2016 IEEE 36th International Conference on Distributed Computing Systems

Toward Optimal DoS-resistant authentication in Crowdsensing Networks via Evolutionary Game

Na Ruan[†‡], Lei Gao[†‡], Haojin Zhu[*†‡], Weijia Jia[†‡], Xiang Li[†], Qi Hu[†]

[†] Shanghai Jiao Tong University, China
[‡] {naruan,zhu-hj,jia-wj}@cs.sjtu.edu.cn,gaolei_work@foxmail.com

***Abstract*—With the increasing demand of Quality of Service (QoS) in Crowdsensing Networks, providing broadcast authentication and preventing Denial of Service (DoS) attacks become not only a fundamental issue but also a challenging security service. The multi-level μTESLA is a series of lightweight broadcast authentication protocols, which can effectively mitigate DoS attacks via randomly selected messages. However, the rule of the parameter selection still remains a problem. In this paper, we formulate the attack-defense model as an evolutionary game accordingly, and then present an optimal solution, which achieves security assurance along with minimum resource cost. We then analyze the stability of our evolutionary strategy theoretically. Simulation results are given to evaluate the performance of the proposed algorithm under low QoS channels and severe DoS attacks, which demonstrates that our proposed protocol can works even in the extreme case.**

***Index Terms*—Denial of Service, Broadcast Authentication Protocol, Lightweight Network, Timed Efficient Stream Loss-tolerant Authentication (TESLA), Evolutionary Game.**

I. Introduction

A. Background

With the rapid development of Internet, more and more resources have been integrated into the network for a higher efficiency and a wider dissemination. In 2013, Daren C. Brabham puts forth a problem-based typology of crowdsourcing [1], which is developed rapidly and has a huge advantage in reducing the costs of enterprises and converting intelligence into benefit. The implementation of crowdsensing network depends on the Mobile Crowdsourcing Networks (MCNs) [2], [3] for the contact among service requester, service providers and task participants. However, due to inherent characteristics of MCNs, such as the openness of task participation, limitation of device resources, privacy of data and dynamics of network topology, etc.Sealrity and privacy still represent a challenge.

Due to the openness of MCNs, namely the dynamics and diversity of service participants, MCNs without an effective authentication mechanism are very vulnerable to malicious attacks. For instance, by injecting invalid data, a malicious or even unintended attacker will not only abuse the network resources, but also bring about a task abortion. This vulnerability can be manipulated for a DoS attack, which will seriously affect QoS.

*Haojin Zhu is the corresponding author

Traditional asymmetric encryption algorithm requires service providers of MCNs to create a large amount of the ciphertext copies for different users, as well as saving public keys of all users. This requirement will take up a lot of resources, while the computing and storage resources of MCNs are very limited according to their structural features.

These problems on security and resource limitations will seriously affect the QoS of MCNs, especially in sensing tasks [4] and computing tasks [5] with specific time requirements, thus putting forward a demand for networks with lightweight user authentication. Multi-level TESLA, based on TESLA (Timed Efficient Stream Loss-tolerant Authentication), is the first to provide both lightweight broadcast authentication and DoS-resistant ability.

B. Motivation

Although multi-level μTESLA provides lightweight authentication, we find that its storage cost can be further lessened by means of hashing long message into a shorter Message Authentication Code (MAC). This motivated us to revise the multi-level μTESLA protocol to enhance its efficiency.

Game theory has been a successful tool to help analyze problems in network security [6]. However, most of the applications of game theory in network security are based on the hypothesis of perfect rationality, which is impractical since in reality we only have bounded rationality. Evolutionary game, on the contrary, holds the assumption of bounded rationality. This advantage makes it stand out among the other game models when applied to analyze attack-defense model in network security. This motivated us to apply evolutionary game theory to formulate and analyze the attack-defense model in DoS-tolerant multi-level μTESLA.

C. Challenge

The main challenge of our work lies in the following 4 parts.

i) To reduce storage cost in DoS-tolerant multi-level μTESLA, we need to design an algorithm to hash long messages to shorter MACs without losing the authentication ability of the protocol.
ii) While revising the multi-level μTESLA, we shall not harm its DoS-tolerant nature.

1063-6927/16 $31.00 © 2016 IEEE
DOI 10.1109/ICDCS.2016.66

iii) We need to formulate the pay-off of attackers and defenders thus we are able to know if there is any Evolutionary Stable Strategy (ESS) and why there is such ESS.
iv) Based on the game theoretical analysis, we need to design a strategy to help increasing the pay-off of defenders.

D. Contribution

The contributions of our work can be summarized as follows.

i) To further reduce the storage cost in DoS-tolerant multi-level μTESLA, we propose a DoS-resistant Authentication Protocol (DAP).
ii) To solve the buffer size selection problem in DoS-tolerant multi-level μTESLA, we formulate the attack-defense model as an evolutionary game.
iii) Base on the evolutionary game model, we derive the solution to the problem, present a theoretical analysis of the stability of the strategy, and then propose an implementation algorithm.
iv) We carry out various simulations to verify the superiority of our proposed algorithm, and compare it with some fixed pre-set algorithms.

The remainder of the paper is organized as follows. In section II, we introduce some previous works in this field and discuss how our work differs from other related works. In section III, we present our previous work about the attack-defense model in Mobile Crowdsensing Networks (MCNs) and novel promotion protocol for MCNs in section IV. Considering QoS, we propose an implementation algorithm based on the equilibrium strategy accordingly in section V. The evaluations of our two algorithms shown in section VI, respectively. The last section draws a conclusion of our work.

II. Preliminaries and Related Work

In this part, we discuss some related works on DoS-resistant schemes in lightweight networks, including multi-level μTESLA and TESLA++. Moreover, since we cover the Evolutionary Game Theory (EGT) later, we also introduce EGT and discuss some previous applications in wireless networks. Before discussing related work, we also introduce some preliminary knowledge for consistency.

A. MAC, TESLA, μTESLA

To elaborate preliminary techniques used to resist DoS attacks in lightweight networks, we first give some notations of variables.

- I_i: the ith time interval
- k_i, K_i: the shared secret key used in time interval I_i
- F: one-way hash function used to generate keys
- d: number of time intervals of key disclosure delay
- $P_{i,m}$: the mth packet received in time interval I_i
- $MAC_{K_i}(M)$: MAC computed by encrypting message M with key K_i

TESLA (Timed Efficient Stream Loss-tolerant Authentication) [7] and μTESLA [8], along with many variants, have been successively designed to achieve broadcast authentication in severely resource-constrained environments like MCNs. They are proposed to use symmetric cryptography to achieve asymmetric property, taking advantage of sender's delayed disclosure of keys.

