Trust-Based Secured Cooperation Incentive Scheme For Multi-Hop Wireless Network

Shravanthi T *Dept of CSE, BIT, VTU, India*, **Sowmya T** *Dept of CSE, BIT, VTU,* ,**Shresta N M** *Dept of CSE, BIT, VTU,India,*

***Abstract*- Existing credit card payment schemes are designed for different system and threat models, which are infeasible for MWNs, we propose a RACE, a report-based payment scheme for multihop wireless networks to stimulate node cooperation, regulate packet transmission, and enforce fairness. The nodes submit lightweight payment reports (instead of receipts) to the accounting center (AC) and temporarily store undeniable security tokens called *Evidence*s. The reports contain the alleged charges and rewards without security proofs, e.g., signatures. The AC can verify the payment by investigating the consistency of the reports, and clear the payment of the fair reports with almost no processing overhead or cryptographic operations. For cheating reports, the *Evidences* are requested to identify and evict the cheating nodes that submit incorrect reports. Instead of requesting the *Evidence*s from all the nodes participating in the cheating reports, RACE can identify the cheating nodes with requesting few *Evidence*s. Moreover, RACE can secure the payment and precisely identify the cheating nodes without false accusations.**

Keywords- Cooperation incentive schemes, payment schemes, accusation.

I. INTRODUCTION

In multihop wireless networks (MWNs), the traffic originated from a node is usually relayed through the other nodes to the destination for enabling new applications and enhancing the network performance and deployment [1]. MWNs can be deployed readily at low cost in developing and rural areas. Multihop packet relay can extend the network coverage using limited transmit power, improve area spectral efficiency, and enhance the network throughput and capacity. MWNs can also implement many useful applications such as data sharing [2] and multimedia data transmission [3]. For example, users in one area (residential neighborhood, university campus, etc) having different wireless-enabled devices, e.g., PDAs, laptops, tablets, cell phones, etc, can establish a network to communicate, distribute files, and share information.

In this paper, we propose **RACE,** a **R**eport-based p**A**yment s**C**hem**E** for MWNs. The nodes submit lightweight payment reports (instead of receipts) to the AC to update their credit accounts, and temporarily store undeniable security tokens called *Evidence*s. The reports contain the alleged charges and rewards of different sessions without security proofs, e.g., signatures.

The AC verifies the payment by investigating the consistency of the reports, and clears the payment of the fair reports with almost no cryptographic operations or computational overhead. For cheating reports, the *Evidences* are requested to identify and evict the cheating nodes that submit incorrect reports, e.g., to steal credits or pay less. In other words, the *Evidence*s are used to resolve disputes when the nodes disagree about the payment. Instead of requesting the *Evidence*s from all the nodes participating in the cheating reports, RACE can identify the cheating nodes with submitting and processing few *Evidence*s. Moreover, *Evidence* aggregation technique is used to reduce the storage area of the *Evidences*.

In RACE, *Evidence*s are submitted and the AC applies cryptographic operations to verify them only in case of cheating, but the nodes always submit security tokens, e.g., signatures, and the AC always applies cryptographic operations to verify the payment in the existing receipt-based schemes. RACE can clear the payment nearly without applying cryptographic operations and with submitting lightweight reports when *Evidences* are not frequently requested. Wide-spread cheating actions are not expected in civilian applications because the common users do not have the technical knowledge to tamper with their devices. Moreover, cheating nodes are evicted once they commit one cheating action and it is neither easy nor cheap to change identities.

To the best of our knowledge, RACE is the first payment scheme that can verify the payment by investigating the consistency of the nodes' reports without systematically submitting and processing security tokens and without false accusations. RACE is also the first scheme that uses the concept of *Evidence* to secure the payment and requires applying cryptographic operations in clearing the payment only in case of cheating.

II. RELATED WORKS

The existing payment schemes can be classified into tamperproof- device (TPD) based and receipt-based schemes. In TPDbased payment schemes [7-10], a TPD is installed in each node to store and manage its credit account and secure its operation. For receipt-based payment schemes [11-20], an offline central unit called the accounting center stores and manages the nodes' credit accounts. The nodes usually submit undeniable proofs for relaying packets, called receipts, to the AC to update their credit accounts.

Table 2 gives the description of the used symbols in this paper.

Table 2: Description of the used symbols.

