity. The most important question is how can we guarantee the anonymity with small network traffic usages and a collision free scheme. We propose an algorithm to access $M(\kappa_i, \beta)$ without constructing such a large matrix M as shown in Algorithm 2. In this algorithm we share only a small number of small matrices (N small matrices $\{M_j\}_{j=1}^N$) and the matrices are used to access a certain element when the system administrator or nodes ask the elements of the large matrix with κ_i and β. As a result, we do not need to distribute such a large matrix over network and we can save network traffic usage and memory storage both for system administrator and nodes. The details of the matrix accessing algorithm is described in the following Section IV.

IV. Construction of Permutation Matrix

Before describing the constructing algorithm, several terms are defined [10]. PM is an acronym of Permutation Matrix and we use PM in this paper rather than PA (Permutation Array). C is a PM over $\Re$ of size n and then C is $n \times n$ matrix. S_n denotes the set of all $n!$ permutations with n distinct elements of some fixed set R. For example, for $n = 3$ and $R = \{0, 1, 2\}$,

Algorithm 1 A collision-free pseudonym protocol

1: Initial step:
- B generates and sends initial pseudo IDs κ_i to nodes.
- B sends $\{M_j\}_{j=1}^N$ to a node A_i for all $i = 1, \cdots, n$.
- B sends a seed α to all nodes.

2: **for** $t = 1$ to m **do**
3:      for each A_i:
- Let κ_i be the ID of A_i.
- A_i generates a random number $\beta = g(\alpha, t)$.
- A_i access $ID_i^t = M(\kappa_i, \beta)$ according to Algorithm 2.

4:      **return** $\mathrm{ID}_i^{(t)}$: this is the A_i's pseudo ID at time t.
5: **end for**

$S_3 = \{012, 021, 120, 102, 201, 210\}$. (n, d) PM stands for a subset of S_n with the property that the Hamming distance between any two distinct permutations in the array is at least d. Particularly, $(n, n-1)$ PM is equivalent to Circular arrays $(d = n - 1)$. C is r-bounded if no element of R appears more than r times in any column of C and C is r-balanced if each element of R appears exactly r times in each column of C. In addition, we say that C is r-separable if is the disjoint union of r $(n, n-1)$ PM of size n.

A. Construction of Permutation Matrix from small Permutation Matrices

r-bounded $(m_1 m_2, m_1 m_2 - 1)$ PM is built by combining r-bounded $(m_1, m_1 - 1)$ PM and s-separable $(m_2, m_2 - 1)$ PM . Also, r-balanced $(m_1, m_1 - 1)$ PM and s-separable $(m_2, m_2 - 1)$ PM construct r-balanced $(m_1 m_2, m_1 m_2 - 1)$ PM [10]. Especially, we use 1-balanced $(m_1, m_1 - 1)$ PM in order that PM is orthogonal and magic square. That is, the matrix combination is defined by

$$C \times T =$$
$$\begin{bmatrix} T_{f(1,1)} + m_2 c_{1,1} J & \dots & T_{f(1,m_1)} + m_2 c_{1,m_1} J \\ \vdots & \ddots & \vdots \\ T_{f(m_1,1)} + m_2 c_{m_1,1} J & \dots & T_{f(m_1,m_1)} + m_2 c_{m_1,m_1} J \end{bmatrix} \quad (2)$$

where C is an r-balanced $(m_1, m_1 - 1)$ PM and T is a s-separable $(m_2, m_2 - 1)$ PM. Here, J is a square matrix with value 1 in all elements and the size of J is dependent on the size of T_i. $c_{i,j}$ denotes an element in ith row and jth column of C. In Eq. (2), the index function is defined as $f(i, j) = t$ if $c_{i,j} = \alpha$ and the α appears t times in the jth column.

