International Journal of Emerging Research in Management &Technology ISSN: 2278-9359 (Volume-6, Issue-9)

Research Article | **September 2017**

Path Reconstruction and Effective Routing with Advanced Congestion Diversity in Wireless Adhoc Networks

Dr. O. Srinivasa Rao
Associate Professor of CSE, UCEK, JNTUK, Kakinada, Andhra Pradesh, India

K. Madhuri
M.Tech (IT) Student, Dept of CSE, UCEK, JNTK, Kakinada, Andhra Pradesh, India

Abstract—

Link error and malicious packet dropping are two sources for packet losses in multi-hop wireless ad hoc network. In this paper, while observing a sequence of packet losses in the network, we are interested in determining whether the losses are caused by link errors only, or by the combined effect of link errors and malicious drop. We are especially interested in the insider-attack case, whereby malicious nodes that are part of the route exploit their knowledge of the communication context to selectively drop a small amount of packets critical to the network performance. Because the packet dropping rate in this case is comparable to the channel error rate, conventional algorithms that are based on detecting the packet loss rate cannot achieve satisfactory detection accuracy. To improve the detection accuracy, we propose to exploit the correlations between lost packets. Furthermore, to ensure truthful calculation of these correlations, we develop a homomorphic linear authenticator (HLA) based public auditing architecture that allows the detector to verify the truthfulness of the packet loss information reported by nodes. This construction is privacy preserving, collusion proof, and incurs low communication and storage overheads. To reduce the computation overhead of the baseline scheme, a packet-block based mechanism is also proposed, which allows one to trade detection accuracy for lower computation complexity. Through extensive simulations, we verify that the proposed mechanisms achieve significantly better detection accuracy than conventional methods such as a maximum-likelihood based detection.

Keywords— packet dropping, secure routing, attack detection, homomorphic linear signature, auditing.

I. INTRODUCTION

In a multi-hop wireless network, nodes cooperate in relaying/routing traffic. An adversary can exploit this cooperative nature to launch attacks. For example, the adversary may first pretend to be a cooperative node in the route discovery process. Once being included in a route, the adversary starts dropping packets. In the most severe form, the malicious node simply stops forwarding every packet received from upstream nodes, completely disrupting the path between the source and the destination. Eventually, such a severe Denial-of-Service (DoS) attack can paralyze the network by partitioning its topology. Even though persistent packet dropping can effectively degrade the performance of the network, from the attacker's standpoint such an "always-on" attack has its disadvantages [1]. First, the continuous presence of extremely high packet loss rate at the malicious nodes makes this type of attack easy to be detected. Second, once being detected, these attacks are easy to mitigate. For example, in case the attack is detected but the malicious nodes are not identified, one can use the randomized multi-path routing algorithms to circumvent the black holes generated by the attack, probabilistically eliminating the attacker's threat. If the malicious nodes are also identified, their threats can be completely eliminated by simply deleting these nodes from the network's routing table. A malicious node that is part of the route can exploit its knowledge of the network protocol and the communication context to launch an insider attack–an attack that is intermittent, but can achieve the same performance degradation effect as a persistent attack at a much lower risk of being detected. Specifically, the malicious node may evaluate the importance of various packets, and then drop the small amount that are deemed highly critical to the operation of the network [11], [12]. For example, in a frequency-hopping network, these could be the packets that convey frequency hopping sequences for network-wide frequency-hopping synchronization; in an ad hoc cognitive radio network, they could be the packets that carry the idle channel lists (i.e., white spaces) that are used to establish a network-wide control channel. By targeting these highly critical packets, the authors in have shown that an intermittent insider attacker can cause significant damage to the network with low probability of being caught [2]. In this paper, we are interested in combating such an insider attack [4]. In particular, we are interested in the problem of detecting the occurrence of selective packet drops and identifying the malicious

