Comments on "On the Design of Conditional Privacy Preserving Batch Verification-Based Authentication Scheme for Internet of Vehicles Deployment"

Yuhao Yang, and Xiujie Huang, *Member, IEEE*

***Abstract*—To maintain the secure information sharing among vehicles in the Internet of Vehicles, various message authentication schemes were proposed. Recently, Sutrala *et al.* proposed a conditional privacy preserving authentication scheme ("On the Design of Conditional Privacy Preserving Batch Verification-Based Authentication Scheme for Internet of Vehicles Deployment," *IEEE* Trans. Veh. Technol., vol. 69, no. 5, pp. 5535-5548, May 2020.) to against various potential attacks. However, our observations show that, contrary to what is claimed, the scheme is insecure. Any (malicious) vehicle can forge signature for any message, which can be validated successfully and cannot be traceable. Our observations also show that, the security proof based on the standard random oracle model is wrong.**

***Index Terms*—Authentication, forgery, internet of vehicles, privacy preserving, random oracle model.**

I. Introduction

INTERNET of vehicles (IoV) is evolving as a global heterogeneous vehicular networks, which improves vehicles communication as well as intelligence. Due to the openness of networks, information sharing among vehicles is prone to suffer potential attacks, such as, man-in-the-middle attack, impersonation attack, modification attack, replay attack and so on. From the standpoint of safety, the traffic information including weather, collision, jam, accident and emergency notifications should be available at vehicles in time. Hence, a secure and efficient message authentication scheme is required for the IoV system. Recently, Sutrala *et al.* proposed a conditional privacy preserving batch verification-based authentication scheme for IoV [1]. However, this scheme is not secure. In this paper, we give a successful signature forgery on any message, which can be validated and is untraceable. We also point out that the security analysis using the standard random oracle model is wrong.

The rest of this paper is organized as follows. Section II describes the scheme proposed by Sutrala *et al.*. Section III provides a forgery of the message-signature tuple for Sutrala *et al.*'s scheme to show its insecurity. In Section IV, we point out errors in the security analysis for Sutrala *et al.*'s scheme. Finally, in Section V, we conclude this paper.

Yuhao Yang and Xiujie Huang are with the College of Information Science and Technology, Jinan University, Guangzhou 510632, China, also with the Guangdong Key Laboratory of Data Security and Privacy Preserving, Guangzhou 510632, China. (e-mail: yyh_yangyuhao@163.com; t_xiujie@jnu.edu.cn). Xiujie Huang is the corresponding author.

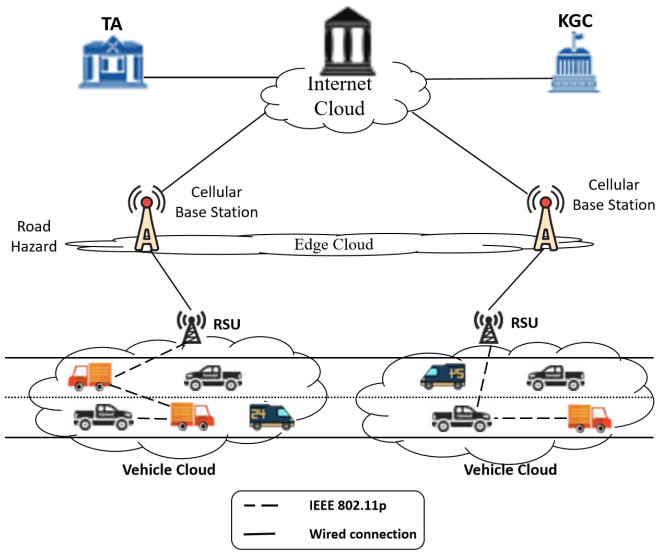

Fig. 1. The IoV system model [1].

II. Sutrala et al.'s Scheme

The IoV system model used in [1] is shown in Fig. 1. The model mainly consists of four entities: trusted authority (TA), key generation center (KGC), road side units (RSUs) and vehicles. The TA is a trusted entity with huge storage space and powerful computing power, and is responsible for the initialization of system parameters. The KGC is another trusted entity in the system. It has abundant computing and communication resources and is responsible for the generation of partial private keys for vehicles. RSUs are communication entities distributed along both sides of the road, acting as intermediate nodes for communication between the vehicle and TA or KGC. At the same time, the RSU is also responsible for the generation of pseudo identities for vehicles. A vehicle is equipped with an On-Board Units (OBU) which stores the sensitive information into the tamper proof device (TPD) and broadcasts messages to other vehicles and the nearby RSUs by the IEEE 802.11p wireless protocol.

