Quantifying Security in Hybrid Cellular Networks

Markus Jakobsson[1] and Liu Yang[2]

[1] School of Informatics, Indiana University,
Bloomington, IN 47406, USA
markus@indiana.edu
[2] Software Engineering College, Sichuan University,
Chengdu, 610065, P.R.China
yangliutww@gmail.com

Abstract. We propose a micro-payment scheme for symmetric multi-hop cellular networks that encourages intermediaries to transmit packets and recipients to provide auditing information. Our scheme is an extension of [6], where the authors addressed the simpler asymmetric case. We then analyze the possible rational abuses to our protocol, and construct a detailed statistical model to detect attacks. We show that for typical applications, the statistical detection model we develop will detect cheating very rapidly. For example, in a VoIP application, one particular attack will be detected within a dozen of seconds.

1 Introduction

Multi-hop cellular networks [1, 13] is a new and promising paradigm for wireless communication. Several benefits [8, 5, 9, 7] can be expected from the use of this approach. For one thing, the energy consumption of the mobile device can be reduced, as the distance the signal has to cover is smaller. For another, the interference between nodes is reduced, which means more bandwidth. And then, the number of fixed antennas can be reduced, because packets travel between the user and base station by one or more hops. Finally, the coverage of the network can be increased. To make the communication between users go smoothly in such a scheme, users need to collaborate with each other by transmitting packets for others.

One approach to encourage *intermediaries* to collaborate in packet forwarding is to pay them. A micro-payment scheme based on [10] was proposed by Jakobsson, Hubaux, and Buttyán [6] to encourage collaboration in multi-hop cellular networks. The networks they considered consist of cellular nodes, base stations (connected to the backbone), and accounting and auditing centers. If an originator wants to send some packets to a recipient, each packet first needs to travel by one or more hops to the base station which the *originator* belongs to, then the packet is forwarded to the *recipient's* base station, after which this base station transmits the packet to the recipient by one hop. Such a structure is called an asymmetric cellular network.

Jakobsson, Hubaux, and Buttyán [6] suggested using a micro-payment scheme influenced by the work of Micali and Rivest [10] to allow payments for intermediaries who help forwarding packets. Therein, an intermediary is paid by checking whether packets it handles corresponds to winning tickets. Thus, instead of being paid per packet, intermediaries are given a "lottery ticket" for each packet. This allows for global aggregation of payment information, since intermediaries only record the winning tickets, and only forward the same to the accounting/auditing center. This approach is more efficient than being paid per packet because of the reduced cost of processing total payments. As records are being aggregated, though, detection of abuse becomes more challenging. In [6] , the notion of attack *footprints* was proposed; a footprint is a statistical aberration of audit information that can be associated with a given form of abuse. More in detail, they are ways to infer the likely occurrence of specific attacks by comparing a user's frequency as claimant, sending neighbor, and receiving neighbor, for some attacks also taking into consideration the frequency with which a potential offender is reported as sender vs. recipient by other nodes. While these footprints appear likely to be theoretically sound, it is not evident that they are practically meaningful. In other words, it is not clear that one could determine the presence of an attack (with reasonably small error margins) within a reasonable amount of time. Another potential drawback of [6] is the use of an asymmetric communication model. Namely, the scheme proposed therein uses an unusual multi-hop up / single-hop down communication model, which introduces routing complexities.

1.1 Our contributions

Our first contribution is to extend the micro-payment scheme in [6] from the asymmetric case to the symmetric case, i.e. to a setting in which both the uplink and downlink are one or more hops. Here, *uplink* refers to the routing of a packet from an *originator* to the nearest base station; and *downlink* from the recipient's base station to the *recipient*. The symmetric case is fraught with two difficulties. First, the nodes on the downlink need to know where to send the packet (in the uplink it is always to the closest base station). Second, while the base station knows about all packets on the uplink (since it receives them), it does not automatically know what packets were successfully received by the recipient on the downlink. This is not a problem of the same magnitude if the downlink is one-hop, since then there is no risk a packet will be dropped on the route. If the base station knows nothing about the downlink transmission, cheating is undetectable. In our design, we not only encourage *intermediaries* to transmit packets, but also encourage recipient to report receipt of packets by checking winning tickets, and thereby address these problems. This constitutes an extension of the techniques proposed in [6].