For example, let C be an 1-balanced $(2, 1)$ PM and T be a 2-separable $(3, 2)$ PM such as the following:

$$C = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, T_1 = \begin{bmatrix} 0 & 1 & 2 \\ 1 & 2 & 0 \\ 2 & 0 & 1 \end{bmatrix}, T_2 = \begin{bmatrix} 0 & 2 & 1 \\ 2 & 1 & 0 \\ 1 & 0 & 2 \end{bmatrix}.$$

By Eq. 2 the constructed matrix is

$$C \times T_1 = \begin{bmatrix} T_1 & T_1 + 3J \\ T_1 + 3J & T_1 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 2 & 3 & 4 & 5 \\ 1 & 2 & 0 & 4 & 5 & 3 \\ 2 & 0 & 1 & 5 & 3 & 4 \\ 3 & 4 & 5 & 0 & 1 & 2 \\ 4 & 5 & 3 & 1 & 2 & 0 \\ 5 & 3 & 4 & 2 & 0 & 1 \end{bmatrix}.$$

B. Modified permutation algorithm

Our improved algorithm for construction of the permutation matrix using a finite number of small permutation matrices has following core concepts.

1) Construction of an LUT from several small matrices:
The simple way to generate permuted sequences is using the look-up table (LUT) which is 1-balanced $(m, m - 1)$ PM for a public key. The pseudo IDs are the values retrieved with indexes in the large permutation matrix. As the size of LUT
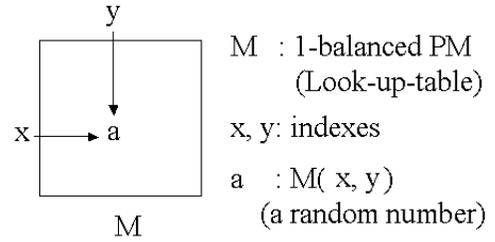

Fig. 1. Searching for $M[4][3]$

increases, it can be more flexible and useful in science and engineering. The arising question in using LUT is whether we can manage such a extremely large 1-balanced PM. This question is addressed by the construction of a large matrix by combining several small 1-balanced $(m_i, m_i - 1)$ PMs [10]. Since N 1-balanced PMs construct a large 1-balanced PM which is used for LUT, permuted sequences are generated from the LUT which is constructed by combination of small matrices.

$$M = M_1 \times M_2 \times \cdots \times M_{N-1} \times M_N$$
$$\begin{cases} M_i \text{ is an 1-balanced PM where } i \in \{1, \ldots, N\}. \\ M \text{ is also an 1-balanced PM.} \end{cases}$$

As combining two 1-balanced PMs results in an 1-balanced PM, combination of N 1-balanced PMs can build an 1-balanced PM as described in Eq. (2). Thus, if m_i is the size of the ith small PM, the constructed PM has the size of m ($= \prod_{i=1}^N m_i$). For example, LUT of size 1200 may be generated by 5 small PMs which have sizes of $5, 5, 4, 4$, and 3. The sequential combination is described as $(((M_5 \times M_4) \times M_3) \times M_2) \times M_1 \longrightarrow M$ where M is an 1-balanced $(1200, 1199)$ PM and M_i is also 1-balanced PM for $i \in \{1, 2, \cdots, 5\}$. However, it has still a serious problem in that the method using LUT should calculate and maintain the whole large matrix M during the operation. This seems inefficient and impossible if the size of LUT is over 2^{100}.

2) Efficient access to LUT: However, there are still a big issue in calculating LUT by combining small matrices. The next questions related to the issue are whether we can hide the information and whether we can access only a value of the LUT without load and calculating whole LUT. We addressed this problem by using a division algorithm to enable us to access a value of the large matrix without loading such a large matrix. We can calculate a value of LUT only with values of small matrices. Fig. 2 explains this idea with a simple example. Suppose that $M = C \times T$ and our proposed method searches

$$C = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad T = \begin{bmatrix} 0 & 1 & 2 \\ 1 & 2 & 0 \\ 2 & 0 & 1 \end{bmatrix}$$

$$M = \begin{bmatrix} 3 & 4 & 5 & 0 & 1 & 2 \\ 4 & 5 & 3 & 1 & 2 & 0 \\ 5 & 3 & 4 & 2 & 0 & 1 \\ 0 & 1 & 2 & 3 & 4 & 5 \\ 1 & 2 & 0 & 4 & 5 & 3 \\ 2 & 0 & 1 & 5 & 3 & 4 \end{bmatrix} \rightarrow M[4][3]$$

$$\begin{aligned} & i = 4 = q_x \times m_1 + r_x = 1 \times 3 + 1 \\ \implies \quad & j = 3 = q_y \times m_1 + r_y = 1 \times 3 + 0 \\ & (q_x, q_y) = (1, 1) \\ & (r_x, r_y) = (1, 0) \end{aligned}$$

