SER. Finally, the SER computes

*CSP,*2

**MSOC.Eval**(P P R, F, sk*f,ser*, sk*f,csp*, pbk*CSP* , pvk*CSP* , C*Seni* (i = 1, 2, · · · , n*S* )): The cloud server

*f,rec F F*

C*rec,ser* = f

*pk* (r), C*′* = H1(r ∥ C ), (8)

SER firstly decrypts r*i* = f *−*1 (C*i,ser*) by

*CSP ′*

using its secret key sk

*skf,ser*

*f,ser* and checks whether

and sends C*F* = (C*F* , C*rec,ser*, C*F* ) to the receiver

REC.

*′*

C

*i,ser*

= H0(r*i* ∥ C*i,*1 ∥ · · · ∥ C*i,ni* ) holds. If it fails, SER

halts the protocol; otherwise, it sends C*i,i′* , C*i,csp*, C*′*

*i,csp*

to the CSP .

**MSOC.Dec**(P P R, sk*f,rec*, C*F* ): The receiver REC

firstly decrypts

The CSP firstly decrypts N*i* = f *−*1 (C*i,csp*) by using

*skf,csp*

its secret key sk*f,csp* and checks whether C*′* = H0(N*i* ∥

*i,csp*

r = f *−*1

*f,rec*

*sk*

(C*rec,ser* ),

C*i,*1 ∥ · · · ∥ C*i,n* ) holds. If it fails, CSP halts the protocol;

C*CSP,*1 = f *−*1

(C*CSP,*2), (9)

*i* 2*λ F*

*skf,rec F*

otherwise, it randomly selects r*i,csp* ∈*R* {0, 1}

, and re-

by using its secret key sk

. Then, it checks whether

encrypts the blinded inputs

*′*

both

C*CSP,*3

*f,rec*

= H1(C*CSP,*1 ∥ C*CSP,*2

*F*

*F*

*F*

*F*

) and C*′*

= H1(r ∥

C*i,i′* = C*i,i′* mod N*i* = r*i*m*i,i′* mod N*i*,

C*CSP,*2) hold. If it fails, the REC halts the protocol;

C*′ ′* = C*′ ′* mod q, C*′ ′* = C*′ ′* mod p,

*F*

otherwise, it decrypts the result of outsourced computation

*i,i ,q*

*i,i*

*i,i ,p*

*i,i*

”

C

*i,i′*

= (p*−*1p(C*′ ′*

)*q* + q*−*1q(C*′ ′* )*p*

F (m1, m2, · · · , m*n*) = r*−degF* C*CSP,*1. (10)

+r*i,csp*N ) mod T,

*i,i ,q*

*i,i ,p*

*F*

*′*

”

*′*

C

*rec,csp*

”

”

*′*

1*,csp*

= H1(C

* · · · ∥ C*nS ,csp*

∥ C1*,*1

∥ · · · ∥

**5 THE PROPOSED LIGHTWEIGHT PRIVACY- PRESERVING AUTHENTICATION PROTOCOL**

”

C

1*,n*1

∥ · · · ∥ C*nS ,*1

* · · · ∥ C*nS ,nnS*

), (4)

**LPPA**

where p*−*1p 1 mod q and q*−*1q 1 mod p. Finally, the

≡ ≡

CSP sends C*CSP* = (C” *′* , C*′* ) to the cloud server

In this section, a lightweight privacy-preserving authenti- cation protocol LPPA for LBS in VANETs is proposed.

SER.

*i,i*

*rec,csp*

Before giving the description in detail, an efficient and

Without loss of generality, it is assumed that the degree of the j-th item Item*j* = a*j* ∏*n* x*tl,j* (j = 1, 2, · · · , K)

*l*=1

*l*

secure comparison protocol LSCP is firstly devised as the cornerstone of our final design.

of the outsourced multivariate polynomial F is deg*j*. After

receiving C*CSP* , the cloud server SER firstly checks

whether C*′*

*rec,csp*

∥ C

”

*′*

1*,csp*

= H1(C

*′*

*nS ,csp*

∥ · · · ∥ C

∥ · · · ∥ C

1*,*1 ∥

# 5.1 The Proposed LSCP

· · · ∥ C”

1*,n*1

∥ · · · ∥ C”

”

*nS ,nnS*

) holds. If it

In this subsection, based on our proposed MSOC, a

lightweight and secure comparison protocol LSCP is

fails, the SER halts the protocol; otherwise it randomly selects r *R* 0, 1 2*λ* with the condition that r Z*∗T* and computes

*nS ,*1

∈ { } ∈

C” *′* = r*−*1C” *′* , C” *′* = rC” *′* . (5)

proposed, which comprises the following algorithms **LSCP.Setup**, **LSCP.Enc**, **LSCP.Comp** and **LSCP.Dec**. The proposed **LSCP.Setup** and **LSCP.Enc** are the same as **MSOC.Setup** and **MSOC.Enc**.