Symbol	Description
X, Y	X is concatenated to Y.
F	A flag bit indicating whether the last received packet by a node is for acknowledgement (ACK) or data.
$h^{(i)}$	The hash value number i in a hash chain created by the destination node.
H(P)	The hash value resulted from hashing P.
$H_K(P)$	The keyed hash value resulted from hashing P using the key K.
ID_A	The identity of an intermediate node A.
ID_S and ID_D	The identities of the source node (S) and the destination node (D), respectively.
K_A	The shared key between node A and the TP.
M_X	The message sent in the Xth data packet.
n	The number of nodes in a route.
$P_C(n)$	The average payment clearance delay for a route with n nodes.
R	The concatenation of the identities of the nodes in a route, e.g., R = ID_S, ID_A, …, ID_D.
$Sig_S(P)$ and $Sig_D(P)$	The signatures of the source and the destination nodes on P, respectively.
T_{Cert}	The lifetime of a certificate.
Ts	The time stamp of a route establishment.

In this paper, we adopt the network model used in [7-17] that targets the civilian applications of MWNs, where the network has long life and the nodes have long-term relations with the network. As illustrated in Fig. 1, the considered MWN has an offline trusted party (TP) and mobile nodes. The TP contains the accounting center (AC) and the certificate authority (CA). The AC maintains the nodes' credit accounts and the CA renews and revokes the nodes' certificates. Each node A has to register with the trusted party to receive a symmetric key KA, private/public key pair, and certificate. The symmetric key is used to submit the payment reports and the private/public keys are required to act as source or destination node. We assume that the clocks of the nodes are synchronized. The details of this synchronization process are out of the scope of the paper, but several mechanisms have been proposed to synchronize the nodes' clocks [21]. Once the AC receives the payment reports of a session and verifies them, it clears the payment if the reports are fair; else, it requests the *Evidence*s to identify the cheating nodes. The CA evicts the cheating nodes by denying renewing their certificates.

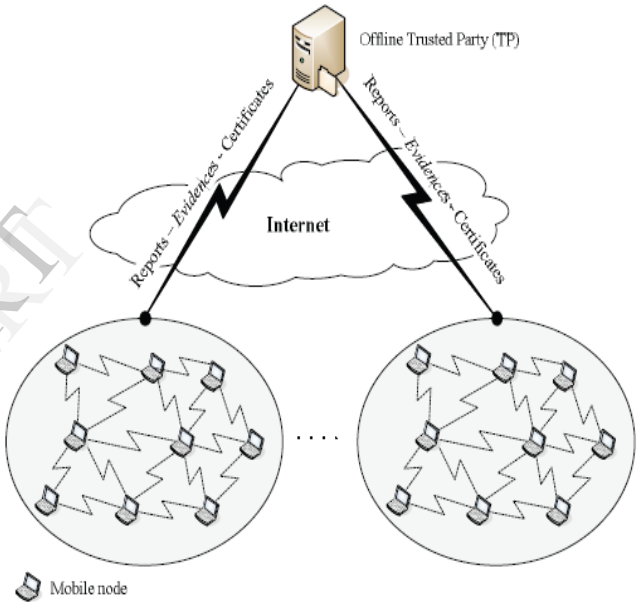


Fig. 1: The architecture of the considered network.

For the payment model, source nodes are charged for every transmitted message even if it does not reach the destination nodes, but the intermediate nodes are rewarded only for the delivered messages.

III. PROPOSED SCHEME

As shown in Fig. 2, RACE has four main phases. In *Communication* phase, the nodes are involved in communication sessions and *Evidence*s and payment reports are composed and temporarily stored. The nodes accumulate the payment reports and submit them in batch to the TP. For the *Classifier* phase, the TP classifies the reports into fair and cheating. For the *Identifying Cheaters* phase, the TP requests the *Evidence*s from the nodes that are involved in cheating reports to identify the cheating nodes. The cheating nodes are evicted and the payment
reports are corrected. Finally, in *Credit-Account Update* phase, the AC clears the payment reports.