The main idea of TESLA is that each packet is attached with a message authentication code (MAC), which is computed with a shared secret key k_i, over the contents of the packet, i.e. $MAC_{K_i}(Message)$. In addition, keys are derived from a one-way key chain, $k_i = F(k_{i+1})$, where F is a one-way function, which implies k_{i+1} cannot be derived from k_i even if F is known. Specifically, each key k_i is used in the time interval I_i, and would be disclosed after $(d-1)$ time intervals to make keys secret during this period of time.

For the receiver, each packet with attached MAC should be buffered as long as corresponding key is still secret. When the key is disclosed after $(d-1)$ time intervals, the receiver can use disclosed key to compute the theoretical MAC of the buffered packet and then compare it with the MAC attached to authenticate it. On the other hand, TESLA also provides the property of tolerating packet loss. Since each key is derived from a one-way key chain, the receiver can use k_{i+1}, which is disclosed in the time interval I_{i+1+d}, and the one-way function F to compute k_i if k_i is lost, by relation $k_i = F(k_{i+1})$.

μTESLA contributes in adapting to resource-constrained networks differing with TESLA mainly in two parts. First, μTESLA uses symmetric mechanisms instead of digital signature in authentication of the initial packet. Second, μTESLA discloses the key once per epoch, instead of disclosing a key in each packet as TESLA, to avoid communication overheads caused by excess key disclosures.

Figure 1 gives an example of the key generation and usage in μTESLA, which also roughly illustrates those mechanisms in TESLA. One-way function F_0 is used to generate K_i. In each time interval I_i, packets from $P_{i,1}$ to $P_{i,m}$ share the same key K_i to compute respective MACs.

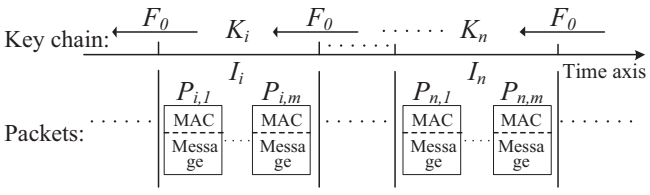


Fig. 1. Keys' generation and usage in μTESLA

Unlike normal networks, where asymmetric cryptography like digital signature is commonly used, a common broadcast authentication protocol called μTESLA, a variant of TESLA, is designed for lightweight networks [8]. TESLA-like protocols have been widely discussed, such as multi-level μTESLA in WSNs and TESLA++ in VANETs.

In multi-level μTESLA, a multiple-buffer random-selection method is used to mitigate DoS attacks, since it makes flooding DoS attacks relatively meaningless. However, since

this method consumes more bandwidth and storage resource, directly applying this into lightweight networks is not recommended.

TESLA++ resists DoS attacks by senders sending data packets and message authentication codes (MACs) separately as well as receivers rehashing MACs to significantly decrease the occupation of memory space [9]. However, since TESLA++ uses digital signatures after symmetric authentication, it may not be suitable for many highly resource-constrained lightweight networks.

B. Evolutionary Game Theory and its Application in Wireless Networks

The evolutionary game formalism is a central mathematical tool developed by biologists for predicting population dynamics in the context of interactions between populations. In classical evolutionary game theories, players are assumed to be rational, which means they pursue the maximum profit and make decisions accordingly in the game; however, this assumption may not hold in many real scenarios, such as the case that interaction dynamic and stability weights more than rationality in populations' evolution. The bounded rationality is an important characteristic of lightweight wireless networks, for attackers may not fully consider their own payoff and network nodes may not be capable of acquiring complete information of the whole network. That will be discussed further in Section V.

Assuming that all players in a population are identical in the sense they only choose strategies that succeeded before, the evolutionary game theory leads to two important concepts: replicator dynamics and evolutionary stable strategy (ESS). The replicator dynamics is a model for changes of the size of the populations in the game. The evolutionary stable strategy is the final stable strategies for players in the game. According to evolutionary game theory (EGT), we have

Definition 1. When there are two players in the game, if there exists a strategy σ, $\sigma' \neq \sigma$, $\varepsilon(\sigma') \in (0, 1)$, such that

$$u(\sigma, \varepsilon\sigma' + (1-\varepsilon)\sigma) > u(\sigma', \varepsilon\sigma' + (1-\varepsilon)\sigma),$$

then σ is the evolutionary stable strategy.

Furthermore, there are two main properties of evolutionary game theory.

- An ESS can be interpreted as a Nash equilibrium. If an ESS is reached, then the proportions of each population do not change in time
- The replicator dynamics may lead to stability results, ESS, and the populations are immune from being invaded by other small populations at ESS.

C. Applications of Evolutionary Game Theory in Wireless Networks

Game theories, including evolutionary game theory, have been widely applied in wireless networks to study the optimal strategies for many problems, such as intrusion detection, security, energy efficiency [10], [11]. In this section, we present several previous works related to ours and compare the differences between them and our work.

Liu X. et.al studied how evolutionary game can be applied in crowdsensing network [12]. This paper examines the evolutionary process among participants sensing networks and proposes an evolutionary game model to depict collaborative game phenomenon in the crowd sensing networks.It also established a incentive mechanism to correct the penalty function of the game model accordance with the cooperation rates of the participants.

Baik H. et.al studied participation and data quality issues in mobile crowdsourcing parking services [13]. This work proposed an incentive mechanism platform where high quality parking data can be obtained from unreliable crowds of mobile users using evolutionary game theory. Here the evolutionary game approaches helps to simulate the participation of mobile users for contributing information as well as determining the user utility.

Aside from those above, works such as [14] adapted the theory of evolutionary games with a random number of players or nodes in wireless networks. This paper derived unique ESS for different scenarios and applied the game framework in a context of W-CDMA network. Analogously, Altman et.al used EGT to design framework which supports evolution of congestion control protocols in wireless networks [15]. Nonetheless, these approaches are also not related to safety concern.

III. Our Previous Work

Before we propose our new protocol, we briefly review and introduce our previous work on resisting DoS attacks in WSNs. Based on multi-level μTESLA, we successively proposed two broadcast authentication protocols, Efficient Fault-Tolerant Protocol (EFTP) and Enhanced DoS-Resistant Protocol (EDRP), to enhance the resistance to DoS attacks by efficiently tolerating packet loss [16].