The second contribution is to construct an efficient auditing model. We propose the following view of the packet flow: If two intermediaries transmit the same number of packets within a certain time interval, they are expected to get the same number of winning tickets on average. This is because each packet

corresponds to a winning ticket with the same probability. (Note that one intermediary will win with a probability independent of whether other intermediaries win on the same packet.) Whether a ticket is winning or not depends on secret information associated with an intermediary, as well as on the contents of the packet. Because a winning intermediary also reports his *sending neighbor* – the neighbor who sent him the packet with the winning ticket, and the *receiving neighbor* – the neighbor he sent the packet to, we can expect that the frequency of an intermediary as winning claimant equal his frequencies as sending and receiving neighbor (in which case others will report the event.) After extending the analysis of [6] to the symmetric setting, and analyzing the characteristics of different attacks, we construct statistical models to detect them, and obtain the confidence intervals of some parameters related with winning frequency. Some attacks can be detected by hypothesis testing. Our construction allows for the detection or prevention of the following attacks: packet dropping, originator and recipient collusion, selective acceptance, packet tampering, intermediary credits a friend, ticket sniffing, greedy ticket collection, and packet replay. The simulation results show our method is practical, effective, and efficient.

Our third contribution is that we allow the recipient to verify authenticity of all packets without having to share a key with the sender, and without the use of costly public key operations. This is achieved by making each node share a key with its home base station (which has to be done anyway, for charging purposes.) The uplink message is authenticated by the originator's base station; and the downlink base station computes a MAC with a key shared with the recipient – thus, the recipient can verify the packet he received. Apart from being a beneficial feature to users, the authentication is also indirectly useful as part of the auditing mechanism.

1.2 Related Work

Probabilistic micropayments based on the use of electronic lottery tickets were first suggested by Rivest [11]. To illustrate the idea, a lottery ticket for a $1.00 prize with a 1/100 winning rate has the expected value of one cent. A payer can pay somebody one cent by giving him such a lottery ticket. Since the bank only needs to process winning tickets, this significantly reduces its effort, and the amount of storage and communication needed for both the winner and the bank. In [11], payers will be billed when the lottery tickets they issue win. This creates a psychological disincentive to use the scheme; an alternative version without this drawback was later proposed by Micali and Rivest [10].

Instead of using one payment token *per payee* (as is done in traditional micropayment schemes [10, 11]), Jakobsson, Hubaux, and Buttyán [6] used one token *per packet*, letting each relaying node verify whether this token corresponds to a winning ticket for him. In their scheme, payers (i.e., originators of packets) are charged per packet, and not per winning ticket, while users performing packet forwarding are paid per winning ticket. While there is only one payee per payer in traditional payment schemes, *each* node on a route in [6] may win for a ticket associated with one specific packet, which means one packet may result in more

than (or less than) one winning ticket. The originator pays a cost that – on average – covers the cost of routing as well as other network maintenance. The cost an originator needs to pay for sending out a packet is a fixed value which does not depend on the number of winning tickets associated with this packet. Such an approach is a generalization of the averaging techniques proposed in [10].

Jakobsson, Hubaux, and Buttyán's work [6] is a packet based scheme, while Ben Salem, Buttyán, Hubaux, and Jakobsson considered a session based scheme in multi-hop cellular networks [12]. The latter differs from [6] in several aspects. The session based scheme needs exchanging some packets to authenticate users and set up a session, which results in a certain amount of overhead. If the nodes in a cell are highly mobile, the chance a session to be broken is high. Thus this scheme is suitable only in relatively stable networks. The packet based scheme has much less overhead for short sessions and can be used in networks with poor connection or large topology changes.

In [2], Avoine analyses potential weaknesses in the scheme by [6]. Some of these weaknesses relate to a modified version of [6], namely one in which transcripts are not encrypted before being transmitted. The protocol we propose herein inherits the general properties of [6], and exhibits the same vulnerabilities as well. However, these are easily overcome by the use of encryption of transcripts (as specified in [6]), along with an appropriate choice of the function used to determine what tickets win.

Like in [6], our approach is a packet based scheme, which is different from [12]. While [12] charges the recipient, we instead *pay* the recipient to encourage it to report receipt of packet, which is essential to detect cheating. (This payment simply becomes part of the charge to the packer originator.)

1.3 Outline

We begin by describing our work in general and covering some related work in the first part, and then detail our trust and communication models in section 2. In section 3, we describe the protocols for routing, trasmission, and reward recording. Abuse detection and simulation results are described in section 4. Details of our simulation are described in the Appendix.

2 Model

2.1 Trust Model

User model. Like in [6], we assume the existence of three types of participants: users, base stations, and one or more accounting and auditing centers. In addition, there may be multiple networks, each one of them considered the home network for some users.