node(s) responsible for these drops. Detecting selective packet-dropping attacks is extremely challenging in a highly dynamic wireless environment. The difficulty comes from the requirement that we need to not only detect the place (or hop) where the packet is dropped, but also identify whether the drop is intentional or unintentional. Specifically, due to the open nature of wireless medium, a packet drop in the network could be caused by harsh channel conditions (e.g., fading, noise, and interference, a.k.a., link errors), or by the insider attacker. In an open wireless environment, link errors are quite significant, and may not be significantly smaller than the packet dropping rate of the insider attacker. So, the insider attacker can camouflage under the background of harsh channel conditions. In this case, just by observing the packet loss rate is not enough to accurately identify the exact cause of a packet loss [8]. The above problem has not been well addressed in the literature. As discussed in Section II, most of the related works preclude the ambiguity of the environment by assuming that malicious dropping is the only source of packet loss, so that there is no need to account for the impact of link errors. On the other hand, for the small number of works that differentiate between link errors and malicious packet drops, their detection algorithms usually require the number of maliciously-dropped packets to be significantly higher than link errors, in order to achieve an acceptable detection accuracy.

II. RELATED WORK

Depending on how much weight a detection algorithm gives to link errors relative to malicious packet drops, the related work can be classified into the following two categories. The first category aims at high malicious dropping rates, where most (or all) lost packets are caused by malicious dropping. In this case, the impact of link errors is ignored. Most related work falls into this category. Based on the methodology used to identify the attacking nodes, these works can be further classified into four sub-categories. The first sub-category is based on credit systems. A credit system provides an incentive for cooperation. A node receives credit by relaying packets for others, and uses its credit to send its own packets. As a result, a maliciously node that continuous to drop packets will eventually deplete its credit, and will not be able to send its own traffic. The second sub-category is based on reputation systems. A reputation system relies on neighbours to monitor and identify misbehaving nodes. A node with a high packet dropping rate is given a bad reputation by its neighbours. This reputation information is propagated periodically throughout the network and is used as an important metric in selecting routes. Consequently, a malicious node will be excluded from any route. The third sub-category of works relies on end-to end or hop-to-hop acknowledgements to directly locate the hops where packets are lost. A hop of high packet loss rate will be excluded from the route. The fourth sub-category addresses the problem using cryptographic methods. For example, the work in utilizes Bloom filters to construct proofs for the forwarding of packets at each node. By examining the relayed packets at successive hops along a route, one can identify suspicious hops that exhibit high packet loss rates. Similarly, the method in traces the forwarding records of a particular packet at each intermediate node by formulating the tracing problem as a Renyi-Ulamgame [14]. The first hop where the packet is no longer forwarded is considered a suspect for misbehaving. The second category targets the scenario where the number of maliciously dropped packets is significantly higher than that caused by link errors, but the impact of link errors is non-negligible. Certain knowledge of the wireless channel is necessary in this case. The authors proposed to shape the traffic at the MAC layer of the source node according to a certain statistical distribution, so that intermediate nodes are able to estimate the rate of received traffic by sampling the packet arrival times. By comparing the source traffic rate with the estimated received rate, the detection algorithm decides whether the discrepancy in rates, if any, is within a reasonable range such that the difference can be considered as being caused by normal channel impairments only, or caused by malicious dropping, otherwise. The works proposed to detect malicious packet dropping by counting the number of lost packets [9]. If the number of lost packets is significantly larger than the expected packet loss rate made by link errors, then with high probability a malicious node is contributing to packet losses. All methods mentioned above do not perform well when malicious packet dropping is highly selective. More specifically, for the credit-system-based method, a malicious node may still receive enough credits by forwarding most of the packets it receives from upstream nodes. Similarly, in the reputation-based approach, the malicious node can maintain a reasonably good reputation by forwarding most of the packets to the next hop [7]. While the Bloom-filter scheme is able to provide a packet forwarding proof, the correctness of the proof is probabilistic and it may contain errors. For highly selectively attacks (low packet-dropping rate), the intrinsic error rate of Bloom filer significantly undermines its detection accuracy. As for the acknowledgement-based method and all the mechanisms in the second category, merely counting the number of lost packets does not give a sufficient ground to detect the real culprit that is causing packet losses. This is because the difference in the number of lost packets between the link-error-only case and the link-error-plus-malicious-dropping case is small when the attacker drops only a few packets. Consequently, the detection accuracy of these algorithms deteriorates when malicious drops become highly selective. Our study targets the challenging situation where link errors and malicious dropping lead to comparable packet loss rates [10]. The effort in the literature on this problem has been quite preliminary, and there is a few related works. Note that the cryptographic methods proposed in to counter selective packet jamming target a different issue than the detection problem studied in this paper. The

methods in delay a jammer from recognizing the significance of a packet after the packet has been successfully transmitted, so that there is no time for the jammer to conduct jamming based on the content/importance of the packet. Instead of trying to detect any malicious behaviour, the approach in is proactive, and hence incurs overheads regardless of the presence or absence of attacker's.