What multi-level μTESLA differs from other TESLA-variants is that it uses multiple key layers (high-level key with MAC together called CDM_i), besides providing a DoS-resistent strategy. Multiple key layers benefit WSNs good scalability in initializing thousands of sensor nodes with reasonable cost. Specifically, the high-level key chain can cover a long period of time without a too-long key chain due to long time intervals, while the low-level key chain provides short key chain intervals to avoid the high demand of computation and storage resource of long key chains.

However, just because of this complicated key generation and usage structure, several issues arise accordingly, including package loss of high-level CDM_i and weaker resistance to DoS attacks because of packet loss. Therefore, we propose EFTP and EDRP to solve this problem and enhance resistance to DoS attacks.

A. Efficient Fault-Tolerant Protocol (EFTP)

In multi-level μTESLA, though it connects low-level key chain and high-level key chain to tolerate low-level packet loss, high-level packet loss remains unsolved, which makes the resistance to DoS attackers weaker since MACs cannot be verified consecutively in a long period (possibly two high-level time intervals).

EFTP shortens the recovery time of lost packets in WSNs. Specifically, when high-level packets (MACs with other commitments in higher level) are lost, our protocol can shorten the recovery time by one high-level time interval. Since the time needed for buffering packets before authentication can be shortened, DoS attacks based on memory can also be mitigated.

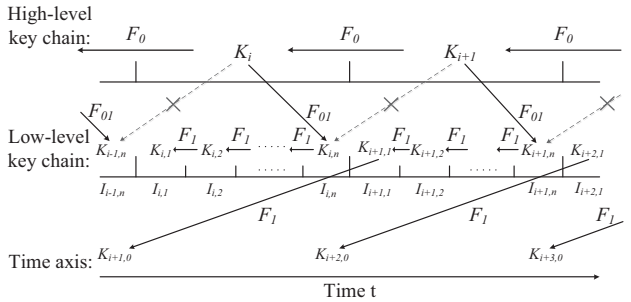


Fig. 2. Generation and usage of keys in EFTP

The key-layer construction is illustrated in Figure 2. Different from original scheme in multi-level μTESLA, the one-way function F_{01} is used to connect K_i and $K_{i,n}$, instead of K_{i+1} and $K_{i,n}$, by relation $K_{i,n} = F_{01}(K_i)$. The dash line represents the original connection, while the solid line represents the new connection.

Our theoretical analysis and evaluation result show that EFTP shortens the recovery time by one high-level time interval (varying from 100 seconds to 30 hours in real life), without losing security assurance.

B. Enhanced DoS-Resistant Protocol (EDRP)

EDRP contributes in tolerating packet loss. In other words, when one or more packets are lost, while DoS-resistant mechanism will no longer take effects in original scheme, it can still take effects in our protocol, which is especially meaningful in communication lossy channels.

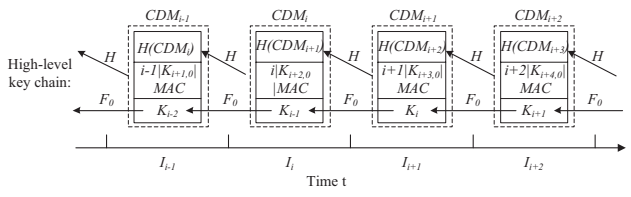


Fig. 3. Construction of high-level packets in EDRP

Figure 3 describes the new high-level key chain in the Enhanced DoS-Resistant Protocol. Here, $H(CDM_i)$ and K_i are pointed out to emphasize their utility and connection. Note that $H(CDM_i)$ is the image of CDM_i, with pseudorandom function H. Moreover, K_i is the high-level key chain, generated by one-way function F_0.

Consequently, when sensor nodes fail to receive CDM_i in time interval I_i, the sensor nodes can shorten the recovery time of CDM_i by taking advantage of high-level key chain, K_i. Specifically, combining K_i received in I_{i+1} and one-way function F_0 yields theoretical value of K_{i-2}, which is $F_0(F_0(K_i))$, while the authentic K_{i-2} has been stored in sensor nodes and could be used to make a comparison.

In summary, EDRP guarantees the continuous authentication of high-level packages, which furthermore guarantees the continuous resistance to DoS attacks, while maintaining same performance in security realm as the original scheme.

IV. DoS-Resistant Authentication Protocol (DAP)

As introduced in section III, using message authentication codes (MACs) and taking advantage of time intervals between packet broadcasting and key disclosure could provide authentication for lightweight networks. To furthermore resist DoS attacks in MCNs, we proposed a new authentication protocol, namely DoS-Resistant Authentication Protocol (DAP).

A. Protocol Sketch

Since using MACs to achieve authentication is symmetric and has a low computation cost, we focus on memory-based DoS attacks, and combine two strategies together to resist DoS attacks under different requirements.

First, considering possible forged packets from attackers, DAP sets multiple buffers for nodes, and randomly selects packages received to store in nodes' buffers. The idea is similar to multi-level μTESLA, since they both use multiple buffers to make such DoS attacks useless. The probability that a node stores at least one copy of the authentic packet from legitimate sender is $P = 1 - p^m$, where m is the number of buffers and p is the percentage of forged packets among all packets received. For different network environments, nodes can hold different buffers, namely different memory spaces, to achieve different security requirements.

Second, when nodes store packets in buffers and wait for one time interval, I_i, to authenticate received packets, the more buffers held by nodes, the higher possibility P for successful authentications DAP could guarantee. To minimize the packet size, in DAP, only MACs are broadcasted at first, and then $\mu MACs$, calculated with hash function, are stored in nodes. Besides, messages are broadcasted at the same time with key disclosure, like TESLA++, without losing proper authentication. Consequently, compared with storing MAC and message together, like TESLA, storing μMAC saves about 80% of memory space, the percentage varying depending on hash functions used when calculating MACs.

Figure 4 shows the broadcasting and authentication process of DAP between senders and receivers. In MCNs, the sender and receiver can be any mobile node.