Functional model. As in [6], users can be categorized to belong to one or more of the following classes: originators; recipients; and intermediaries.

Trust model. In our context, a node and a user refers to the same entity. We consider the user as a software module running on a multi-purpose computer, with an appropriate communicating module. Users, which can be modeled as polynomial time Turning Machines, may cheat in an arbitrary but rational manner: Users only abuse the protocol when they can benefit from doing so. Users trust base stations of their home network not to disclose their secret information; no such trust has to be placed on base stations outside their home network. All base stations are trusted to correctly transmit packets, and to forward billing and auditing information to the accounting center of the user's home network. Because the users are assumed to be selfish, and because they may depart from the protocol if they can benefit from it, we have that the base station cannot trust users to be honest in terms of routing, packet transmission, and reward claiming. Users do not trust each other, either. The accounting center, finally, is trusted by all to correctly perform billing and auditing.

Rewarding model. Our protocol encourages the intermediaries to transmit packets, and also encourages both intermediaries and the recipient to provide auditing information to the auditing center. This is achieved using micro-payments. Similar to the proposal by Micali and Rivest, our remuneration technique works by a probabilistic selection of payment tokens. A winning user records the ticket as well as his sending neighbor (where packet comes from) and receiving neighbor (where packet is sent to). Like in [6], all these three users are rewarded. The recorded claims are sent to the base station; the latter forwards all claims to accounting center periodically.

Abuse. Intermediaries may cheat if this could increase their profit or lower their effort. Thus, we need to consider detecting all possible rational attacks when designing our protocol. In particular, we prevent or detect the following abuses.

- *Packet dropping.* An intermediary agrees to forward packets, but does not retransmit them – whether he claims credit for winning tickets or not.
- *Originator and recipient collusion.* An originator helps a recipient gain a profit by sending packets with winning tickets. For example, an originator constructs a packet and checks if it contains a winning ticket for the recipient (by knowing the secret key of the recipient). If this is the case then he sends out the packet to the recipient; If no, then he makes a minor modification and checks again, until the packet contains a winning ticket to the recipient.
- *Selective acceptance.* An intermediary agrees to receive and forward packets with winning tickets, but not packets without winning tickets. (This attack can be combined with collusion between the originator and the intermediary, and has the same footprint on a per-originator basis; due to lack of space, we do not consider this special case in our description.)
- *Packet tampering.* An intermediary changes the payload of a packet before re-transmitting it.

- *Intermediary credits a friend.* An intermediary with a winning ticket claims to have received the packet from (or have sent to) a user different from the one he actually received it from (resp. sent it to).
- *Ticket sniffing.* A user claims credit for packets he intercepted, but neither agreed to re-transmit nor actually re-transmitted.
- *Greedy ticket collection.* This is a collection of cheating strategies which users try to claim credits in excess of what the protocol specifies – by collecting and sharing tickets with colluders.
- *Packet replay.* A user checks a packet and finds that it contains a winning ticket. Then he makes a copy of the packet and replays it one or more times to increase his profit.

We do not consider the communication attack mentioned in [2], since the attacker benefits nothing from performing it.

2.2 Communication Model

We consider a symmetric multi-hop hybrid network. In other words, as a packet travels from the originator to the receiver, it is transmitted in one or more hops to the closest base station. Then the packet is sent over the backbone to the base station where the recipient resides. The base station of the recipient transmits the packet to the recipient in one or more hops.

The routing of uplink transmission can be performed by either of the two approaches. In one of these, the base station infers topology changes from observed communications and signals, then maintains and propagates routing tables. Route discovery information of each cell is maintained and stored by each local base station, and the updates of routing information are sent to all local nodes periodically. In another approach, routing tables are maintained by the users themselves. Here, it is worth noting that uplink routing is relatively easy because there is only one destination, the base station. However, downlink routing is more complex. Therefore, we assume that the downlink routing is performed by the base station of the corresponding cell. We therefore use a source routing protocol (e.g., Dynamic Source Routing [3]) for the downlink, and a packet sent out by an originator will carry full routing information. In other words, the routing tables of users do not have to take downlink routing into consideration – this also translates into a privacy benefit as general user location data does not have to be propagated to peers.

3 Protocol

Setup. When a user registers to access the home network, he is assigned a pair (id_u, K_u), where id_u is its identity and K_u is a symmetric key.

Packet origination. First of all, the originator u_o of a packet p selects a route path r_p; then he calculates $\mu = MAC_{K_{u_o}}(p)$; finally, he assembles a tuple (p, u_o, u_r, r_p, μ) and transmits it to the next node indicated by the r_p. Here, u_r is the final recipient, and r_p is the route to the closest base station.