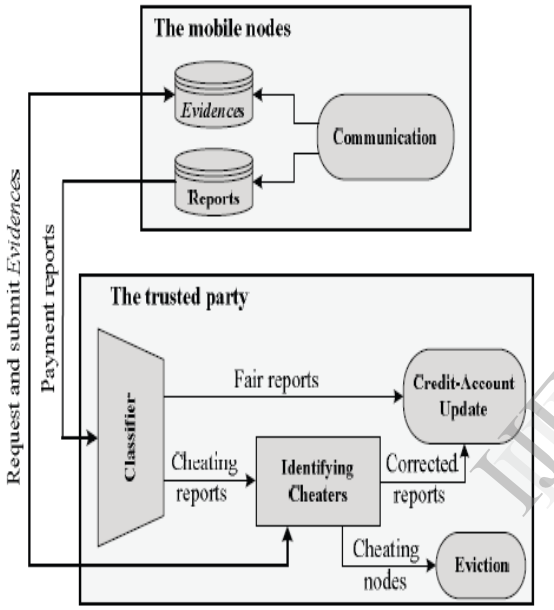


Fig. 2: The architecture of RACE.

Communication

The *Communication* phase has four processes: route establishment, data transmission, *Evidence* composition, and payment report composition/submission.

Route establishment: In order to establish an end-to-end
route, the source node broadcasts the *Route Request* (*RREQ*) packet containing the identities of the source (IDS) and the destination (IDD) nodes, time stamp (Ts), and Time-To-Live (TTL). TTL is the maximum number of intermediate nodes. After a node receives the *RREQ* packet, it appends its identity and broadcasts the packet if the number of intermediate nodes is fewer than TTL. The destination node composes the *Route Reply* (*RREP*) packet for the nodes broadcasted the first received *RREQ* packet, and sends the packet back to the source
node. The destination node creates a hash chain by iteratively hashing a random value (h(K)) K times to produce the hash chain root (h(0)), where h(i-1) = H(h(i)) and 1 □ i □ K. The optimal value of K depends on many factors such as the number of messages the source node needs to send, and the average number of messages sent through a route before it is broken, i.e., due to node mobility. Estimating a good value for K can save the destination node's resources because once a route is broken, the unused hash values in the hash chain should not be used for another route to secure the payment. The nodes can estimate the value of K and periodically tune it.

The *RREP* packet contains the identities of the nodes in the route (e.g., R = IDS, IDA, IDB, IDD in the route shown in Fig. 3), h(0), and the destination node's certificate and signature (SigD(R, Ts, h(0))). This signature authenticates the hash chain and links it to the route. The intermediate nodes verify the destination node's signature, relay the *RREP* packet, and store the signature and h(0) for composing the *Evidence*.

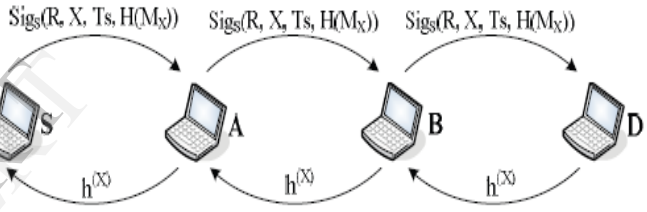


Fig. 3: The security tokens of the *X*th data and ACK packets.

Data transmission: The source node sends data packets to the destination node through the established route and the destination node replies with ACK packets. For the *X*th data packet, the source node appends the message MX and its signature to R, *X*, Ts, and the hash value of the message (H(MX)) and sends the packet to the first node in the route. The security tokens of the *X*th data and ACK packets are illustrated in Fig. 3. The source node's signature is an undeniable proof for transmitting *X* messages and ensures the message's authenticity and integrity. Signing the hash of the message instead of the message can reduce the *Evidence* size because the smaller-size H(MX) is attached to the *Evidence* instead of MX. Before relaying the packet, each intermediate node verifies the signature to ensure the message's authenticity and integrity, and verifies R and *X* to secure the payment. Each node stores only the last signature for composing the *Evidence*, which is enough to prove transmitting *X* messages, e.g., after receiving the *X*th data packet, the nodes should store SigS(R, *X*, Ts, H(MX)) and remove SigS(R, *X-1*, Ts, H(MX-1)), and so on. The data transmission process ends when the source node transmits its last message, or if the route is broken,

e.g., due to node mobility or channel impairment. Algorithm 1 gives the pseudo code of the processes of data transmission and composition of *Evidence* and report.