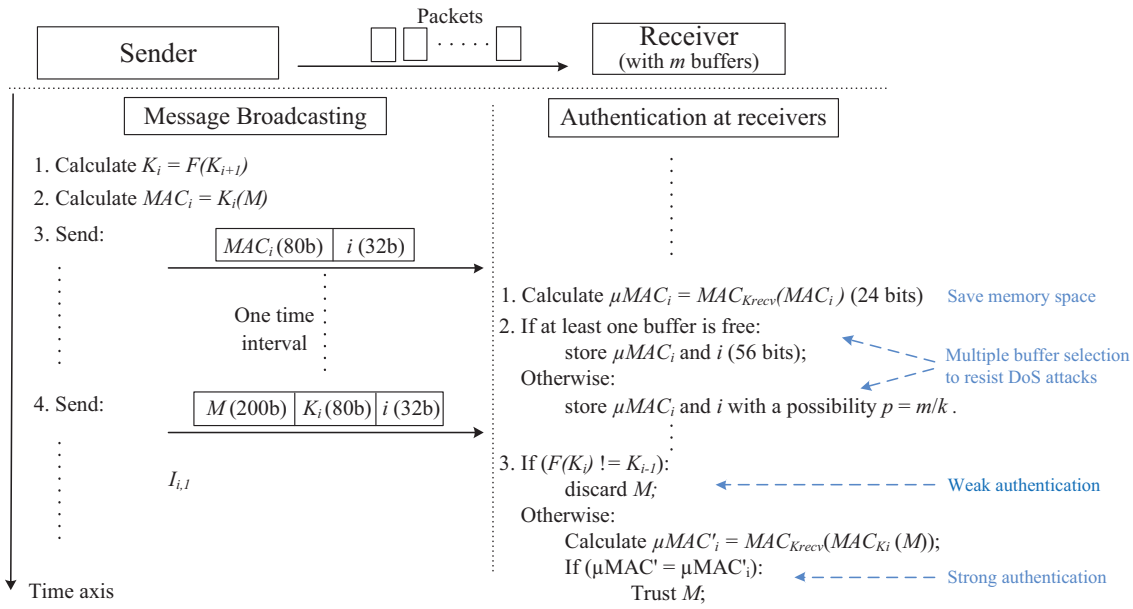


Fig. 4. A sketch of DAP

B. Protocol Details

DAP, as a variant of TESLA, keeps the same in the sender setup part, the bootstrapping receivers part, and key disclosure part [7]. However, DAP's message broadcasting and authentication at receivers needs to be specified. We illustrate the details step by step, and propose Algorithm 1 and Algorithm 2 accordingly.

1) Message Broadcasting: First, in time interval I_i, the sender broadcasts message M_i. It uses key K_i, which is calculated by the one-way hash function h, to calculate $MAC_i = MAC_{K_i}(M_i)$. Then, it sends MAC_i and index i to receivers.

Second, after one time interval, the sender discloses key K_i, and sends it to receivers together with message M_i and index i.

Algorithm 1 DAP: Message Broadcasting

Require: Message M_i, disclosed key K_i, index i; One-way hash function h;

Ensure: Broadcast message M_i and other to receivers;

1: Before broadcasting M_i in time interval I_i
2: Select K_i from the key chain generated by h;
3: $MAC_i = MAC_{K_i}(M_i)$;
4: Send MAC_i, i to receivers;
5: Wait;
6: In time interval I_{i+1}, send K_i, M_i, i to receivers

2) Authentication at Receivers: Upon receiving MAC_i and i in time interval I_x, the receiver first checks whether $i + d < x$, according to the loose time synchronization between senders and receivers. If so, the receiver discards this unsafe packet, since the key has already been disclosed. Otherwise, calculate $\mu MAC_i = MAC_{K_{Recv}}(MAC_i)$, where K_{Recv} is a local secret key held by the receiver in DoS attack. Meanwhile, it marks the package as the kth copy received in I_x, since lots of copies are sent to flood the network. Line 2 to line 5 in Algorithm 2 illustrates this procedure accordingly.

Afterwards, the receiver checks whether all buffers are occupied. If not, the receiver picks one empty buffer and puts the package into it. If so, it stores this package with a possibility of $\frac{m}{k}$, where m is the amount of receiver's buffers. If a copy is to be kept, the receiver randomly selects one of m buffers and replaces the corresponding copy. This is to ensure that all copies are kept by receiver with a probability $\frac{m}{n}$, where n is the amount of all copies received. Line 6 to line 13 illustrates this process.

Upon receiving message M_i and disclosed key K_i in time interval I_{x+1}, the receiver first uses the weak authentication, *i.e.* discarding M_i if $h(K_i) \neq K_{i-1}$, since trustable M_i should contains a K_i in the key chain. To furthermore perform the strong authentication, the receiver calculates $\mu MAC' = MAC_{K_{Recv}}(MAC_{K_i}(M_i))$ with messages newly received, and then compares it with μMAC stored. Indeed, the equivalence implies the authenticity of M_i. In Algorithm 2, the procedure is shown from line 17 to line 26.

C. Security Analysis of DAP

Since the way how message and MACs are broadcasted in DAP is different from previous protocol, it is of necessity to show that DAP is still secure in the sense attackers cannot forge messages which may be authenticated as trustable at network nodes. Indeed, while DAP changes the broadcasting method to reduce requirement of memory storage, it remains

Algorithm 2 DAP: Authentication at Receivers

Require: kth Packet received, P_k; Message M_i, disclosed key K_i, index i; One-way hash function h; Local secret key, K_{Recv}; Key disclosure delay, d; Current interval index, x; Amount of buffers, m;

Ensure: Authentication of M_i;

```
MAC_i and i received in in time interval I_x;
if i + d < x then
    discard MAC_i and i;
else
    μMAC = MAC_{K_Recv}(MAC_i);
    if k < m then
        store P_k in one empty buffer;
    else
        keep P_k with probability p = m/k;
        if P_k is kept then
            store the packet in a randomly selected buffer;
        end if
    end if
end if
Wait till M_i, K_i, i received in time interval I_{x+1};
if h(K_i) ≠ K_{i-1} then
    discard message M_i;
else
    μMAC' = MAC_{K_Recv}(MAC_{K_i}(M_i));
    if μMAC' ≠ μMAC then
        discard M_i;
    else
        message M_i is authenticated;
    end if
end if
```

same security level since there's no essentially changes in the authentication process.