III. SYSTEM MODELS AND PROBLEM STATEMENT

A. Network and Channel Models Consider an arbitrary path PSD in a multi-hop wireless ad hoc network, as shown in Figure 1. The source node S continuously sends packets to the destination node D through intermediate nodes n1, . . . , nK, where ni is the upstream node of ni+1, for $1 \leq i \leq K - 1$. We assume that S is aware of the route PSD, as in Dynamic Source Routing (DSR) [15]. If DSR is not used, S can identify the nodes in PSD by performing a trace route operation. Here we mainly focus on static or quasi-static wireless ad hoc networks, i.e., we assume that the network topology and link characteristics remain unchanged for a relatively long period of time. Recommended font sizes are shown in Table 1. Example networks include wireless mesh networks (WMNs) and ad hoc networks formed in nomadic computing. Extension to a highly mobile environment is out of our scope and will be considered in the future work.

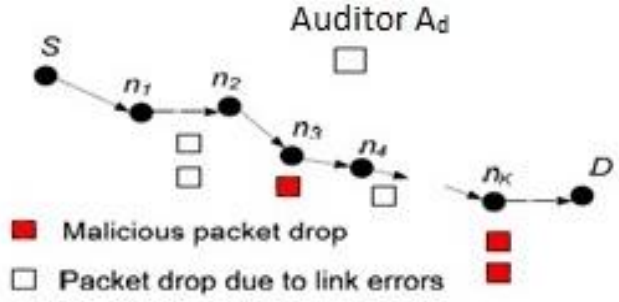


Figure 3(a): Example to illustrate the basic idea of network and attack model.

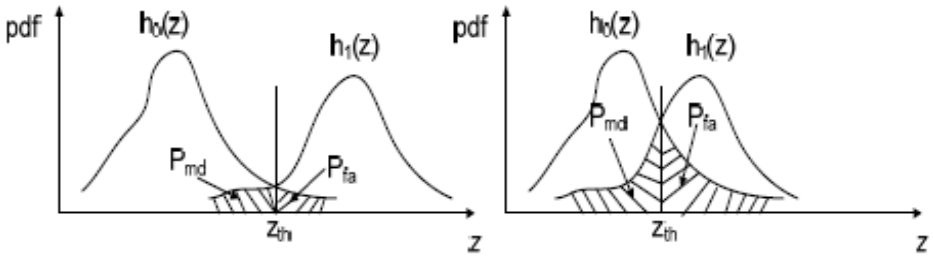
We model the wireless channel of each hop along PSD as a random process that alternates between good and bad states. Packets transmitted during the good state are successful, and packets transmitted during the bad state are lost. In contrast to the classical Gilbert-Ellioit (GE) channel model, here we do not assume any Markovian property on the channel behaviour. We only require that the sequence of sojourn times for each state follows a stationary distribution, and the autocorrelation function of the channel state, say fc(i), where i is the time lag in packets, is also stationary. Here we limit our study to quasistatic networks[10], whereby the path PSD remains unchanged for a relatively long time, so that the link error statistics of the wireless channel is a wide-sense stationary (WSS) random process (i.e., fc(i) is stationary). Detecting malicious packet drops may not be a concern for highly mobile networks, because the fast-changing topology of such networks makes route disruption the dominant cause for packet losses. In this case, maintaining stable connectivity between nodes is a greater concern than detecting malicious nodes. The function fc (i) can be calculated using the probing approach. In brief, a sequence of M packets are transmitted consecutively over the channel. By observing whether the transmissions are successful or not, the receiver obtains a realization of the channel state (a1, . . . ,aM), where aj∈ {0, 1} for j = 1, . . . , M. In this sequence, "1" denotes the packet was successfully received, and "0" denotes the packet was dropped. fc(i) is derived by computing the autocorrelation function of this sample sequence: fc(i) def = E{ajaj+i} for i = 0, . . . , M, where the expectation is calculated over all transmitted packets j = 1, . . . , M. This autocorrelation function describes the correlation between packet transmissions (successful/lost) at different times, as a function of the time lag. The time invariant nature of fc is guaranteed by the WSS assumption of the wireless channel. The measurement of fc(i) can take place online or offline. A detailed discussion on how fc(i) is derived is out of the scope of this paper, and we simply assume that this information is given as input to our detection algorithm. There is an independent auditor Ad in the network. Ad is independent in the sense that it is not associated with any node in PSD and does not have any knowledge of the secrets (e.g., cryptographic keys) held by various nodes. The auditor is responsible for detecting malicious nodes on demand. Specifically, we assume S receives feedback from D when D suspects that the route is under attack. Such a suspicion may be triggered by observing any abnormal events, e.g., a significant performance drop, the loss of multiple packets of a certain type, etc. We assume that the integrity and authenticity of the feedback from D to Scan be verified by S using resource efficient cryptographic methods such as the Elliptic Curve Digital Signature Algorithm (ECDSA). Once being notified of possible attacks, S submits an attack-detection request (ADR) to Ad. To facilitate its investigation, Ad needs to collect certain information (elaborated on in the next section) from the nodes on route PSD. We assume that each such node must reply to Ad's inquiry, otherwise the node will be considered as misbehaving. We assume that normal nodes will reply with truthful information, but malicious nodes may cheat. At the same time, for privacy reasons, we require that Ad cannot determine the content of the normal packets delivered over PSD from the information collected during the auditing.