The scheme proposed by Sutrala *et al.* for the above IoV system is restated as follows.

A. System Initialization

This phase generates the initial system parameters.

1) TA chooses two large prime numbers p and q. TA selects elliptic curve additive group G of order q, which is defined by $\mathbb{E}: y^2 = x^3 + ax + b (\text{mod } p)$, where $a, b \in Z_p$ and $4a^3 + 27b^2 (\text{mod } p) \neq 0$. P is a generator of the group G. TA randomly selects $r_1 \in Z_q^*$ as the master key and computes the public key $T_{pub} = r_1 P$.
2) KGC randomly selects $r_2 \in Z_q^*$ as its master key and computes the public key $K_{pub} = r_2 P$.
3) For each RSU, KGC randomly selects $r_3 \in Z_q^*$ as the RSU's private key, and computes the public key $R_{pub} = r_3 P$.
4) TA selects secure hash functions $h_1, h_2 : \{0,1\}^* \rightarrow Z_q^*$, $H_1, H_2, H_3, H_4, H_5 : G \rightarrow Z_q^*$ and $H_6 : \{0,1\}^* \times G \times \{0,1\}^* \times G \times \{0,1\}^* \times \{0,1\}^* \rightarrow Z_q^*$.
5) TA and KGC announce system parameters: $\{T_{pub}, K_{pub}, \{R_{pub} \text{ for each RSU}\}, h_1(\cdot), h_2(\cdot), H_1(\cdot), H_2(\cdot), H_3(\cdot), H_4(\cdot), H_5(\cdot), H_6(\cdot), G, P\}$.

B. Vehicle Registration

The vehicle sends a registration request to the TA with its real identity VID_i. TA then computes $a_i = h_1(VID_i)$, $vt_i = VID_i \oplus H_1(r_1 K_{pub})$, and then loads $< vt_i, a_i >$ to the vehicle V_i's TPD. TA stores a_i and marks V_i as a registered vehicle.

C. Vehicle Partial Key Generation

V_i sends a_i to KGC. KGC checks whether a_i is in the revocation list obtained from TA through a secure channel. If not, KGC randomly selects $r_i \in Z_q^*$, computes $R_i = r_i P$, $vpk_{1i} = r_2 a_i (\text{mod } q)$ and $vpk_{2i} = H_2(a_i K_{pub}) \oplus (r_i + r_2)$, and then sends $< vpk_{1i}, vpk_{2i}, R_i >$ back to the vehicle V_i.

D. Vehicle Key Generation

After receiving the partial key $< vpk_{1i}, vpk_{2i}, R_i >$ from KGC, V_i randomly chooses $k_i \in Z_q^*$, computes $K_i = k_i P$ and $vpk_{3i} = H_3(K_{pub} \oplus K_i)(vpk_{2i} \oplus H_2(a_i K_{pub}))(\text{mod } q) = H_3(K_{pub} \oplus K_i)(r_i + r_2)(\text{mod } q)$. V_i stores the private key $< vpk_{1i}, vpk_{3i}, k_i >$ and publishes the public key $< R_i, K_i >$.

E. Pseudo-Identity Generation

When V_i enters the coverage of a certain RSU, it requests the RSU to generate a pseudo identity for it.

1) V_i randomly chooses $r_{vi1}, r_{vi2} \in Z_q^*$, computes $R_{vi1} = r_{vi1} P$, $R_{vi2} = r_{vi2} P$, $vt_i' = vt_i \oplus H_4(r_{vi1} R_{pub})$ and $a_i' = a_i \oplus H_5(r_{vi2} R_{pub})$, and then sends the information $< vt_i', R_{vi1}, R_{vi2}, a_i' >$ to RSU.
2) After receiving the information, the RSU calculates $vt_i = vt_i' \oplus H_4(r_3 R_{vi1})$ and $a_i = a_i' \oplus H_5(r_3 R_{vi2})$. If a_i is not in the revocation list obtained from TA. RSU randomly selects $x_i \in Z_q^*$, computes $PID_i = \{PID_{1i}, pid_{2i}, T_i\}$, where T_i is the effective time of PID_i, $PID_{1i} = x_i P$, $pid_{2i} = vt_i \oplus h_2(a_i || x_i || T_i)$. RSU sends back the pseudo identity PID_i to V_i.