After receiving the *X*th data packet, Fig. 3 shows that the destination node sends back an ACK packet containing the pre-image of the last sent hash value (or h(X)) to acknowledge receiving the message MX, where h(1) is released in the first ACK and h(2) in the second and so on. Each intermediate node verifies the hash value by making sure that h(X-1) is obtained from hashing h(X). The nodes store only the last released hash value for composing the *Evidence*. The possession of h(X) by a node is a proof of delivering *X* messages, but the possession of SigS(R, *X*, Ts, H(MX)) is a proof of delivering *X*-1 messages and receiving one. The number of delivered messages can be computed from the number of hashing operations to map h(X) to h(0), and the number of transmitted messages (*X*) is signed by the source node. An intermediate node cannot drop the *X*th data packet and claim delivering it because the hash function is one way, i.e., it is computationally infeasible to compute h(X) from h(X-1). Hash chains have been used for many purposes due to their low energy and computational overhead, and nonrepudiation and one-way properties. In RACE, hash chains are used to reduce the number of public key cryptography operations, i.e., instead of generating a signature per ACK packet to secure the payment, one signature is generated by the destination node per K ACK packets.

***Evidence* composition:** *Evidence* is defined as information that is used to establish proof about the occurrence of an event or action, the time of occurrence, the parties involved in the event, and the outcome of the event. The purpose of an *Evidence* is to resolve a dispute about the amount of the payment resulted from data transmission. Fig. 4 gives the general format of an *Evidence*. The figure shows that an *Evidence* contains two main parts called DATA and PROOF. The DATA describes the payment, i.e., who pays whom and how much, and contains the necessary data to regenerate the nodes' signatures. From Fig. 4, the DATA contains the identities of the nodes in the route (R), the number of received messages (*X*), the session establishment time stamp (Ts), the root of the destination node's hash chain h(0), the hash value of the last message (H(MX)), and the last received hash value (h(V)). V = *X*-1 when the last received packet is the *X*th data packet because the route is broken before receiving the *X*th ACK packet that carries h(X), but V = *X* when the last received packet is the *X*th ACK packet. The DATA does not have h(1) when the route is broken after receiving the first data packet because the ACK that has h(1) is not received. The PROOF is an undeniable security token that can prove the correctness of the DATA and protect against payment manipulation, forgery, and repudiation. The PROOF is composed by hashing the destination node's signature and the last signature received from the source node, instead of attaching the signatures to reduce the *Evidence* size.

*Evidence*s have the following main features:

1. *Evidence*s are unmodifiable: If *X* messages are delivered, the intermediate nodes can compose *Evidence*s for fewer than *X* messages, but not for more. This is because the intermediate nodes have SigS(R, i, Ts, H(Mi)) and h(i) for i = {1, 2, .. , *X*}, which are sufficient for composing *Evidence*s for fewer than *X* messages. However, the intermediate nodes cannot compose *Evidence*s for more than *X* because it is computationally infeasible to compute SigS(R, i, Ts, H(Mi)) or h(i) for i > *X*.
2. If the source and destination nodes collude, they can create *Evidences* for any number of messages because they can compute the necessary security tokens.
3. *Evidence*s are unforgeable: If the source and destination nodes collude, they can create *Evidence* for sessions that did not happen, but the intermediate nodes cannot, because forging the source and destination nodes' signatures is infeasible.
4. *Evidence*s are undeniable: This is necessary to enable the TP to verify them to secure the payment. A source node cannot deny initiating a session or the amount of payment because it signs the number of transmitted messages and the signature is included in the *Evidence*.
5. An honest intermediate node can always compose valid *Evidence* even if the route is broken or the other nodes in the route collude to manipulate the payment. This is because it can verify the *Evidence*s to avoid being fooled by the attackers.

Reducing the storage area of the *Evidence*s is important because they should be stored until the AC clears the payment. Onion hashing technique can be used to aggregate *Evidence*s. The underlying idea is that instead of storing one PROOF per session, one compact PROOF can be computed to prove the credibility of the payment of a group of sessions. The compact *Evidence* contains the concatenation of the DATAs of the individual *Evidence*s and one compact PROOF that is computed by onion hashing the PROOFs of the individual *Evidence*s. Let

PROOF(i) refer to the PROOF of the *Evidence* number i, the compact PROOF is computed as follows:

H(….,H(H(PROOF(1), PROOF(2)), PROOF(3)),… ,PROOF(n))

PROOF(1) and PROOF(2) are concatenated and hashed, and then PROOF(3) is added to the compact PROOF by

adding one hashing layer and so on. The compact PROOF has the same size of the PROOF of individual *Evidence*, but it can prove the credibility of the payment of multiple sessions. The onion hashing technique enables the nodes to aggregate a recent *Evidence* with the old compact *Evidence*, i.e., *Evidence*s are always stored in an aggregated form to reduce their storage area. The technique is called onion hashing because each aggregation operation requires adding one hashing layer.

$$DATA = R, X, Ts, H(M_X), h^{(0)}, [h^{(V)}]$$

$$PROOF = H(Sig_S(R, X, Ts, H(M_X)), Sig_D(R, Ts, h^{(0)}))$$

Fig. 4: The general format of an *Evidence*.