Transmission. For both uplink and downlink, each intermediary transmits the packet tuple according to r_p (resp. the downlink route, r'_p).

Network processing. When a packet $P = (p, u_o, u_r, r_p, \mu)$ is received by a base station in the originator's home network, the base station looks up the secret key K_{u_o} of the originator and verifies whether $\mu = MAC_{K_{u_o}}(p)$, dropping the packet if it does not hold.

If the verification of MAC succeeds, the base station (resp. home network register) transmits the packet portion p to the base station associated with the desired recipient, u_r.

The recipient's base station first selects a downlink route r'_p from its routing table or routing cache, then he looks up the recipient's secret key K_{u_r} and calculates $\mu_r = MAC_{K_{u_r}}(p)$. He then assembles a tuple (p, u_r, r'_p, μ_r) and transmits the tuple to the first downlink hop according to the routing path r'_p.

Packet receiving. When a recipient receives a tuple (p, u_r, r'_p, μ_r), he verifies its authenticity by calculating $MAC_{K_{u_r}}(p)$ and verifying whether this equals μ_r. If this holds, then he accepts the packet; otherwise he drops it. The recipient also checks if this was a winning ticket by performing the reward recording protocol.

Reward recording. After an intermediary u has forwarded a tuple $P = (p, u_o, u_r, r_p, \mu)$ (resp. $P = (p, u_r, r'_p, \mu_r)$), he verifies whether $f(\mu, K_u) = 1$ (resp. $f(\mu_r, K_u) = 1$) for some function f (we do not discuss the choice of f in this paper). If this holds, it means that the considered ticket is winning; he then records (u_1, u_2, μ) (resp. (u_1, u_2, μ_r)), where u_1 is the identity of the user (or base station) he received the the packet from, and u_2 is the identity of the user (or base station) he forwarded it to. To a recipient who gets a winning ticket, he just records (u_1, μ), where u_1 is the node who sent him this packet. We use M to denote the list of recorded reward tuples. u_1 is called the *sending neighbor*, and u_2 is referred to as the *receiving neighbor*.

Reward claim. A user periodically transmits claims (u, M, μ') to the base station, where $\mu' = MAC_{K_u}(hash(M))$. When a base station receives a claim, it verifies the correctness of μ' with the key K_u of user u. If the MAC is correct, then it records the claim and computes an acknowledgement $ack = MAC_{K_u}(\mu')$ that is sent to u; otherwise it ignores the claim. When u receives the ack, he verifies the ack and erases M if (but only if) it is correct. Within a time Δ, each base station forwards all recorded claims to an accounting center, and then erases the list.

4 Abuse Detection and Simulation

4.1 Flow Equations and Confidence Intervals

Not all users will collect the same expected number of winning tickets due to several factors, such as the users' location, energy, and power state (on or off). However, if we consider the packet flow (which can be approximated using the rate of winning tickets), we see that the flow-in should be equal to the flow-out, where flow-in denotes the number of packets sent to this node, and flow-out denotes that sent out by this node. Significant divergence in terms of the reported flows is indicative of network problems or attacks. The flows can be approximated using the claims for winning tickets. We use hypothesis testing to detect cheating.

Our attacks detecting approach is based on the following equalities (We suppose the probability of a node to get a winning ticket is s):

To a given cell within a certain period, the number of packets sent out by an originator (resp. base station in downlink) times the winning rate s is expected to be equal to the number of winning tickets reported by the recipient. In particular, to the downlink transmission:

$$N_{B_r} \cdot s = \tilde{w}_r \tag{1}$$

where $\tilde{w}_r$ is the expected value of winning frequency w_r of the recipient. The form of the equation is the same for the uplink transmission. (In the following statements, we focus on downlink transmission, noting that the uplink can be analyzed in the same way.)

Since each intermediary along a path has an equal probability to obtain a winning ticket, the winning frequency of each intermediary is expected to be equal to its frequency being reported as receiving neighbor, and also be equal to that being reported as sending neighbor.

$$\tilde{w}_i = \tilde{r}_i = \tilde{s}_i \tag{2}$$

Here, r_i is the frequency of u_i being reported as receiving neighbor; and s_i is the frequency of u_i being reported as sending neighbor. If someone reports a winning frequency either too high or too low, this indicates cheating may exist during the transmission. We will show that most attacks violate at least one of the above equalities.