B. Proposed Detection Scheme

Overview

The proposed mechanism is based on detecting the correlations between the lost packets over each hop of the path. The basic idea is to model the packet loss process of a hop as a random process alternating between 0 (loss) and 1 (no loss). Specifically, consider that a sequence of M packets that are transmitted consecutively over a wireless channel. By observing whether the transmissions are successful or not, the receiver of the hop obtains a bitmap (a1, . . . ,aM), where aj∈ {0, 1} for packets j = 1, . . . , M. The correlation of the lost packet is calculated as the auto-correlation function of this bitmap. Under different packet dropping conditions, i.e., link error vs. malicious dropping, the instantiations of the packet loss random process should present distinct dropping patterns (represented by the correlation of the instance). This is true even when the packet loss rate is similar in each instantiation. To verify this property, we have simulated the auto-correlation functions of two packet loss processes, one caused by 10% link errors, and the other by 10% link errors plus 10% malicious uniformly-random packet dropping. It can be observed that significant gap exists between these two auto-correlation functions. Therefore, by comparing the auto-correlation function of the observed packet loss process with that of a normal wireless channel (i.e., fc(i)), one can accurately identify the cause of the packet drops. The benefit of exploiting the correlation of lost packets can be better illustrated by examining the insufficiency of the conventional method that relies only on the distribution of the number of lost packets. More specifically, under the conventional method, malicious-node detection is modelled as a binary hypothesis test, where H0 is the hypothesis that there is no malicious node in a given link (all packet losses are due to link errors) and H1 denotes there is a malicious node in the given link (packet losses are due to both link errors and malicious drops). Let z be the observed number of lost packets on the link during some interval t. Then, z = { x, under H0 (no malicious nodes) x + y, under H1 (there is a malicious node) (1) where x and y are the numbers of lost packets caused by link errors and by malicious drops, respectively. Both x and y are random variables. Let the probability density functions of z conditioned on H0 and on H1 be h0(z) and h1(z), respectively, as shown in Figure 3(a). We are interested in the maximum-uncertainty scenario where the a priori probabilities are given by Pr{H0} = Pr{H1} = 0.5, i.e., the auditor has no prior knowledge of the distributions of H0 and H1 to make any biased decision regarding the presence of malicious nodes. Let the false-alarm and miss-detection probabilities be Pfa and Pmd, respectively. The optimal decision strategy that minimizes the total detection error Pdedef = 0.5(Pfa + Pmd) is the maximum-likelihood (ML) algorithm: { if z ≤ zth, accept H0 otherwise, accept H1 (2) where the threshold zth is the solution to the equation h0(zth) = h1(zth). Under this strategy, Pfa and Pmd are the areas of the shaded regions shown in Figure 3(a), respectively. The problem with this mechanism is that, when the mean of y is small, h1(z) and h0(z) are not sufficiently separated, leading to large Pfa and Pmd, as shown in Figure 3(b). This observation implies that when malicious packet drops are highly selective, counting the number of lost packets is not sufficient to accurately differentiate between malicious drops and link errors. For such a case, we use the correlation between lost packets to form a more informative decision statistic. To correctly calculate the correlation between lost packets, it is critical to enforce a truthful packet-loss bitmap report by each node. We use HLA cryptographic primitive for this purpose. The basic idea of our method is as follows. An HLA scheme allows the source, which has knowledge of the HLA secret key, to generate HLA signatures