As shown in Algorithm 2, the message is authenticated and held in nodes only when μMAC_i is stored since last time interval I_x, is equal to the $\mu MAC'_i$, calculated by newly received message M in I_{x+1}. If an attacker would like to forge message M by M_{forged}, it has to send MAC_{forged} to network nodes during I_x so that M_{forged} could be authenticated as trustable later. However, unless the attacker eavesdrops or hacked the key which disclosed in I_{x+1}, there is no way to calculate MAC_{forged} since the key is not yet disclosed. Consequently, DAP is still secure, provided with the loose time synchronization.

D. Performance Analysis of DAP

As discussed in section 4.1, DAP uses multiple-buffer selection of received packets strategy to resist DoS attacks with a success possibility $P = 1-p^m$. Clearly P increases with the growth of buffer number m, provided p is fixed. Indeed, this implies that the more buffers we have, the higher successful authentication possibility P we can get.

Since DAP stores only μMAC before authentication, if a packet with 200-bit message and corresponding 80-bit MAC arrives, DAP will store just 56 bits (24 for μMAC and 30 for index i) instead of 280 bits. Because 80% memory spaces are saved in DAP, the number of buffers in a node could be 5 times as before. As evaluated in section 7 later, this improvement greatly reduce the requirement of bandwidth percentage of MACs, if maintaining same P. In other words, given identical bandwidth percentage of MACs, DAP provides a higher possibility P.

V. QoS-balanced DoS-resistant Authentication Protocol

A. Motivation for QoS-security balance and Game Selection

As mentioned above, DAP resists DoS attacks partially due to multiple-buffer storage. Indeed, the multiple-buffer selection method achieves enhanced security with lower Quality of Service (QoS) support, since higher security demands more buffers in nodes and repeated message broadcasted. Moreover, as pointed out by Liu and Ning, the different choices of these parameters need to be examined for the balance of the trade-off above. Consequently, it is of necessity to determine the optimal defense strategy by selecting corresponding parameters based on real scenarios.

In order to determine the optimal strategy, we formulate this attack-defense model as an evolutionary game, and then determine the evolutionary stable strategy (ESS). Indeed, we choose the evolutionary game model instead of the classical game model for several reasons:

i) Sensor nodes are not capable of acquiring complete information about the whole network and thus cannot be regarded as a complete rational player, because : i) MCNs usually consist of thousands of sensor nodes and the network topology is changing frequently; ii) either frequently communicate with the base station or store all possible strategies of all nodes in advance are impractical. Indeed, this meets the assumption in evolution game theory that players have bounded rationality;

ii) Sensor nodes' strategy is formulated during the evolution by observing other nodes' behavior following the replicator dynamics. Afterwards, sensor nodes can determine their optimal strategies. Furthermore, once the optimal strategy is determined, it remains stable overtime, unlike the case that the classical non-cooperative game which yields to multiple Nash equilibriums.

Our final goal is to use the analysis of evolutionary stable strategy to figure out a method to save resource as well as resisting DoS attacks. Specifically, we study how parameters should be specified so that attackers give up DoS attacks, thereby separated memory space for buffers are not needed.

B. Game Formulation

The evolutionary game for the DoS-resistance problem in lightweight networks can be described as follows.

TABLE I
NOTATIONS IN EVOLUTIONARY GAME

m	number of buffers defenders use to store packets
x_a	fraction of bandwidth used by attackers
p	fraction of forged data
P	the success possibility of an attack
L_d	the damage of defenders under attack
R_a	the reward of a successful attack
C_a	cost of attackers
C_d	cost of defenders

- $Players$: As shown in the DoS attack-defense model, attackers (uncertified malicious parties) attack defenders (normal network nodes) by performing DoS attacks. Thus, two groups of players make up the game in terms of game theory, denoted as {Defender, Attacker}.
- $Population$: Since nodes in lightweight networks try to save resource like memory space or power consumption, etc, we assume there are two strategies when applying DAP: apply buffer-selection method (to resist DoS attacks) versus no buffers (secure when no DoS attacks). The population formed by defenders is $S_D = \{$Buffer selection, no buffers$\}$. Similarly, the population formed by attackers is S_A = {DoS attacks , no attack}, since attackers need to decide whether such DoS attacks are worthwhile with certain extent of cost.
- $Pay - off$: The pay-off of each player is determined by value of packets broadcasted and corresponding security levels, and this will be discussed later.

C. Specification of Player's Pay-off

To specify the pay-off matrix of all players, we first need to specify following parameters as shown in TABLE I.

Based on these parameters, we can furthermore determine more parameters needed.

1) Specification of P: First, we know

$$p = x_a$$

By Liu et.al [17] we have:

$$P = p^m = x_a^m$$

2) Specification of R_a, L_d, C_a, C_d: Here, we assume the reward is mainly decided by the correctness of data. Since it is difficult to measure these parameters without specific instances, we use reference values to reflect relative relationships among these parameters. We use following coefficients: k_1, k_2 to derive corresponding expressions. It is reasonable to suggest that C_a is positively correlated to x_a and Y. C_d is positively correlated to m and X. Thus we have:

$$\begin{aligned} R_a &= L_d \\ C_a &= k_1 x_a Y \\ C_d &= k_2 m X \end{aligned}$$

3) Specification of Pay-off Matrix: The initial unspecified pay-off matrix is shown in Table II. It is clear when attackers do not attack, they have no reward, and defenders' rewards depend on their costs in defense, C_d. On the other hand, when attackers attack, whether they succeed depends on whether defenders set buffers or not. When there is no defense, the reward is $R_a - C_a$ for attackers, and $-L_d$ for defenders. In case there is defense, attackers succeed with probability P. Note that $L_d = R_a$ since they are both directly decided by the value of data in our assumption.

TABLE II
PAY-OFF MATRIX BETWEEN ATTACKERS AND DEFENDERS

Defender \ Attacker	DoS attacks	No DoS attacks
Buffer selection	$-C_d - PL_d$, $PR_a - C_a$	$-C_d$, 0
No buffers	$-L_d$, $R_a - C_a$	0, 0

Therefore, we derive Table II as the pay-off matrix of players. Then, we specified Table II by combining equations derived from last two subsections.

D. Replicator Dynamic Expression

To specify the replicator dynamic in the evolutionary game, we first need to specify following notations as shown in TABLE III.