Confidence Intervals. Equation (1) and (2) hold probabilistically speaking, but may fail to do so in the short run. Thus, there might exist a difference between any two of w_i, r_i, and s_i for user u_i. To all users in a given cell, we denote $x_i = s_i - w_i$, $y_i = w_i - r_i$, $z_i = r_i - s_i$, and assume that x, y, and z are three random variables which satisfy the same normal distribution $N(0, \sigma)$, where σ is the standard deviation, and 0 is the expected value of x, y, and z.

To evaluate if all nodes in a given cell work well or not, we give out confidence intervals of variables x, y, and z which can be used for hypothesis testing to detect attacks. The process is as follows:

1. Make hypothesis $H_0 : e = 0$, equation (2) holds; and $H_1 : e \neq 0$, equation (2) was voilated; where e is the expected value of x, y, or z.
2. Construct statistics $t_x = \frac{\overline{x}-e}{S_x/\sqrt{n}}$, $t_y = \frac{\overline{y}-e}{S_y/\sqrt{n}}$, and $t_z = \frac{\overline{z}-e}{S_z/\sqrt{n}}$ satisfying a $t - distribution$ with freedom of $n-1$, where n is the number of samples, and $S_x^2 = \frac{1}{n-1}\sum(x_i - \overline{x})^2$, $S_y^2 = \frac{1}{n-1}\sum(y_i - \overline{y})^2$, $S_z^2 = \frac{1}{n-1}\sum(z_i - \overline{z})^2$ are the square deviations of samples.
3. Choose a value of level of significance α and calculate the confidence interval of e:
$$CI_x = (-\frac{S_x}{\sqrt{n}}t_0, \frac{S_x}{\sqrt{n}}t_0); CI_y = (-\frac{S_y}{\sqrt{n}}t_0, \frac{S_y}{\sqrt{n}}t_0); CI_z = (-\frac{S_z}{\sqrt{n}}t_0, \frac{S_z}{\sqrt{n}}t_0) \quad (3)$$
4. Calculate sample means $\overline{x} = \frac{1}{n}\sum x_i$, $\overline{y} = \frac{1}{n}\sum y_i$, and $\overline{z} = \frac{1}{n}\sum z_i$. If $\overline{x} \in CI_x$, $\overline{y} \in CI_y$, and $\overline{z} \in CI_z$, we accept H_0; otherwise we reject H_0, which means abuses might have occurred during the transmission.

In the following statements, we show that all practical attacks we are aware of will violate at least one of the above equations or confidence intervals.

4.2 Analysis

Packet dropping. If an intermediary u_i consistently performs packet dropping, then the intended recipients will receive less packets than expected, and thus, will report fewer winning tickets; u_i will have a higher *claimant* frequency than being reported as *sending neighbor*. This can be expressed as the following model:

$$N_{B_r} \cdot s > w_r \quad (4)$$
$$w_i > s_i \quad (5)$$
$$\tilde{r}_i = \tilde{w}_i \quad (6)$$

equation (1) and (2) are violated. Inequalities (4) to (6) are equivalent to:

$$\frac{N_{Br} \cdot s}{w_r} \geq \delta_0;\ \overline{y} \in CI_y;\ \overline{x} < -\frac{S_x}{\sqrt{n}}t_0 \quad (7)$$

where $\delta_0 > 1$ is a threshold chosen by the auditing center. If $\frac{N_{Br} \cdot s}{w_r}$ exceeds this threshold, the difference between $N_{Br} \cdot s$ and w_r can not be ignored. The higher value we choose for δ_0, the more certain we are to declare a cheating. The above (7) shows: the recipient's winning frequency is unusually much lower than expected; $\overline{y}$ is still in its confidence interval; and $\overline{x}$ is smaller than the lower bound of its confidence interval.

A VoIP Example Assume the nodes are running a VoIP application using G.711 Codec($Rate = 64kbit/s$) with a frame size (including the headers) of 200 bytes [4]. Then, an originator is able to send out 40 packets per second. If at least 40 packets need to be sent to detect the dropping, then the minimum time to detect this attack is 1 second. The minimum time t can be calculated by $t = \frac{minN_{sent}}{r}$, where $minN_{sent}$ denotes the minimum number of packets needed to be sent to detect the attack, and r is the packet rate of a node.