Payment report composition/submission: A payment report contains the session identifier, a flag bit (F), and the number of messages (X). The session identifier is the concatenation of the identities of the nodes in the session and the time stamp. The flag bit is zero if the last received packet is data and one if it is ACK. Table 3 gives numerical examples for the payment reports of node A. For the first report, A is the source node and claims sending 12 messages, but it did not receive the ACK of the last message because F is zero. For the second report, A is the destination node and claims receiving 17 messages. For the third report, A is an intermediate node and claims receiving 15 messages, but it did not receive the ACK of the last message. The submission of reports and *Evidence*s are illustrated in Algorithm 2 and Fig. 5.

Table 3: Numerical examples for reports submitted by node A.

Session identifier	F	X
ID_A, ID_W, ID_C, ID_B, Ts_1	0	12
ID_C, ID_W, ID_Y, ID_Z, **ID_A**, Ts_2	1	17
ID_W, ID_Y, **ID_A**, ID_Z, Ts_3	0	15
……		

Algorithm 2: Submission/clearance of reports and *Evidence*s

1: $n_i \rightarrow$ TP: **Submit**(Reports[t_{i-1}, t_i));
2: TP $\rightarrow n_i$: ***Evidences*_Request**(Ses_IDs[t_{i-2}, t_{i-1}));
3: $n_i \rightarrow$ TP: **Submit**(Req_Evs[t_{i-2}, t_{i-1}));
4: TP: **Identify_Cheaters**();
5: TP: Clear the payment of the reports;
6: **if** (n_i is honest) **then**
7: TP $\rightarrow n_i$: A renewed certificate;
8: **end if**

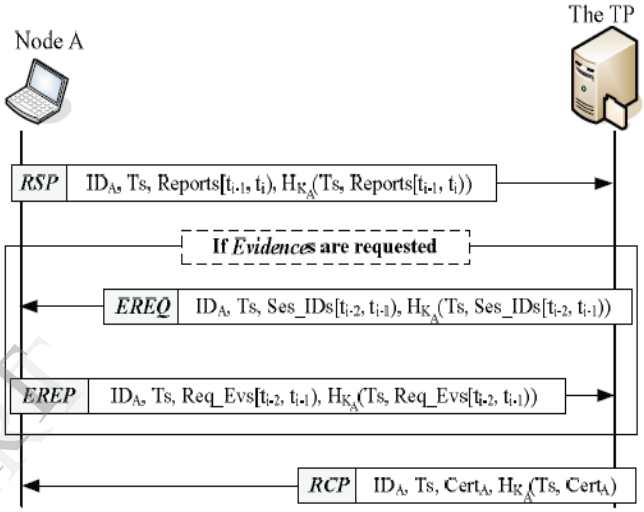
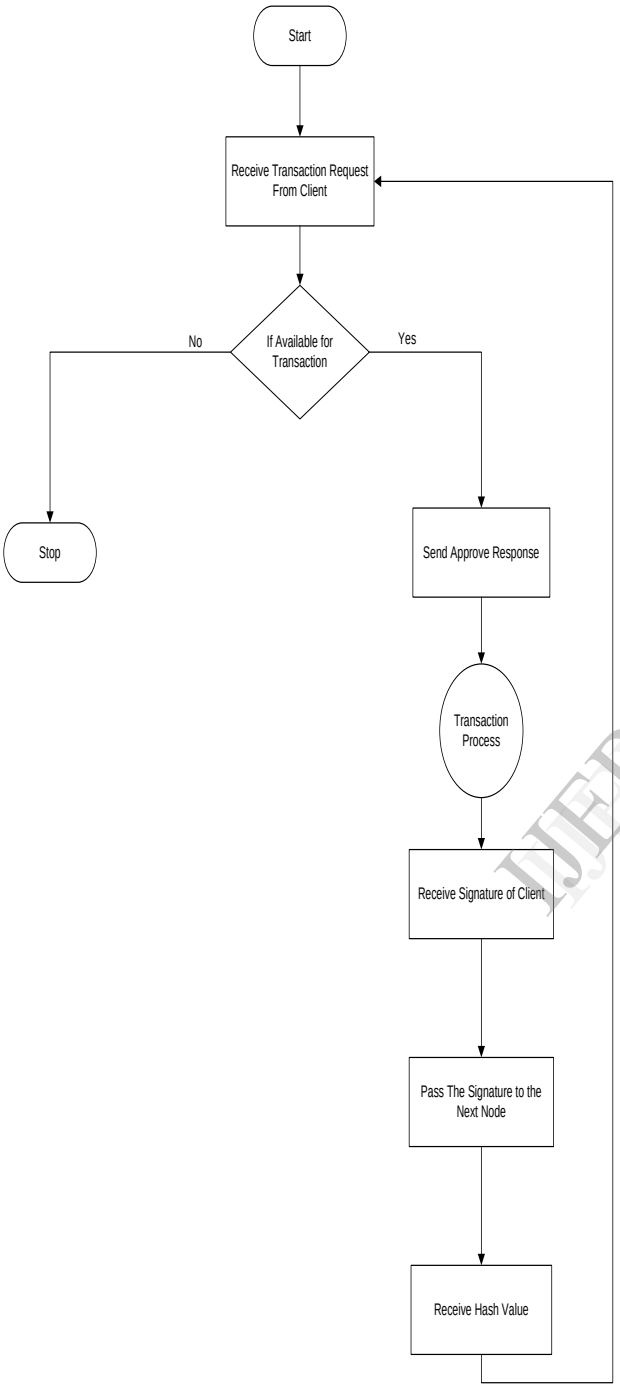