*i,i ,ser i*

Then, it computes

*i,i*

*i,i ,SER*

*i,i ,ser*

**LSCP.Comp**(P P R, f*jud*, sk*f,ser*, sk*f,csp*, C*m* , C*m* ):

*i j*

*n*

∏(C

C*Itemj*

= r*degF −degj* a*j*

*K*

”

*l,SER*

*l*=1

)*tl,j* ,

Taking the ciphertexts C*mi* = M SOC.Enc(P P R, pbk*i*, pvk*i*, m*i*) and C*mj* = M SOC.Enc(P P R, pbk*j*, pvk*j*, m*j*) of two integers

*′*

*bld*

C*bld* = ∑ C*Item* , C*bld,′* = H1(C*′*

∥ C ), (6)

m*i* and m*j* with λ

= 2λ-bit long as input, the issue

*F j F*

*j*=1

*rec,csp F*

of deciding whether m*i* is larger than m*j* in the

encrypted domain can be transformed into evaluating a

where n = ∑*nS* n*i*, ∪*n* {C” } = corresponding judging polynomial f*jud*(C*m* , C*m* ) in the

*i*=1 *n*

*l*=1

*l,SER*

*nS ,ni*

*i*

*j*

*nS ,ni*

”

∪*i*=1*,i′* =1{C*i,i′ ,SER*}, ∪*l*=1{t*l,j* } = ∪*i*=1*,i′* =1{t*i,i′ ,j* }

and sends C*SER* = (C*bld*, C*′* , C*bld,′* ) to the CSP.

encrypted domain.

We show how to construct the judging polynomial f*jud*

*F rec,csp F*

After receiving C

*′*

*SER*, the CSP checks whether C*bld,′* =

in the plaintext as follows. Firstly, without loss of general-

ity, it is noted that m*i* can be represented in the following

H1(C*rec,csp* C*bld*) holds. If it fails, CSP halts the protocol; otherwise, it computes

*F*

*F*

∥

form (i.e. C*mj*

can be performed the same)

*CSP,*1

C*F* = C*F* mod N,

*bld*

m = m *′*

2*λ′ −*1 + m *′*

2*λ′ −*2 + · · · + m

. (11)

C*CSP,*2 = f*pk* (C*CSP,*1),

*i*

*i,λ −*1

*i,λ −*2

*i,*0

Therefore, the binary representation of m*i*, namely

*F f,rec F*

C*CSP,*3 = H1(C*CSP,*1 ∥ C*CSP,*2), (7) m*i,λ′ −*1m*i,λ′ −*2 · · · m*i,*0 can be straightforwardly derived

*F F F*

and sends C*CSP* = (C*CSP,*2, C*CSP,*3) back to the cloud server SER.

*F F F*

by bit decomposition.

Then, it is observed that for single bit comparison between m*i,s*(s = 0, 2, · · · , λ*′* − 1) and m*j,s*, we can

obtain the judging polynomial f

|  |  |
| --- | --- |
| Notation | Description |
| *mi,j* | The *j*-th LBS message generated by vehicular user *Ui* |
| *Indexi,j* | The message identifier of *mi,j* |
| (*xi,j , yi,j* ) | The coordinates denoting the location where LBS  message *mi,j* is collected |
| *ti,j* | The time when LBS message *mi,j* is collected |
| *Ri,k,j* | The rating of vehicular user *Ui* on user *Uk*’s *j*-th LBS  message *mk,j* |
| *S*(*Ui, Ut*) | The similarity between vehicular users *Ui* and *Ut* |
| *REDk,j′* | The redundancy factor denoting whether the *j′* -th LBS  message *mk,j′* of vehicular user *Uk* is redundant |
| *P Ri,k,j′* | The predicted rating of vehicular user *Ui* on LBS  message *mk,j′* |
| *Ta* | The threshold for LBS message filtering |

*jud,l*

(m*i,s*

, m*j,s*

)(s =

TABLE 2: Notation Description for LPPA

1, 2, , λ*′* 1; l = 1, 2, 3) by the truth table method such that

· · · −

f*jud,*1(m*i,s*, m*j,s*) = m*i,s* − m*i,s*m*j,s* = 1

if and only if m*i,s* > m*j,s*,

f*jud,*2(m*i,s*, m*j,s*) = 2m*i,s*m*j,s* − m*i,s* − m*j,s* + 1 = 1

if and only if m*i,s* = m*j,s*,

f*jud,*3(m*i,s*, m*j,s*) = m*j,s* − m*i,s*m*j,s* = 1

if and only if m*i,s* < m*j,s*. (12)

By Eqn. (12) we can transfer the single bit comparison to checking whether the corresponding judging polynomial f*jud,l*(m*i,s*, m*j,s*) equals 0 or 1. Therefore, the comparison between two integers m*i* and m*j* of size λ*′* can be achieved by sequentially performing the bit comparison from the most significant bit to the least significant one and a

binary chopping method would enhance the efficiency of

M SOC.Evl(P P R, f*jud*, sk*f,ser*, sk*f,csp*, C*mi* , C*mj* ).