We use the above VoIP example to demonstrate the relationship among α, s, and $minN_{sent}$ (Figure 1). The second vertical axis in the figure indicates the time needed to detect this attack. We take standard deviation $\sigma = 5$, and $n = 16$ in the simulation. The details of the calculation are described in Appendix. Figure 1 shows:

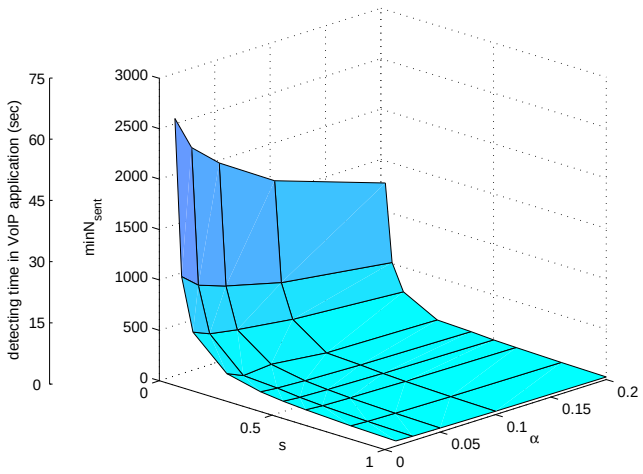

Fig. 1. Simulation of detecting packet dropping

- To a given point $D = (s, \alpha, N)$ in the 3-D space, if $\frac{N.s}{w_r} \geq \delta_0$, then one or more intermediaries are suspected of packet dropping; otherwise no dropping attack is believed to have occurred.
- Given a winning rate s, the minimum number of packets $minN_{sent}$ needed to be sent increases as the level of significance α decreases. This indicates that to detect packet dropping with higher accuracy (corresponding to higher $1 - \alpha$), more packets need to be sent.
- Given a value of α, $minN_{sent}$ increases greatly as s decreases (proportional to $\frac{1}{s}$). This indicates that the smaller the winning rate is, the more packets need to be sent to detect a packet dropping attack.

Remark 1. Selective or probablistic packet dropping also results in a smaller number of winning tickets reported by the recipient, which corresponds to a case described in the above model.

Originator and recipient collusion. If an originator benefits a recipient by sending the latter winning tickets, then the recipient will have an unusually high winning rate. Equation (1) will be violated, but equation (2) still holds:

$$N_{B_r} \cdot s < w_r \tag{8}$$

$$\tilde{w}_i = \tilde{r}_i = \tilde{s}_i \tag{9}$$

Formulae (8) and (9) can be verified by the following conditions:

$$\frac{w_r}{N_{Br} \cdot s} \geq \delta_0 \tag{10}$$

$$\overline{x} \in CI_x, \overline{y} \in CI_y, \overline{z} \in CI_z \tag{11}$$

where $\delta_0 > 1$ is a threshold chosen by the auditing center. From (10) we also get $\delta_0 \cdot s \leq 1$ (because $w_r \leq N_{B_r}$).

Inequality (10) can be rewritten as $N_{Br} \leq \frac{w_r}{s \cdot \delta_0}$. We use an example to illustrate it: suppose $s = 0.05$, $\delta_0 = 1.3$, and $w_r = 20$, then we get $\frac{w_r}{s \cdot \delta_0} = 308$, which means if a recipient reports 20 or more (much higher than 308×0.05) winning tickets within 308 packets, then the originator and the recipient are suspected of colluding. The relationship among w_r, s, and N_{B_r} is shown in Figure 2 with the same VoIP example. The second vertical axis indicates the time needed to detect such an attack. Figure 2 shows:

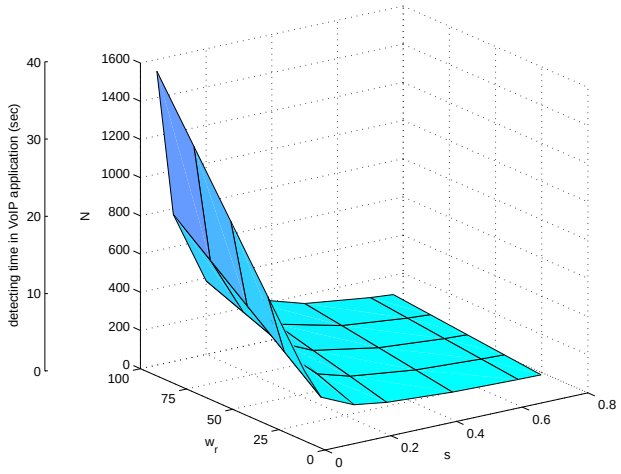

Fig. 2. Simulation when originator and recipient are colluding

- To a given point $D = (w_r, s, N)$ in the 3-D coordinate system, if D is located in the upper side of the surface, no colluding attack is believed to have occurred; otherwise the originator and the recipient are suspected as colluding.
- The number of packets needed to be transmitted to detect this attack is proportional to w_r and $\frac{1}{s}$.