Fig. 5: The submission of reports and *Evidence*s.

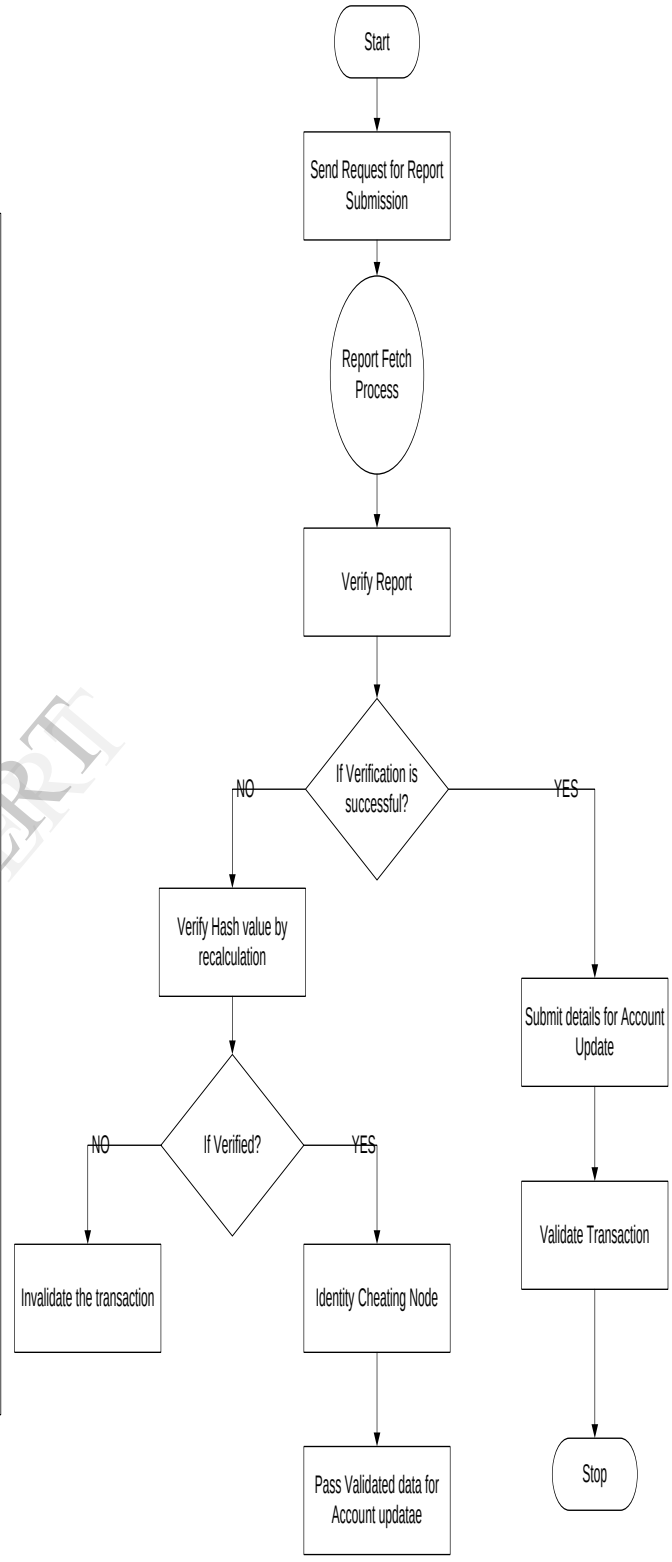
As shown in Fig. 5, node A sends a *Report Submission Packet* (*RSP*) to the TP at time ti to submit the reports of the sessions held since the last contact at ti-1. The packet contains the reports of the sessions held in [ti-1, ti) (Reports[ti-1, ti)), timestamp (Ts), and a keyed hash value (HKA()) to ensure the packet's integrity and authenticity, where KA is the long-term symmetric key shared between node A and the TP. Thus, the TP can make sure that the packet has not been manipulated and the reports are indeed sent by the intended node, which is important to secure the payment and hold the nodes accountable for any misbehavior. If the TP requests *Evidence*s from node A, it sends an *Evidences Request Packet* (*EREQ*) containing the identifiers of the reports that their *Evidences* are requested (Ses_IDs[ti-2, ti-1)). Node A replies with *Evidences Reply Packet* (*EREP*) containing the requested *Evidence*s (Req_Evs[ti-2, ti-1)). If node A is honest, the TP sends a *Renewed Certificate Packet* (*RCP*) containing a renewed certificate for node A with the same identity and public/private keys but with updated lifetime. Therefore, only the efficient hashing operations are used to submit the reports and *Evidence*s securely to the TP. Note that *RSP* and *RCP* are also required in receipt-based payment schemes to submit the receipts.