s1, . . . ,sM for M independent messages r1, . . . , rM, respectively. The source sends out the ri's and si's along the route. The HLA signatures are made in such a way that they can be used as the basis to construct a valid HLA signature for any arbitrary linear combination of the messages, ∑M i=1 ciri , without the use of the HLA secret key, where ci's are randomly chosen coefficients. A valid HLA signature for ∑M i=1 ciri can be constructed by a node that does not have knowledge of the secret HLA key if and only if the node has full knowledge of s1, . . . , sM. So, if a node with no knowledge of the HLA secret key provides a valid signature for ∑M i=1 ciri, it implies that this node must have received all the signatures s1, . . . , sM. Our construction ensures that si and ri are sent together along the route, so that knowledge of s1, . . . , sM also proves that the node must have received r1, . . . , rM. Our detection architecture consists of four phases: setup, packet transmission, audit, and detection. We elaborate on these phases in the next section. B. Scheme Details 1) Setup Phase: This phase takes place right after route PSD is established, but before any data packets are transmitted over the route. In this phase, S decides on a symmetric-key crypto-system (encrypt key, decrypt key) and K symmetric keys key1, . . . ,key K, where encrypt key and decrypt key are the keyed encryption and decryption functions, respectively. S securely distributes decrypt key and a symmetric key key j to node nj on PSD, for j = 1, . . . , K. Key distribution may be based on the public-key crypto-system such as RSA: S encrypts key j using the public key of node nj and sends the cipher text to nj . nj decrypts the cipher text using its private key to obtain key j . S also announces two hash functions, H1 and HMAC key, to all nodes in PSD. H1 is unkeyed while HMAC key is a keyed hash function that will be used for message authentication purposes later on. Besides symmetric key distribution, S also needs to set up its HLA keys [3]. Let e: G × G → GT be a computable bilinear map with multiplicative cyclic group G and support Zp, where p is the prime order of G, i.e., for all α, β ∈ G and q1, q2 ∈Zp, e(α q1 , βq2) = e(α, β) q1q2 . Let g be a generator of G. H2(.) is a secure map-to-point hash function: {0, 1} ∗ → G, which maps strings uniformly to G. S chooses a random number x ∈ Zp and computes v = g x . Let u

be another generator of G. The secret HLA key is sk = x and the public HLA key is a tuple pk = (v, g, u). 2) Packet Transmission Phase: After completing the setup phase, S enters the packet transmission phase. S transmits packets to PSD according to the following steps. Before sending out a packet Pi , where i is a sequence number that uniquely identifies Pi , S computes ri = H1(Pi) and generates the HLA signatures of ri for node nj , as follows sji = [H2(i||j)u ri] x , for j = 1, . . . , K (3) where || denotes concatenation. These signatures are then sent together with Pi to the route by using a one-way chained encryption that prevents an upstream node from deciphering the signatures intended for downstream nodes. More specifically, after getting sji for j = 1, . . . , K, S iteratively computes the following: s˜Ki = encrypt keyK (sKi) τKi = ˜sKi||MACkeyK (˜sKi) s˜K−1i = encryptkeyK−1 (sK−1i ||τKi) τK−1i = ˜sK−1i ||MACkeyK−1 (˜sK−1i) . . . s˜ji = encrypt key j (sji||τj+1i) τji = ˜sji||MACkeyj (˜sji) . . . s˜1i = encrypt key1 (s1i ||τ2i) τ1i = ˜s1i ||MACkey1 (˜s1i) (4) where the message authentication code (MAC) in each stage j is computed according to the hash function HMAC keyj . After getting τ1i , S puts Pi ||τ1i into one packet and sends it to node n1. When node n1 receives the packet from S, it extracts Pi , s˜1i , and MACkey1 (˜s1i) from the received packet [6]. Then, n1 verifies the integrity of s˜1i by testing the following equality: MACkey1 (˜s1i) = HMAC key1 (˜s1i). (5) If the test is true, then n1 decrypts s˜1i as follows: decryptkey1 (˜s1i) = s1i ||τ2i .