Then we have

$$\begin{aligned} E(U_d) &= Y(-C_d - PL_d) + (1-Y)(-C_d) \\ E(U_a) &= X(PR_a - C_a) + (1-X)(R_a - C_a) \\ E(U_{nd}) &= Y(-L_d) + (1-Y)\cdot 0 \\ E(U_{na}) &= 0 \cdot X + 0 \cdot (1-X) \\ E(d) &= XE(U_d) + (1-X)E(U_{nd}) \\ E(a) &= YE(U_a) + (1-Y)E(U_{na}) \end{aligned}$$

Then, we get the replicator dynamic expressions for defenders and attackers as follows, which describe how X and Y change with time t.

$$\begin{aligned} \frac{dX}{dt} &= X[E(U_d) - E(d)] \\ &= X(1-X)[L_d Y(1-P) - C_d] \\ &= X(1-X)[R_a Y(1-p^m) - k_2 m X] \\ \frac{dY}{dt} &= Y[E(U_a) - E(a)] \\ &= Y(1-Y)[R_a(1+PX-X) - C_a] \\ &= Y(1-Y)[(p^m - 1)XR_a + R_a - k_1 x_a Y] \end{aligned}$$

E. Evolution Stable Strategy Analysis

Here, we Let

$$\begin{aligned} \frac{dX}{dt} &= 0 \\ \frac{dY}{dt} &= 0 \end{aligned}$$

TABLE III
NOTATIONS IN REPLICATOR DYNAMIC

X	proportion of defenders which use buffer-selection strategy
Y	proportion of attackers which launches DoS attacks
$E(U_d)$	expectation of defender's payoff when playing "defense"
$E(U_{nd})$	expectation of defender's payoff when playing "no defense"
$E(U_a)$	expectation of attack's payoff when playing "attack"
$E(U_{na})$	expectation of attacker's payoff when playing "no attack"
$E(a)$	expectation of attacker's payoff
$E(d)$	expectation of defender's payoff

We have

$$X = 0, 1 \ or \ \frac{(1-p^m)R_a}{k_2 m} Y \tag{1}$$

$$Y = 0, 1 \ or \ \frac{R_a - (1-p^m)XR_a}{k_1 x_a} \tag{2}$$

Let (X, Y) denote the solution. (X, Y) is the possible ESS of the evolutionary game. Here we discuss on 9 possible ESS.

1) Consider $X = 0$. Since $R_a > C_a$, we know

$$R_a - k_1 x_a Y > 0$$

. Then $Y = 0$ or 1. Notice that when $0 < X < 1, 0 < Y < 1$,

$$\frac{dX}{dt} = 0, \ \frac{dY}{dt} > 0$$

Therefore $(0, 0)$ can not be ESS. And $(0, 1)$ can be ESS.

2) Consider $Y = 0$, we have $X = 0$ or 1. Notice that when $0 < X < 1$,

$$\frac{dX}{dt} = k_2 m x^2 (x - 1) < 0$$

Thus $(1, 0)$ cannot be ESS.

3) Consider $X = 1$, we have

$$Y = 0, \ 1, \ \frac{p^m R_a}{k_1 x_a} (named \ Y')$$

When $Y = 1$,

$$\frac{dX}{dt} = \frac{dY}{dt} = 0$$

Thus $(1, 1)$ can be ESS.

When $Y = Y' < 1$, notice that when $0 < Y < Y'$, $\frac{dY}{dt} > 0$, when $Y' < Y < 1$, $\frac{dY}{dt} < 0$. Thus $(1, Y')$ can be ESS.

4) Consider $Y = 1$, we have

$$X = 0, \ 1, \ \frac{(1-p^m)R_a}{k_2 m} (named \ X')$$

When $X = X' < 1$, since $\frac{dX}{dt} > 0$ when $0 < X < X'$ and $\frac{dX}{dt} < 0$ when $X' < X < 1$, we know $(X', 1)$ can be ESS.

5) Consider $X \neq 0, 1$ and $Y \neq 0, 1$ By (1),(2),we have

$$X' = \frac{(1-p^m)R_a^2}{k_1 k_2 m x_a + (1-p^m)^2 R_a^2}$$

$$Y' = \frac{k_2 m R_a}{k_1 k_2 m x_a + (1-p^m)^2 R_a^2}$$

Notice that $\frac{dX}{dt} > 0$ when $X < X'$, $\frac{dX}{dt} < 0$ when $X > X'$. Same as $\frac{dY}{dt}$. Thus (X, Y) can be ESS if $0 < X < 1$ and $0 < Y < 1$. In sum, the ESS can be $(0, 1), (\frac{(1-p^m)R_a}{k_2 m}, 1), (1, \frac{p^m R_a}{k_1 x_a}), (1, 1)$ and $(\frac{(1-p^m)R_a^2}{k_1 k_2 m x_a + (1-p^m)^2 R_a^2}, \frac{k_2 m R_a}{k_1 k_2 m x_a + (1-p^m)^2 R_a^2})$

For convenience, we name these ESS as $(0, 1), (X', 1), (1, Y')(1, 1), (X, Y)$.

F. Optimization of m

For a given p, we propose a method to find the optimal m. Let E denote the average cost of the MCN nodes.

$$E = -E(d) = k_2 m X^2 + [1 - (1-p^m)X] R_a Y$$

where (X, Y) is the ESS. Next we present the optimization of m.

Algorithm 3 Optimizing the number of buffers

Require: the attacker bandwidth fraction, p; coefficient, k_1; coefficient, k_2; reward for attacker, R_a;
Ensure: Optimal m_{optm};

```
m_optm = 0;
E_0 = ∞;
for m = 1 to 100 do;
    calculate the possible ESS (X,Y);
    E_m = k_2 m X^2 + [1 − (1 − p^m)X] R_a Y;
    if E_m < E_{m−1} then
        m_optm = m;
    end if
end for
return m_optm;
```

The above algorithm takes the attacker's information as input, outputs the optimal number of buffers for defender nodes. The correctness of this method relies on the reasonable quantifying of each player's utility.

VI. EVALUATIONS ON DAP PROTOCOL

A. Evaluation on Memory Cost

1) Evaluation Settings: Without loss of generality, let $x_d = 0.2$; Storage $Mem = 1024kb, 512kb$; Storage needed per packet $s_1 = 280kb$ in TESLA++; Storage needed per packet $s_2 = 56kb$ in DAP; Maximum buffer number $M_1 = Mem/s_1$, $M_2 = Mem/s_2$, while:

$x_m = p(1 - x_d) = \sqrt[m]{P}(1 - x_d)$, where $m = M_1$ or M_2.