Fig. 2. Searching for $M(4, 3)$

for the value in $M(4, 3)$ where $i, j \in \{0, 1, \ldots, m - 1\}$. In order to get $M(4, 3)$ without M, this approach applies Division algorithm

$$n = q \times m + r \tag{3}$$

where q and r are quotient and remainder respectively [11]. The LUT has an interesting property. M is expressed by four internal rectangles and a small circle as shown in Fig. 2. That is, after our proposed method finds a rectangle including the $M(4, 3)$, our proposed method can search for the expected value locally. First of all, the rectangular position is calculated by quotient and the exact position of the selected internal rectangle is obtained from remainder.

As shown in Fig. 2, we can calculate quotients $(q_x, q_y) = (1, 1)$ and remainders $(r_x, r_y) = (1, 0)$. Therefore, we have the values of $C(1, 1)$ and $T(1, 0)$ and then we have $M(4, 3) = 1 + 3 \times 1 = 4$ by Eq. (2). This relation is applicable to N PMs. The general algorithm described in Algorithm 2 is a recursive function and the number of iterations is the same as the number of small matrices which is obtained by a key. Our proposed method does not load or calculate the large matrix, LUT during the operation at all.

V. ANALYSIS

A. Anonymity

We use the term "anonymity" as a synonym of "anonymity in terms of unlinkability" [12] as follows. Anonymity in terms of unlinkability is defined as unlinkability of an item of interest

---

Algorithm 2 Access to LUT with small matrices

Input: $x \leftarrow$ the xth row of LUT and $y \leftarrow$ the yth column of LUT
Output: $M(x, y)$

1: **for** i=1 to N **do**
2:    **if** $i = 1$ **then**
3:       $q_{x_1} = x$ and $q_{y_1} = y$
4:    **else**
5:       $q_{x_i} = \left\lfloor \frac{q_{x_{i-1}}}{m_{i-1}} \right\rfloor$ and $q_{y_i} = \left\lfloor \frac{q_{y_{i-1}}}{m_{i-1}} \right\rfloor$
6:    **end if**
7:    $r_{x_i} = q_{x_i} \bmod m_i$ and $r_{y_i} = q_{y_i} \bmod m_i$
8: **end for**
9: **for** $i = N$ to 1 decreasing by 1 **do**
10:    **if** $i \neq N$ **then**
11:       $D_i = T_i(r_{x_i}, r_{y_i}) + m_i \times D_{i+1}$
12:    **else**
13:       $D_i = T_i(q_{x_i}, q_{y_i})$
14:    **end if**
15: **end for**
16: $M(x, y) = D_1$

(IOI) and a pseudo ID. An anonymous IOI is not linkable to any pseudo ID, and an anonymous pseudo ID is not linkable to any IOI.

We consider an adversary that, based on the jth number of ith permutation, $M(i, j)$, tries to find the position j. Its advantage is defined as the probability that it succeeds in finding j. Our claim is that the adversary's advantage is the same as the probability of the random choice on the permutation, $1/m$.

Ideally, we assume that the constructed permuation matrix consists of random permutations. Given $M(i, j)$, the attacker's advantage is not changed regardless of the information about the previous columns since the ith column is independent of the previous columns. Also, a value x on the permutation can be uniqually positioned at any position on the permutation. Therefore the adversary's advantage is exactly $1/m$.

In the proposed system, pseudonyms are newly updated by the permutation matrix in Algorithm 2 in each round.

B. Collision-free

We show that all numbers are unique in ith column of the constructed permutation matrix. Previously we mentioned, as combining two 1-balanced PMs results in an 1-balanced PM, combination of N 1-balanced PMs can build an 1-balanced PM as described in Eq. (2). By repeating it sequentially, the finally constructed permutation matrix is also a 1-balanced PM. Therefore all numbers are unique in ith column of the permutation matrix.

Table I shows the comparison of the collisions among MASK, RINOMO and our approach. Although the system administrator cannot avoid collision due to *birthday attack* of hash function, MASK can be collision free in networks under the administrator's controlling. Since nodes generate their own

TABLE I
COLLISION AND SPACE COMPLEXITY

	MASK	RINOMO	our approach
Collision	collision exists (but controllable)	collisions exist	collision free
Required Space	$O(m) = O(\prod_{j=1}^{N} m_j)$	$O(1)$	$O(\sum_{j=1}^{N} m_j^2)$

pseudo IDs in RINOMO and there is no proper regulator, it is impossible to avoid collision. However, our approach is perfectly collision free even though nodes generates their pseudo IDs.