(a) mean of y much greater than mean of x (b) mean of y is comparable to mean of x

Figure 3(b): Path similarity and routing dynamics in two large-scale deployed sensor networks.

has no prior knowledge of the distributions of H0 and H1 to make any biased decision regarding the presence of malicious nodes. Let the false-alarm and miss-detection probabilities be Pfa and Pmd, respectively. The optimal decision strategy that minimizes the total detection error Pdedef = 0.5(Pfa + Pmd) is the maximum-likelihood (ML) algorithm: { if z ≤ zth, accept H0 otherwise, accept H1 (2) where the threshold zth is the solution to the equation h0(zth) = h1(zth). Under this strategy, Pfa and Pmd are the areas of the shaded regions shown in Figure 3(a), respectively. The problem with this mechanism is that, when the mean of y is small, h1(z) and h0(z) are not sufficiently separated, leading to large Pfa and Pmd, as shown in Figure 3(b). This observation implies that when malicious packet drops are highly selective, counting the number of lost packets is not sufficient to accurately differentiate between malicious drops and link errors. For such a case, we use the correlation between lost packets to form a more informative decision statistic.

Table I Performance Comparison

TCP Packet size bytes	**CM Disabled**	**CM Enabled**
	Good put (%)	
512	87.36	95.10
1536	70.57	93.04
4352	66.17	92.59
9180	60.47	89.92

IV. CONCLUSION

In this paper, we showed that compared with conventional detection algorithms that utilize only the distribution of the number of lost packets, exploiting the correlation between lost packets significantly improves the accuracy in detecting malicious packet drops. Such improvement is especially visible when the number of maliciously dropped packets is comparable with those caused by link errors. To correctly calculate the correlation between lost packets, it is critical to acquire truthful packet-loss information at individual nodes. We developed an HLA-based public auditing architecture that ensures truthful packet-loss reporting by individual

nodes. To reduce the computation overhead of the baseline construction, a packet-block-based mechanism was also proposed, which allows one to trade detection accuracy for lower computation complexity. As a first step in this direction, our analysis mainly emphasize the fundamental features of the problem, such as the untruthfulness nature of the attackers, the public verifiability of proofs, the privacy preserving requirement for the auditing process, and the randomness of wireless channels and packet losses, but ignore the particular behaviour of various protocols that may be used at different layers of the protocol stack. The implementation and optimization of the proposed mechanism under various particular protocols will be considered in our future studies.

REFERENCES

[1] K. Ren, C.Wang, Q.Wang*et al.*, "Security challenges for the public cloud," *IEEE Internet Computing*, vol. 16, no. 1, pp. 69–73, 2012.

[2] J. Katz, A. Sahai, and B. Waters, "Predicate encryption supporting disjunctions, polynomial equations, and inner products," in *Advances in Cryptology–EUROCRYPT 2008*. Springer, 2008, pp. 146–162.

[3] S. Kamara and C. Papamanthou, "Parallel and dynamic searchable symmetric encryption," in *Financial Cryptography and Data Security*. Springer, 2013, pp. 258–274.

[4] O. Goldreich and R. Ostrovsky, "Software protection and simulation on oblivious rams," *Journal of the ACM (JACM)*, vol. 43, no. 3, pp. 431–473, 1996.

[5] D. Boneh, G. Di Crescenzo, R. Ostrovsky, and G. Persiano, "Public key encryption with keyword search," in *Advances in Cryptology-Eurocrypt 2004*. Springer, 2004, pp. 506–522.