Remark 2. We note even if the cheaters only collude some of the time, the recipient will still report a higher number of winning tickets than a expected value ($N_{Br} \cdot s$). Thus this is a detectable case by our above model.

Other attacks. Selective acceptance does not work when the routing is under the control of the base station. Packet tampering is easy to be detected because: (1) both the base station of originator and recipient will verify the MAC of a packet; and (2) the base stations know the routing path. Intermediaries can not easily credit selected friends because they can not choose the routing path. This also affects the greedy ticket collection attack. Collusion between sender and intermediaries can be handled in the same way as collusion between sender and receiver. Ticket sniffing will be easily detected because the attacker will report a fake routing path different to that selected by the base station during a particular transmission. Packet replay will be traced because the attacker will have a too high winning frequency and the base station knows the routing path. Sequence numbers can also be used to avoid this attack.

5 Conclusion

We have extended the asymmetric micro-payment based routing scheme of [6] to the symmetric case, where both the uplink and downlink transmission are one or more hops. Our architecture not only encourages the intermediaries to transmit packets, but also encourages the recipient to report receipt of packet by checking for winning tickets. To discourage and detect dishonest users, we have proposed statistical models based on the perspective of packet flow. The conditions of different attacks in our models are easy to be verified by the auditing center. Our simulation results indicate that abuses will be detected within a short time (ranges from dozens of seconds to a couple of minutes) for common applications. Thus, our model is realistic and meaningful. Due to the space limit, we have not discussed abuse detection when routing is not controlled by the base station. This remains an open problem how to properly address, in particular given the increased complexity of this setting.

References

1. Aggélou G. N., Tafazolli R.: On the Relaying Capacity of Next-Generation GSM Cellular Networks. IEEE Personal Communications, February (2001)
2. Avoine G.: Fraud within Asymmetric Multi-Hop Cellular Networks. accepted by Financial Cryptography (2004)
3. Broch J., Johnson D. B., Maltz D. A.: The Dynamic Source Routing Protocol for Mobile Ad-Hoc Networks. Internet Draft, draft-ietf-manet-dsr-03.txt (1999)
4. Goode B.: Voice Over Internet Protocol (VoIP). Proceedings of the IEEE, Vol. 90. (2002) 1495-1517
5. Hsieh H.-Y., Sivakumar R.: Towards a Hybrid Network Model for Wireless Packet Data Networks. Proceedings of ISCC. IEEE (2002)
6. Jakobsson M., Hubaux J. P. , Buttyan L. : A Micro-payment Scheme Encouraging Collaboration in Multi-hop Cellular Networks. Proceeding of Financial Cryptography (2003)
7. Kubisch M., Mengesha S., Hollos D., Karl H., Wolisz A.: Applying Ad-hoc Relaying to Improve Capacity, Energy Efficiency, and Immission in Infrastructure-based

WLANs. Proceedings of Kommunikation in Verteilten Systemen. Leipzig, Germany, KiVS (2003)
8. Lin Y.-D., Hsu Y.-C.: Multihop Cellular: A New Architecture for Wireless Communications. Proceedings of INFOCOM (2000)
9. Mantel O. C., Scully N., Mawira A.: Radio Aspects of Hybrid Wireless Ad Hoc Networks. Proceedings of VTC. IEEE (2001)
10. Micali S., Rivest R.: Micropayments Revisited. CT-RSA (2002) 149–163
11. Rivest R.: Electronic Lottery Tickets as Micropayments. Financial Cryptography (1997) 307–314
12. Salem N. B., Buttyan L., Hubaux J.P., Jakobsson M.: A Charging and Rewarding Scheme for Packet Forwarding in Multi-hop Cellular Networks. Annapolis, MD, USA, MobiHoc(2003)
13. Zadeh A. N., Jabbari B., Pickholtz R., and Vojcic B.: Self-Organizing Packet Radio Ad Hoc Networks with Overlay (SOPRANO). IEEE Communications Magazine (2002)