C. Space complexity

We compared the space complexity of a system administrator and each node among MASK, RINOMO and our proposed approach in Table I. In this table, n is the number of nodes and m is the total number of rounds. Here, m_j is a column size of the jth small matrix and N is the number of small matrices used.

D. Computational complexity

Let m be the column size of the large matrix M (e.g. M is an $m \times m$ matrix). When we assume that the size of row column of the small matrices is c, then we have $N = \log_c m$ where N is the number of small matrices. Thus, we can compute an element of a large matrix M with $\log_c m$ ($=N$) smaller matrices. According to Algorithm 2, the time complexity to access an element of M using small matrices is $O(\log_c m)$.

VI. CONCLUSION

We proposed a new framework for collision free pseudonyms among nodes in mobile ad-hoc networks. Our proposed approach achieves collision free of pseudo IDs by using a permutation matrix. A large permutation matrix is required to provide a reasonable level of anonymity. Since the mobile nodes have limited resources, the permutation matrix cannot be maintained by the mobile node. In the proposed systems, mobile nodes can compute efficiently the values of the permutation matrix not by loading such a large matrix but by loading shared small matrices. Specifically, $O(\sum_{j=1}^{N} m_j^2)$ is required in space complexity where m_j is the column size of j th small matrix and N is the number of small matrices.

In the extended version, we will experimentally show that the constructed permutation matrix has the similar properties with the ideal permutation matrix that consists of random permutations.

REFERENCES

[1] P. Mohapatra and S. Krishnamurthy. *AD HOC NETWORKS: Technologies and Protocols.* Springer, 1st edition, Sept. 2004.

[2] H. Yang, H. Y. Luo, F. Ye, S. W. Lu, and L. Zhang. Security in mobile ad hoc networks: Challenges and solutions. *IEEE Wireless Communications*, 11(1):38–47, 2004.

[3] Y. Guan, X. Fu, D. Xuan, P. U. Shenoy, R. Bettati, and W. Zhao. Netcamo: Camouflaging network traffic for qos-guaranteed mission critical applications. *IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS–PART A: SYSTEMS AND HUMANS,*, 31(4):253–265, July 2001.

[4] O. Berg, T. Berg, S. Haavik, J. Hjelmstad, and R. Skaug. *Spread Spectrum in Mobile Communication.* IET - Publisher, April 1998.

[5] S. Seys and B. Preneel. Arm: Anonymous routing protocol for mobile ad hoc networks. In *International Conference on Advanced Information Networking and Applications*, volume 2, pages 133–137, April 2006.

[6] G. Montenegro and C. Castelluccia. Statistically unique and cryptographically verifiable (sucv) identifiers and addresses. In *Proceeding of NDSS*, Feb 2002.

[7] C. Cachin and J. Camenisch. Advances in cryptology. In *EUROCRYPT 2004 -International Conference on the Theory and Applications of Cryptographic Techniques*, volume 3027. Springer, 2004.

[8] Y. Zhang, W. Liu, and W. Lou. Anonymous communications in mobile ad hoc networks. In *INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE*, volume 3, pages 1940– 1951, March 2005.

[9] S. M. M. Rahman, N. Nasser, A. Inomata, T. Okamoto, M. Mambo, and E. Okamoto. Anonymous authentication and secure communication protocol for wireless mobile ad hoc networks. *Security and Communication networks*, 1:179–189, Feb 2008.

[10] C. Ding, F. Fu, T. Klove, and V. Wei. Constructions of Permutation Arrays. *IEEE Transactions on Information Theory*, 48:977–980, 2002.

[11] L. J. Goldstein and W. W. Adams. *Introduction to Number Theory*, pages 18–19. University of Maryland, 2002.

[12] J. Kong and X. Hong. Anodr: anonymous on demand routing with untraceable routes for mobile ad-hoc networks. In *MobiHoc '03: Proceedings of the 4th ACM international symposium on Mobile ad hoc networking & computing*, pages 291–302, New York, NY, USA, 2003.