IV. FLOW CHART

Transaction Process Flow at Intermediate Nodes:



Payment Gateway Verification Flow Chart:



CONCLUSION

In this paper, we have designed a system that provides secure offline payment scheme that reduces the processing overhead on PG(Payment gateway).The nodes submit lightweight payment reports and temporarily store *Evidence*s. The fair reports can be cleared with almost no cryptographic operations. The PG runs heavy cryptographic algorithms only when the nodes submit incorrect hash values and those nodes are evicted for future transactions.

REFERENCES

[1] G. Shen, J. Liu, D. Wang, J. Wang, and S. Jin, "Multi-hop relay for next-generation wireless access networks", Bell Labs Technical Journal, vol. 13, no. 4, pp. 175-193, 2009.
[2] C. Chou, D. Wei, C. Kuo, and K. Naik, "An efficient anonymous communication protocol for peer-to-peer applications over mobile ad-hoc networks", IEEE Journal on selected areas in communications, vol. 25, no. 1, January 2007.
[3]H. Gharavi, "Multichannel mobile ad hoc links for multimedia communications", Proc. of the IEEE, vol. 96, no. 1, pp. 77-96, January 2008.
[4] S. Marti, T. Giuli, K. Lai, and M. Baker, "Mitigating routing misbehavior in mobile ad hoc networks", Proc. of ACM Mobile Computing and Networking (MobiCom'00), pp. 255–265, Boston, Massachusetts, USA, August 6-11, 2000.
[5] G. Marias, P. Georgiadis, D. Flitzanis, and K. Mandalas, "Cooperation enforcement schemes for MANETs: A survey", Wiley's Journal of Wireless Communications and Mobile Computing, vol. 6, issue 3, pp. 319–332, 2006.
[6] Y. Zhang and Y. Fang, "A secure authentication and billing architecture for wireless mesh networks", ACM Wireless Networks, vol. 13, no. 5, pp. 663–678, October 2007.
[7] L. Buttyan and J. Hubaux, "Stimulating cooperation in selforganizing mobile ad hoc networks", Mobile Networks and Applications, vol. 8, no. 5, pp. 579-592, October 2004.