[6] C. Orencik, M. Kantarcioglu, and E. Savas, "A practical and secure multi-keyword search method over encrypted cloud data," in *Cloud Computing (CLOUD), 2013 IEEE Sixth International Conference on*. IEEE, 2013, pp. 390–397.

[7] S. Zerr, D. Olmedilla, W. Nejdl, and W. Siberski, "Zerber+ r: Topk retrieval from a confidential index," in *Proceedings of the 12th International Conference on Extending Database Technology: Advances in Database Technology*. ACM, 2009, pp. 439–449.

[8] E.-J. Goh*et al.*, "Secure indexes." *IACR Cryptology ePrint Archive*, vol. 2003, p. 216, 2003.

[9] Y.-C. Chang and M. Mitzenmacher, "Privacy preserving keyword searches on remote encrypted data," in *Proceedings of the Third international conference on Applied Cryptography and Network Security*. Springer-Verlag, 2005, pp. 442–455.

[10] C. Wang, N. Cao, K. Ren, and W. Lou, "Enabling secure and efficient ranked keyword search over outsourced cloud data," *Parallel and Distributed Systems, IEEE Transactions on*, vol. 23, no. 8, pp. 1467–1479, 2012.

[11] J. Li, Q. Wang, C. Wang, N. Cao, K. Ren, and W. Lou, "Fuzzy keyword search over encrypted data in cloud computing," in *INFOCOM, 2010 Proceedings IEEE*. IEEE, 2010, pp. 1–5.

[12] N. Cao, C. Wang, M. Li, K. Ren, and W. Lou, "Privacy-preserving multi-keyword ranked search over encrypted cloud data," in *IEEE INFOCOM*, April 2011, pp. 829–837.

[13] C. Wang, K. Ren, S. Yu, and K. M. R. Urs, "Achieving usable and privacy-assured similarity search over outsourced cloud data," in *INFOCOM, 2012 Proceedings IEEE*. IEEE, 2012, pp. 451–459.

[14] B. Wang, S. Yu, W. Lou, and Y. T. Hou, "Privacy-preserving multikeyword fuzzy search over encrypted data in the cloud," in *IEEE INFOCOM*, 2014.

[15] P. Golle, J. Staddon, and B. Waters, "Secure conjunctive keyword search over encrypted data," in *Applied Cryptography and Network Security*. Springer, 2004, pp. 31–45.

[16] Y. H. Hwang and P. J. Lee, "Public key encryption with conjunctive keyword search and its extension to a multi-user system," in *Proceedings of the First international conference on Pairing-Based Cryptography*. Springer-Verlag, 2007, pp. 2–22.

[17] L. Ballard, S. Kamara, and F. Monrose, "Achieving efficient conjunctive keyword searches over encrypted data," in *Proceedings of the 7th international conference on Information and Communications Security*. Springer-Verlag, 2005, pp. 414–426.

[18] D. Boneh and B. Waters, "Conjunctive, subset, and range queries on encrypted data," in *Proceedings of the 4th conference on Theory of cryptography*. Springer-Verlag, 2007, pp. 535–554.

[19] B. Zhang and F. Zhang, "An efficient public key encryption with conjunctive-subset keywords search," *Journal of Network and Computer Applications*, vol. 34, no. 1, pp. 262–267, 2011.

[20] E. Shen, E. Shi, and B. Waters, "Predicate privacy in encryption systems," in *Proceedings of the 6th Theory of Cryptography Conference on Theory of Cryptography*. Springer-Verlag, 2009, pp. 457–473.

[21] A. Lewko, T. Okamoto, A. Sahai, K. Takashima, and B. Waters, "Fully secure functional encryption: attribute-based encryption and (hierarchical) inner product encryption," in *Proceedings of the 29th Annual international conference on Theory and Applications of Cryptographic Techniques*. Springer-Verlag, 2010, pp. 62–91.

[22] A. Swaminathan, Y. Mao, G.-M. Su, H. Gou, A. L. Varna, S. He, M.Wu, and D.W. Oard, "Confidentiality-preserving rank-ordered search," in *Proceedings of the 2007 ACM workshop on Storage security and survivability*. ACM, 2007, pp. 7–12.