F. Message Signature Generation

Before broadcasting the message M_i, it must be signed by V_i as follows. V_i randomly selects $b_i \in Z_q^*$, computes $B_i = b_i P$, $g_i = (k_i^{-1} vpk_{1i}) (\text{mod } q)$ and $f_i = (b_i^{-1}(vpk_{1i} + vpk_{3i} H_6(M_i, R_i, T_1, PID_i)))(\text{mod } q)$, where T_1 is the current timestamp. Then V_i broadcasts the message-signature tuple $< M_i, \delta_i = \{f_i, g_i\}, B_i, K_i, R_i, PID_i, T_1 >$ within the communication range of the RSU.

G. Message Signature Verification

Once other vehicle V_j and RSU receive the information $< M_i, \delta_i = \{f_i, g_i\}, B_i, K_i, R_i, PID_i, T_1 >$, they verify whether the timestamps T_1 and T_i are valid. If both are valid, they calculate $L_i = H_3(K_{pub} \oplus K_i) H_6(M_i, R_i, T_1, PID_i)(R_i + K_{pub})$, and verify the following equation

$$f_i B_i - g_i K_i = L_i. \tag{1}$$

If Eqn. (1) holds, receive this message M_i. Otherwise, discard it.

H. Batch Verification

The batch verification of messages is done by the RSU. When the RSU receives n signed messages $< M_i, \delta_i = \{f_i, g_i\}, B_i, K_i, R_i, PID_i = \{PID_{1i}, pid_{2i}, T_i\}, T_{1i} >_{i=1,\ldots,n}$, it verifies whether T_{1i} and T_i are valid for each message. If both are valid, the RSU then computes $Z_i = H_6(M_i, R_i, T_{1i}, PID_i)$, $f_i B_i = a_i K_{pub} + H_3(K_{pub} \oplus K_i) Z_i (R_i + K_{pub})$, $g_i K_i = a_i K_{pub}$ and $L_i = H_3(K_{pub} \oplus K_i) Z_i (R_i + K_{pub})$. Let $\{\delta_i\}_{i=1,\ldots,n'}$ be the list of signatures which are freshly generated having valid pseudo identities. Then the RSU randomly chooses $\lambda_i \in \{0,1\}^l$ for $i = 1, \ldots, n'$, where usually $l = 80$ [2], and verify whether the equation

$$\sum_{i=1}^{n'} \lambda_i (f_i B_i - g_i K_i) = \sum_{i=1}^{n'} \lambda_i L_i \tag{2}$$

is true. If it is true, receive the n' messages $\{M_i\}_{i=1,\ldots,n'}$.

III. A Forgery for Sutrala et al.'s Scheme

The attacker's goal is to generate a message-signature tuple $< M_i, \delta_i = \{f_i, g_i\}, B_i, K_i, R_i, PID_i, T_1 >$ such that it can pass the verification as shown in Eqn. (1). It is assumed that the attacker is a vehicle which can be registered or not registered. The attacker performs the following steps to forge a signature on a message M_i, where M_i can be chosen arbitrarily.

1) The attacker randomly chooses $k_i \in Z_q^*$, and computes $K_i = k_i K_{pub}$.
2) The attacker arbitrarily constructs a triple as the pseudo identity $PID_i = \{PID_{i1} \in G, pid_{i2} \in Z_q^*, T_i\}$.
3) The attacker randomly chooses $r_i \in Z_q^*$ and computes $R_i = r_i K_{pub}$.
4) The attacker randomly chooses $b_i, v_i \in Z_q^*$, and computes $B_i = b_i K_{pub}$, $f_i = b_i^{-1}[v_i + (r_i + 1) H_3(K_{pub} \oplus K_i) \cdot H_6(M_i, R_i, T_1, PID_i)] (\text{mod } q)$ and $g_i = k_i^{-1} v_i (\text{mod } q)$.
5) The attacker broadcasts the message-signature tuple $< M_i, \delta_i = \{f_i, g_i\}, B_i, K_i, R_i, PID_i, T_1 >$.