Appendix: Simulation in Details

Packet dropping

Since standard deviation of samples S_x, S_y, and S_z are unknown before the experiment, we use a normal distribution as an approximation of $t-$distribution to simulate the attack. Thus the confidence interval CI_x can be expressed by $CI_x = (-\frac{\sigma}{\sqrt{n}}t_0, \frac{\sigma}{\sqrt{n}}t_0)$, where $\Phi(t_0) = 1 - \frac{\alpha}{2}$, and Φ is the distribution function of $N(0, \sigma)$. For example, if we take the level of significance $\alpha = 0.05, \sigma = 5$, and $n = 16$, we can calculate $CI_x = (-2.45, 2.45)$. To make packet dropping be detected, it is required that $\overline{x} < -2.45$, which means $\sum x_i < -2.45 \times 16 = -39.2$. This indicates at least 40 winning packets need to be dropped. The number of packets need to be dropped can be calculated by $N_{drop} \cdot s \leq -40$, where s is winning rate. If the nodes are running a VoIP application using G.711 Codec($Rate = 64kbit/s$) with a frame size (including the headers) of 200 bytes [4], then an originator can send out 40 packets per second. If at least 40 packets need to be sent to detect the dropping, then the minimum time to detect this attack is 1 second. The minimum time t can be calculated by:

$$t = \frac{minN_{sent}}{r} \tag{12}$$

where r is the packets rate of a node. Numerical results when $\alpha = 0.2$, 0.1, 0.05, 0.025, and 0.01 are listed in the following tables.

In our calculation, $r = 40$ packets/second, and $minN_{sent}$ ($minN_{drop}$) denotes the minimum number of packets need to be sent (dropped) to detect packet dropping.

The originator and recipient collusion

From inequality (10), we get $N_{Br} \leq \frac{w_r}{s \cdot \delta_0}$. So given a w_r and δ_0, the upper bound of N_{Br} can be calculated by this inequality. Suppose $\delta_0 = 1.3$, the relationship

Table 1. Numerical results when $\alpha = 0.2$

s	1	0.8	0.6	0.5	0.4	0.25	0.1	0.05	0.02	0.01
$minN_{drop}$	26	33	44	52	65	104	256	512	1280	2560
$minN_{sent}$	26	33	44	52	65	104	256	512	1280	2560
$t(sec)$	0.7	0.8	1.1	1.3	1.6	2.6	6.4	12.8	32.0	64.0

Table 2. Numerical results when $\alpha = 0.1$

s	1	0.8	0.6	0.5	0.4	0.25	0.1	0.05	0.02	0.01
$minN_{drop}$	33	42	55	66	83	132	330	660	1650	3300
$minN_{sent}$	33	42	55	66	83	132	330	660	1650	3300
$t(sec)$	0.8	1.1	1.4	1.7	2.1	3.3	8.3	16.5	41.3	82.6

Table 3. Numerical results when $\alpha = 0.05$

s	1	0.8	0.6	0.5	0.4	0.25	0.1	0.05	0.02	0.01
$minN_{drop}$	40	50	67	80	100	160	400	800	2000	4000
$minN_{sent}$	40	50	67	80	100	160	400	800	2000	4000
$t(sec)$	1.0	1.3	1.7	2.0	2.5	4.0	10.0	20.0	50.0	100.0

Table 4. Numerical results when $\alpha = 0.025$

s	1	0.8	0.6	0.5	0.4	0.25	0.1	0.05	0.02	0.01
$minN_{drop}$	45	57	75	90	113	180	448	896	2240	4480
$minN_{sent}$	45	57	75	90	113	180	448	896	2240	4480
$t(sec)$	1.1	1.4	1.9	2.3	2.8	4.5	11.2	22.4	56.0	112.0

Table 5. Numerical results when $\alpha = 0.01$

s	1	0.8	0.6	0.5	0.4	0.25	0.1	0.05	0.02	0.01
$minN_{drop}$	52	65	87	104	130	207	516	1032	2580	5160
$minN_{sent}$	52	65	87	104	130	207	516	1032	2580	5160
$t(sec)$	1.3	1.6	2.1	2.6	3.3	5.2	12.9	25.8	64.5	129.0

among s, w_r, and $maxN_{Br}$ are listed in the following table. Here is an example, when $w_r = 20$ and $s = 0.05$, we find $\frac{w_r}{s \cdot \delta_0} = 308$ at the intersection of the corresponding row and column. This means if a recipient gets 20 or more (much higher than 308×0.05) winning tickets within 308 packets, then the originator and recipient are suspected as colluding. The time needs to detect such attack can be calculated similarly as in equation (12).

Table 6. Numerical results when originator and recipient colluding

w_r \ s	1/1.3	0.7	0.5	0.3	0.2	0.1	0.05	0.01
20	20	22	31	52	77	154	308	1539
40	40	44	62	104	154	308	616	3077
60	60	66	93	154	231	462	924	4616
80	80	88	124	206	308	616	1231	6154
100	100	110	154	257	385	770	1539	7693