**LSCP.Dec** is the same as **MSOC.Dec**. If the decryption result f*jud*(m*i*, m*j*) = 1, the receiver decides m*i* > m*j*;

otherwise, m*i* ≤ m*j*.

comparison. Let L =

*′*

⌈ 2 ⌉

*λ*

, we have

# 5.2 The Proposed LPPA

m*i* = m*i,λ′ −*1, · · · , m*i,L* m*i,L−*1, · · · , m*i,*0,

s ˛¸ x s ˛¸ x

*mi,h mi,l*

m*j* = m*j,λ′ −*1, · · · , m*j,L* m*j,L−*1, · · · m*j,*0 . (13)

s ˛¸ x s ˛¸ x

*mj,h mj,l*

As same as is used for constructing the judging polynomial for bit comparison, the judging polynomial f*jud*(m*i*, m*j*) between integers m*i* and m*j* can be constructed by re- cursively exploiting the following judging polynomial for m*i,h*, m*i,l*, m*j,h*, m*j,l* with the binary chopping method

f*jud*(m*i*, m*j*)

= f*jud,*1(m*i,h*, m*j,h*)(1 − f*jud,*2(m*i,h*, m*j,h*))

(1 − f*jud,*3(m*i,h*, m*j,h*))(f*jud,*2(m*i,l*, m*j,l*)

(1 − f*jud,*1(m*i,l*, m*j,l*))(1 − f*jud,*3(m*i,l*, m*j,l*))

+f*jud,*3(m*i,l*, m*j,l*)(1 − f*jud,*2(m*i,l*, m*j,l*))

(1 − f*jud,*1(m*i,l*, m*j,l*))) + f*jud,*1(m*i,l*, m*j,l*)

(1 − f*jud,*2(m*i,l*, m*j,l*))(1 − f*jud,*3(m*i,l*, m*j,l*))

(f*jud,*1(m*i,h*, m*j,h*)(1 − f*jud,*2(m*i,h*, m*j,h*))

(1 − f*jud,*3(m*i,h*, m*j,h*)) + (1 − f*jud,*1(m*i,h*, m*j,h*))

f*jud,*2(m*i,h*, m*j,h*)(1 − f*jud,*3(m*i,h*, m*j,h*))) (14) such that

f (m , m ) = { 1, m*i* > m*j* ,

*jud*

*i*

*j*

0, otherwise. (15)

In this subsection, based on our proposed MSOC and LSCP, a lightweight privacy-preserving authentication protocol LPPA for location-based services in VANETs is proposed, by devising an efficient information filtering system in the encrypted domain. It is assumed that there exist n*u* vehicular users U*i*(i = 1, 2, , n*u*) in district s(s = 1, 2, , n*d*) managed by RSU*s*. To efficiently achieve fine-grained encrypted LBS message access control and permit the authorized vehicular users to successfully decrypt LBS services, a ciphertext-policy attribute-based encryption (CP- ABE) is adopted, which is composed of the algorithms ABE.Setup(1*λ*), ABE.Keytten(M SK, S), ABE.Enc(P P AR, m, A), ABE.Dec(P P AR, C, SK). For LBS

message authentication, an existentially unforgeable secure signature scheme Λ under adaptively chosen message attack is also adopted, which is composed of the algorithms Λ.Keytten(1*λ*), Λ.Sign(sk, m), Λ.V erif y(pk, m, σ).

· · ·

· · ·

Table 2 shows the notations used in LPPA. The proposed LPPA comprises the following four algorithms: **Setup**, **LBS Message Generation**, **LBS Message Filtering** and **LBS Message Decryption and Verification**, which are presented as follows.

**Setup**: On input 1*λ* where λ is the security parameter, it runs the algorithm M SOC.Setup(1*λ*) to generate

pairs of public key and secret key (pk

*f,RSUs*

, sk*f,RSUs*

),

In our case, owing to the property of full homomorphism of our proposed MSOC and the fact that both the bit decomposition (i.e. the ciphertext of power of 2 can be also generated by senders Sen*i* or Sen*j*) and the recursive judging polynomial calculation only require multivariate polynomial evaluation, the cloud SER can calculate the judging polynomial f*jud*(m*i*, m*j*) in the encrypted domain namely C*fjud*(*mi,mj* ) = f*jud*(C*mi* , C*mj* ), by exploiting the algorithm