2) Evaluation Results: Figure 5 shows the fraction of bandwidth required for MACs for different level of Dos attack. Here p stands for the fraction of Forged data. We can see that the bandwidth required for MACs in order to ensure the same P is substantially less when we use DAP instead of TESLA++.

B. Evaluation on Evolutionary Games

1) Evaluation Settings: Here, we set

$$R_a = 200$$

$$k_1 = 20$$

$$k_2 = 4$$

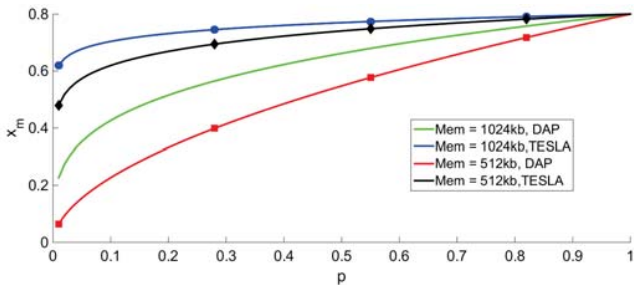


Fig. 5. Required bandwidth fraction at different level of Dos attack

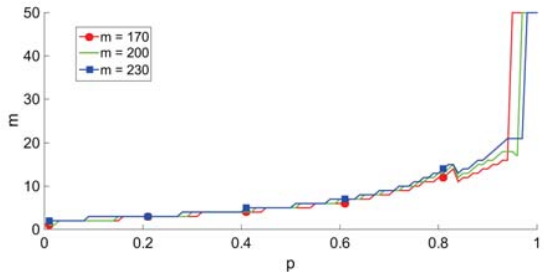


Fig. 7. Optimized number of buffers m at different level of Dos attack

The reasons for those settings are as follows. Generally speaking, $R_a > k_1 \geq C_a$ This means that the reward of attacker is greater than the cost of launching an attack.

By Liu et.al [17], we know that in sensor network, there are at most about 50 buffers for each node. So $m \leq M = 50$ We set $R_a \leq k_2 M$, to assume that putting all resources to defense will cost slightly more price than the value of the packet. This setting will discourages nodes from the strategy of naive defense.

2) Evolution Process: Fig. 6 shows the evolution process of ESS. Here we fix $p = 0.8$. And let $(X, Y) = (0.5, 0.5)$ as the origin setting of nodes and attacker.

We pick different m to see how the evolution process is conducted.

Note that the evolution is updated in this way:

$$X = X + \frac{dX}{dt} \cdot t$$
$$Y = Y + \frac{dY}{dt} \cdot t$$

where $t = 0.01$, the setting of t is to give a justify adjustment toward the previous R_a, k_1, k_2 settings to insure that while updating X and Y we can keep $0 < X \leq 1$ and $0 < Y \leq 1$.

We can find that with different m, we achieve different ESS.

When $1 \leq m \leq 11$, we reach the ESS of $(1, 1)$. The evolution process converges quickly in at most 4 steps.

When $12 \leq m \leq 17$, we reach the ESS of $(1, Y')$. X first quickly converges to 1, then Y converges to 0.44 slowly. It finally converges in about 100 steps.

When $18 \leq m \leq 54$, we reach the ESS of (X, Y). It converges spirally and finally converges in about 200 steps.

When $55 \leq m \leq 100$, we reach the ESS of $(X', 1)$. The evolution process converges quickly in at most 4 steps.

3) Buffer Size Optimization: Fig. 6 shows that the optimal choice of m in terms of reaching ESS and maximizing synthetical benefit of defender (payoff minus cost). p denotes the resource(bandwidth) spent by attacker.

We can find that to reach ESS, the optimal choice of m varied for different level of attacks.

When $p < 0.94$, we can achieve the ESS of (X, Y). When p is relatively low and DoS attacks are not strong, m is then chosen relatively small to resist this weak DoS attack. In addition, m increases when p increases.

When $p > 0.94$, the forged message almost jammed the channel. Here we see that m is set to 50. In fact, m is set to the largest buffer size. At this time, the ESS is $(X', 1)$ It means increasing m to resist the heavy Dos attack is not efficient, rather, it turns to give up.

4) Efficient Enhancement: Here we compare the Evolutionary Game based defense method with the naive defense method.

Let E denote the defense cost of defender when using evolutionary game based defense method, that is, requiring X of all nodes to play strategy defense with parameter m optimized. Let N denote the defense cost when using naive defense method, that is, requiring every node to play strategy defense with fixed parameter $m = M$.

We have:

$$E = k_2 m X^2 + [1 - (1 - p^m)X] R_a Y$$
$$N = k_2 M + p^M R_a Y'$$

where (X, Y) is the ESS with parameter m, $(1, Y')$ is the ESS with parameter M.

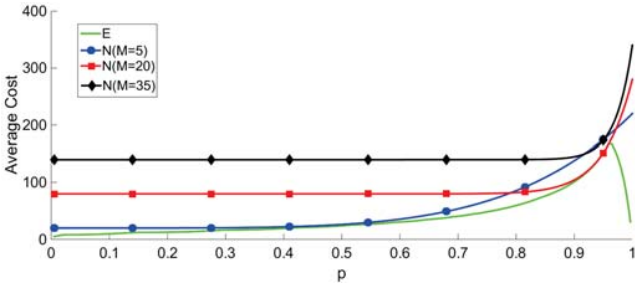
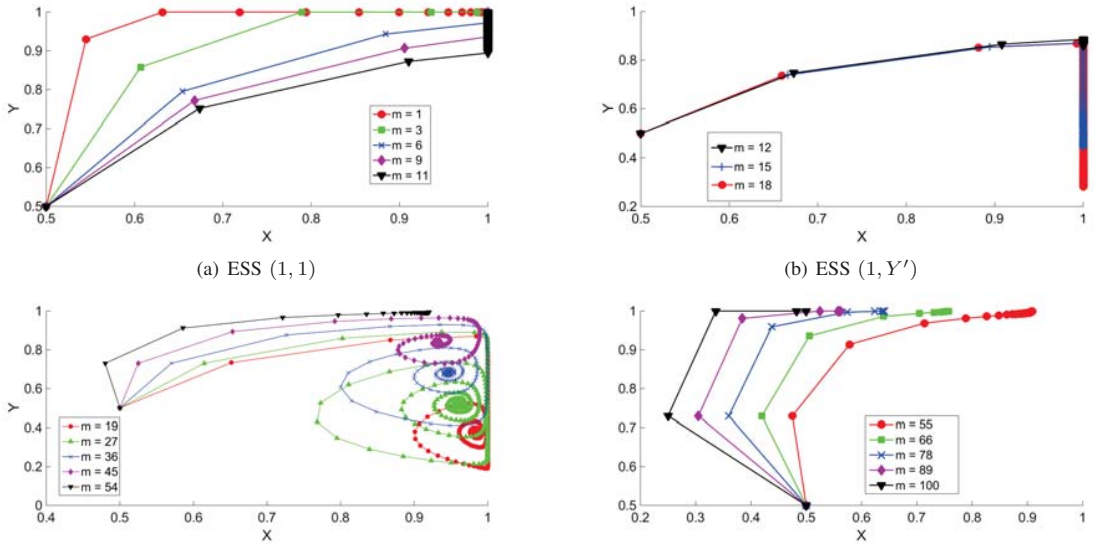