It can be seen that the above message-signature tuple $< M_i, \delta_i = \{f_i, g_i\}, B_i, K_i, R_i, PID_i, T_1 >$ satisfies Eqn. (1), which is shown in detail as follows. Compute $g_i K_i = v_i K_{pub}$,

$$f_i B_i = [v_i + (r_i + 1)H_3(K_{pub} \oplus K_i)H_6(M_i, R_i, T_1, PID_i)]K_{pub},$$
$$L_i = H_3(K_{pub} \oplus K_i)H_6(M_i, R_i, T_1, PID_i)(R_i + K_{pub}).$$

Then

$$\begin{aligned} f_i B_i - g_i K_i &= (r_i + 1)H_3(K_{pub} \oplus K_i)H_6(M_i, R_i, T_1, PID_i)K_{pub} \\ &= H_3(K_{pub} \oplus K_i)H_6(M_i, R_i, T_1, PID_i)(R_i + K_{pub}) \\ &= L_i. \end{aligned}$$

Hence, the message-signature tuple $<M_i, \delta_i = \{f_i, g_i\}, B_i, K_i, R_i, PID_i, T_1 >$ is a successful forgery, which implies that Sutrala *et al.*'s scheme is insecure.

Moreover, from the generation of the message-signature forgery tuple $<M_i, \delta_i = \{f_i, g_i\}, B_i, K_i, R_i, PID_i, T_1>$, we can see that the pseudo identity PID_i=$\{PID_{i1} \in G, pid_{i2} \in Z_q^*, T_i\}$ is chosen arbitrarily, which may not be related to any vehicle or could be a registered legitimate vehicle (since its pseudo identity is always broadcasted along with its signature and can be obtained by the attacker). Hence, provided that the tracking procedure is reasonable, TA either gets nothing about the real identity of the attacker or regards a registered legitimate vehicle as the attacker after tracking.

IV. An Error in the Security Analysis Based on the Stardard Random Oracle Model

It is a common method to use the standard random oracle model to prove the security of a signature scheme. The core idea of the security proof is that, if there exists an adversary who can break the scheme with probability $\varepsilon > 0$ in polynomially-time, then a polynomially-time algorithm can be designed to solve the considered hard problem with non-negligible probability. Unfortunately, we find out that, the security proof for Sutrala *et al.*'s scheme shown by Lemma 1 proposed in [1] is wrong. Here, we first restate Lemma 1 and list the key points of its proof given in [1]. And then we will point out the error in the proof.

A restatement of Lemma 1 given in [1]: Let Λ be a polynomially bounded adversary who acts as a malicious third-party attacker and has the ability to request and replace the public key in the system. Suppose the adversary Λ can break Sutrala *et al.*'s scheme with probability ε>0. Then the challenger Γ can produce an algorithm to solve the ECDLP problem with non-negligible probability.

Key points in the proof of Lemma 1 shown in [1]: Given two elliptic curve points $P, Q = sP \in G$, the goal of the challenger is to compute the discrete logarithm (DL) $s \in Z_q^*$. This is an ECDLP problem. To solve the ECDLP problem, Γ interacts with Λ as the following phases.

Setup Phase: The challenger Γ initializes the system's public parameters $\{T_{pub}, K_{pub} = sP, \{R_{pub}$ for each RSU$\}$, $h_1(\cdot), h_2(\cdot), H_1(\cdot), H_2(\cdot), H_3(\cdot), H_4(\cdot), H_5(\cdot), H_6(\cdot), G, P\}$ and sends it to the adversary Λ. The DL s is kept secret.

Then the adversary Λ interacts with the challenger Γ by performing the following queries.

Query_on_ h_1: When Λ makes h_1 query on ID_i, Γ checks the $list_{h_1}$ for the tuple (ID_i, a_i). If such tuple exists, Γ returns a_i to Λ. Otherwise, Γ randomly chooses $a_i \in Z_q^*$, returns a_i to Λ, and adds the tuple (ID_i, a_i) to $list_{h_1}$.

Query_on_H_3: When Λ makes H_3 query on K_i, Γ checks the $list_{H_3}$ for the tuple (K_i, y_i). If such tuple exists, Γ returns y_i to Λ. Otherwise, Γ randomly chooses $y_i \in Z_q^*$, returns y_i to Λ, and adds the tuple (K_i, y_i) to $list_{H_3}$.

Query_on_H_6: When Λ makes H_6 query on (M_i, R_i, T_1, ID_i), Γ checks the $list_{H_6}$ for the tuple $(M_i, R_i, T_1, ID_i, z_i)$. If such tuple exists, Γ returns z_i to Λ. Otherwise, Γ randomly chooses $z_i \in Z_q^*$ returns z_i to Λ, and adds the tuple $(M_i, R_i, T_1, ID_i, z_i)$ to $list_{H_3}$.

Query_on_Create_Vehicle: When Λ queries on ID_i, Γ checks the $list_{cv}$ for the tuple (ID_i, k_i, K_i). If such tuple exists, Γ returns K_i to Λ. Otherwise, Γ randomly chooses $k_i \in Z_q^*$, computes $K_i = k_i P$, returns K_i to Λ, and adds the tuple (ID_i, k_i, K_i) to $list_{cv}$.

Query_on_Extract_Part_Priv_Key: When Λ queries on (ID_i, K_i), Γ checks the $list_{ppk}$ for the tuple $(ID_i, K_i, a_i, vpk_{1i}, vpk_{3i}, R_i)$. If such tuple exists, Γ returns $(vpk_{1i}, vpk_{3i}, R_i)$ to Λ. Otherwise, Γ queries h_1 on ID_i, H_3 on K_i to obtain a_i and y_i, respectively. Γ randomly chooses $r_i \in Z_q^*$, computes $vpk_{1i} = a_i$, $vpk_{3i} = r_i y_i$ and $R_i = r_i P$, returns $(vpk_{1i}, vpk_{3i}, R_i)$ to Λ, and adds the tuple $(ID_i, K_i, a_i, vpk_{1i}, vpk_{3i}, R_i)$ to $list_{ppk}$.

Query_on_Extract_Secret_Key: When Λ queries on ID_i, Γ checks the $list_{cv}$ for the tuple (ID_i, k_i, K_i). If such tuple exists, Γ returns k_i to Λ. Otherwise, Γ queries the $Create\_Vehicle$ to generate (ID_i, k_i, K_i) and returns k_i to Λ.

Query_on_Sign: When Λ makes $Sign$ queries on (ID_i, M_i, T_1), Γ queries $Create\_Vehicle$, $Extract\_Part\_Priv\_Key$, h_1, H_3 and H_6 to retrieve the tuples (ID_i, k_i, K_i), $(ID_i, K_i, a_i, vpk_{1i}, vpk_{3i}, R_i)$, (ID_i, a_i), (K_i, y_i) and $(M_i, R_i, T_1, ID_i, z_i)$, respectively. Γ randomly chooses $f_i \in Z_q^*$, computes $b_i = f_i^{-1}(vpk_{1i} + vpk_{3i} z_i) (\mathrm{mod}\ q)$, $g_i = k_i^{-1} vpk_{1i} (\mathrm{mod}\ q)$ and

$$B_i = b_i P + f_i^{-1} y_i z_i K_{pub}. \tag{3}$$

Finally, Γ returns the signature $\delta_i = (f_i, g_i)$ along with (B_i, K_i, R_i) to Λ.

From above, we can see that, by querying the $Sign$ oracle, the adversary Λ obtains a valid signature $\{\delta_i = (f_i, g_i), B_i, K_i, R_i\}$ on message (ID_i, M_i, T_1), as

$$\begin{aligned} f_i B_i - g_i K_i &= r_i y_i z_i P + y_i z_i K_{pub} \\ &= y_i z_i (R_i + K_{pub}) \\ &= H_3(K_{pub} \oplus K_i)H_6(M_i, R_i, T_1, ID_i)(R_i + K_{pub}) \\ &= L_i. \end{aligned} \tag{4}$$

The equality in Eqn. (4) shows $f_i B_i - g_i K_i = y_i z_i (R_i + K_{pub})$, which is equivalent to

$$f_i \beta_i - g_i k_i = y_i z_i (r_i + s) \tag{5}$$

since there exists $\beta_i \in Z_q^*$ such that $B_i = \beta_i P$ as defined in Eqn. (3), $K_i = k_i P$, $R_i = r_i P$ and $K_{pub} = sP$. There are four unknows β_i, k_i, r_i, s in Eqn. (5).