Fig. 8. Average defense cost at different level of Dos attack

Fig. 8 shows that the defense mechanism under the guidance of evolutionary game has a better performance than the naive defense mechanism. Especially when $p > 0.94$, our defense mechanism greatly reduces the average overall cost. As shown in Figure 7, it is setting $m = 50$ to move the ESS from (X, Y) to $(X', 1)$ that greatly reduces the overall cost.

This shows the success of our mechanism.



(a) ESS $(1, 1)$

(b) ESS $(1, Y')$

(c) ESS (X, Y)

(d) ESS $(X', 1)$

Fig. 6. The evolution process of evolutionary game

VII. Conclusion

In this paper, we proposed a DoS-resistant authentication protocol based on our previous work on multi-level μTESLA authentication protocol. We implement μMAC mechanism to save memory space in the multi buffer selection process, which is simulated in later sections to be effective in resource constrained MCNs. Further, we apply the evolutionary game theory to present an optimal solution for the parameter setting problem in our proposed protocol, which is shown to be effective in greatly alleviating the defense cost and predicting user behaviors.

Acknowledgement

This work is supported by Chinese National Research Fund (NSFC) Project (No. 61272444, U1401253, U1405251, 61411146001).National China 973 Project No. 2015CB352401; NSFC Key Project No. 61532013; Shanghai Scientific Innovation Act of STCSM No.15JC1402400; 985 Project of Shanghai Jiao Tong University with No. WF220103001, and Shanghai Jiao Tong University 211 Fund.

References

[1] Doan A, Ramakrishnan R, Halevy A Y. “Crowdsourcing systems on the world-wide web”. Communications of the ACM, 2011, 54(4): 86-96.

[2] Yang K, Zhang K, Ren J, Shen X. “Security and privacy in mobile crowdsourcing networks: challenges and opportunities”. Communications Magazine, IEEE, 2015, 53(8): 75-81.

[3] Agarwal Y, Hall M. “ProtectMyPrivacy: detecting and mitigating privacy leaks on iOS devices using crowdsourcing”. in MobiSys’13, Taipei, Taiwan, Jun. 2013.

[4] Lin J, Amini S, Hong J I, Sadeh N, Lindqvist J, Zhang J. “Expectation and purpose: understanding users’ mental models of mobile app privacy through crowdsourcing”. in Ubicomp’12, Pittsburgh, PA, USA. Sep. 2012.

[5] Faggiani A, Gregori E, Lenzini L, Luconi V, Vecchio A. “Network sensing through smartphone-based crowdsourcing”.in SenSys’13, Rome, Italy, Nov.2013.

[6] Manshaei M H, Zhu Q, Alpcan T, Hubuax J P. “Game theory meets network security and privacy”. ACM Computing Surveys, 2013, 45(3): 25.

[7] Perrig A, Canetti R, Tygar J D, Song D, Ran C. “Efficient authentication and signing of multicast streams over lossy channels”. in IEEE S&P, Berkeley, California, USA. May.2000.

[8] Perrig A, Szewczyk R, Tygar J D, Wen V, Culler DE. “SPINS: Security protocols for sensor networks”. Wireless networks, 2002, 8(5): 521-534.

[9] Studer A, Bai F, Bellur B, Perrig A. “Flexible, extensible, and efficient VANET authentication”. Communications and Networks, 2009, 11(6): 574-588.

[10] Han Z. “Game theory in wireless and communication networks: theory, models, and applications”. Cambridge University Press, 2012.

[11] Niyato D, Wang P, Kim D I, Han Z. “Game theoretic modeling of jamming attack in wireless powered communication networks”. in ICC’15, London, UK, Jun. 2015.

[12] Liu X, Ota K, Liu A, Chen Z. “An incentive game based evolutionary model for crowd sensing networks”. Peer-to-Peer Networking and Applications, 2015: 1-20.

[13] Hoh B, Yan T, Ganesan D, Tracton K, . “Trucentive: A game-theoretic incentive platform for trustworthy mobile crowdsourcing parking services”.in ITSC’12, Hilton Anchorage, AK, USA, Sep. 2012.

[14] Tembine H, Altman E, El-Azouzi R, Hayel Y. “Evolutionary games in wireless networks”. IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics, 2010, 40(3): 634-646.

[15] Altman E, Elazouzi R, Hayel Y, Tembine H. “An evolutionary game approach for the design of congestion control protocols in wireless networks”. in WiOpt’08, Berlin, Germany, Apr. 2008.

[16] Li X, Ruan N, Wu F, Jie L. “Efficient and enhanced broadcast authentication protocols based on multilevel TESLA”. in IPCCC’14, Austin, TX, USA, Dec. 2014.

[17] Liu D, Ning P. “Multilevel TESLA: Broadcast authentication for distributed sensor networks”. ACM Transactions on Embedded Computing Systems (TECS), 2004, 3(4): 800-836.

[18] Shim K A, Lee Y R, Park C M. “EIBAS: An efficient identity-based broadcast authentication scheme in wireless sensor networks”. Ad Hoc Networks, 2013, 11(1): 182-189.

[19] Gao Y, Zeng P, Choo K K R. “Multi-sender Broadcast Authentication in Wireless Sensor Networks”. in CIS’14, Kunming, Yunnan, China, Nov. 2014.

[20] Qian J, Qiu F, Wu F, Na R, Chen G, Tang S. “A Differentially Private Selective Aggregation Scheme for Online User Behavior Analysis”. in Globecom’15, San Diego, CA, USA, Dec. 2015.