Forgery: By the forking lemma [3], when the challenger Γ makes query on the message (ID_i, M_i, T_1) using the same procedure as mentioned above, it can obtain another valid signature $\{\delta_i^{(1)} = (f_i, g_i^{(1)}), B_i^{(1)}, K_i, R_i\}$ with different choice of oracles h_1, H_3 and H_6. That is,

$$f_i B_i^{(1)} - g_i^{(1)} K_i = y_i^{(1)} z_i^{(1)} (R_i + K_{pub})$$

which is equivalent to

$$f_i \beta_i^{(1)} - g_i^{(1)} k_i = y_i^{(1)} z_i^{(1)} (r_i + s),$$

where $\beta_i^{(1)} \in Z_q^*$ satisfies $B_i^{(1)} = \beta_i^{(1)} P$ defined in Eqn. (3) and $\beta_i^{(1)}$ is another unknown. The value of $\beta_i^{(1)}$ depends on $a_i^{(1)}, y_i^{(1)}$, $z_i^{(1)}$, and changes as the choice of h_i, H_3 and H_6 changes. Hence, although four equations $f_i \beta_i^{(j)} - g_i^{(j)} k_i = y_i^{(j)} z_i^{(j)} (r_i + s)$ (where $j = 1, 2, 3, 4$) as shown in [1] are obtained from four different signatures $\{\delta_i^{(j)} = (f_i, g_i^{(j)}), B_i^{(j)}, K_i, R_i\}$ with four different choices of h_1, H_3 and H_6 oracles, the value of s can not be solved since there are seven unknowns $\beta_i^{(j)}$ $(j = 1, 2, 3, 4)$, k_i, r_i and s. Therefore, Γ can not solve the ECDLP problem, which is contraty to the statement given by [1] that the solution of ECDLP problem can be computed. So far, we have pointed out an error in the proof of security analysis.

Moreover, we find out that the signature is untraceable even if TA and RSU collaborate. Upon receiving a message-signature tuple $< M_i, \delta_i = (f_i, g_i), B_i, K_i, R_i, PID_i = (PID_{i1}, pid_{i2}, T_i), T_i >$ from the vehicle V_i, the TA can not get V_i's real identity VID_i even with the help of the RSU. To get VID_i is to compute

$$VID_i = pid_{i2} \oplus h_2(a_i || x_i || T_i) \oplus H_1(r_1 K_{pub})$$

where $a_i = h_1(VID_i)$, x_i is the DL of $PID_{i1} = x_i P$. It is hard to compute a_i without VID_i, and x_i. So, it is impossible to get the real identity VID_i. Hence the signature is untraceable, which is contrary to the statement shown in [1].

V. Conclusion

In this paper, an example of forgery was provided to illustrate the insecurity of Sutrala *et al.*'s scheme. The forger, who could be any malicious vehicle or attacker, produced a valid message-signature tuple that could pass the verification. Besides this, we also found out that it was wrong in the proof of security using the standard random oracle model for Sutrala *et al.*'s scheme, and that the traceability was not satisfied since it was impossible to reveal the real identity of the signer by the TA even with the help of the RSU.

References

[1] A. K. Sutrala, P. Bagga, A. K. Das, N. Kumar, J. J. P. C. Rodrigues, and P. Lorenz, "On the design of conditional privacy preserving batch verification-based authentication scheme for internet of vehicles deployment," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 5, pp. 5535–5548, 2020.

[2] J. K. Liu, T. H. Yuen, M. H. Au, and W. Susilo, "Improvements on an authentication scheme for vehicular sensor networks," *Expert Systems with Applications*, vol. 41, no. 5, pp. 2559–2564, 2014.

[3] D. Pointcheval and J. Stern, "Security proofs for signature schemes," in *International Conference on the Theory and Applications of Cryptographic Techniques*. Springer, 1996, pp. 387–398.

PLACE PHOTO HERE

Yuhao Yang is currently working toward the M.Sc. degree in computer software and theory at Jinan University, Guangzhou, China.

His research interests include cryptographic protocols and information security.

PLACE PHOTO HERE

Xiujie Huang received the M.Sc. degree in applied mathematics and the Ph.D. degree in communication and information system from Sun Yat-sen University, Guangzhou, China, in 2006 and 2012, respectively. She was a Post-Doctoral Fellow with the Department of Electrical Engineering, University of Hawaii at Manoa, Honolulu, HI, USA, from July 2012 to October 2013. Since November 2013, she has been with Jinan University, Guangzhou.

Her current research interests include information theory, information security and their applications.