*f,RSUs*

(pk*f,csp*, sk*f,csp*) and (pk*f,Ui* , sk*f,Ui* ) respectively for the RSU*s*(s = 1, 2, , n*d*), the CSP and each vehicular user U*i*(i = 1, 2, , n*u*), where f, f *−*1 on

{ }

· · ·

· · ·

0, 1 2*λ* is a pair of one-way trapdoor permutations. H0, H1 : 0, 1 *∗* 0, 1 2*λ* are cryptographic hash functions. It also runs ABE.Setup(1*λ*) to generate

{ } → { }

public parameter P P AR*ABE* and master secret key M SK*ABE*. The public parameters are P P R = (pk*f,RSUs* , pk*f,csp*, pk*f,Ui* , P P AR*ABE*, H0, H1) and the secret keys are sk*f,RSUs* , sk*f,csp*, sk*f,Ui* , M SK*ABE*

respectively kept private by RSU*s*, CSP , U*i* and the system.

*i*

public keys of CSP and user U*l*, re-encrypts it as

**LBS Message Generation**: Without loss of

*i*

*l,j,x*

C

*i*

C

*l,j,y*

= f*pkf,RSUs*

= f*pkf,RSUs*

*i*

*l,j,x*

(r

*i*

(r

*l,j,y*

),

*i*

), C

*l,j,t*

= f*pkf,RSUs*

*l,j,t*),

generality, it is assumed that each vehicular user U*i* generates its j-th LBS message m*i,j*(i = 1, 2, · · · , n*u*; j = 1, 2, · · · , n*i*) associated to the tuple

(r

),

*i,′*

*l,j,x i,′*

(r

C

C

*l,j,y*

= f*pkf,csp*

= f*pkf,csp*

*i,′*

*l,j,x i,′*

(r

(r

*l,j,y*

),

*i,′ l,j,t*

), C

= f*pkf,csp*

*i,′ l,j,t*

Inf o*i,j* = (U*i*, U*i*, Index*i,j*, x*i,j*, y*i,j*, t*i,j*, r*i,i,j*) where

C*i,′*

= r*i*

r*i,′*

C*Loc* ,

Index*i,j*, (x*i,j*, y*i,j*),t*i,j* and r*i,i,j* denote the message

*Locl,j,x*

*l,j,x l,j,x*

*l,j,x*

identifier of m*i,j*, the coordinates of the location and the

*i,′*

*Locl,j,y*

C

*i*

*l,j,y*

= r

*i,′*

*l,j,y*

r

C*Loc*

*l,j,y* ,

time the message m*i,j* is collected, and the rating of user

C*i,′*

= r*i*

r*i,′* C

. (18)

U*i* on its generated LBS message m*i,j*. It is assumed

*tl,j*

*l,j,t l,j,t*

*tl,j*

that each U*i* rates its own messages with the highest score, namely r*i,i,j* = 5 where the rating values range from 1 to 5 as integers. A rounding function is applied

Finally, user U*i* constructs table T *′*

comprising the tuples Inf o*′* =

*tl,j*

*i*

*i,l,j*

on the location coordinates and the time to guarantee

*Locl,j,x*

(U*i*, U*l*, Index*l,j*, C*i,′*

*i,′*

*Locl,j,y*

, C

, C*i,′* , C*R*

*i,l,j*

) and

(x*i,j*, y*i,j*, t*i,j*) to be integers for cryptographic exploitation.

sends T *′* to the RSU*s*. Note that the blinding factors

Let

· · ·

*i*

*i i i*

*i,′*

*i,′*

*i,′*

r*i,k,j*(k = 1, 2, , n*u*) be the user U*i*’s rating on user

U*k*’s j-th message m*k,j*(j = 1, 2, , n*k*). In the following two cases that U*i* has not received m*k,j* from other users or that U*i* decides m*k,j* as redundant LBS message, the rating r*i,k,j* is set to 0. Each vehicular user U*i* and RSU*s* respectively generates (pbk*i*, pvk*i*) and (pbk*s*, pvk*s*) by exploiting the algorithm M SOC.Keytten(P P R). It also derives sk*ABE,i*, (pkΛ*,i*, skΛ*,i*) for user U*i*, by running ABE.Keytten(M SK*ABE*, S*i*) and Λ.Keytten(1*λ*) where S*i* is the attribute set of user U*i*.

· · ·

1) Each U*i* generates the encrypted LBS message for m*i,j*

C*i,j* = ABE.Enc(P P AR*ABE*, m*i,j*, A),

σ*i,j* = Λ.Sign(skΛ*,i*, C*i,j*),

C*Loci,j,x* = M SOC.Enc(P P R, pbk*i*, pvk*i*, x*i,j*), C*Loci,j,y* = M SOC.Enc(P P R, pbk*i*, pvk*i*, y*i,j*), C*ti,j* = M SOC.Enc(P P R, pbk*i*, pvk*i*, t*i,j*), (16)

where A is the access policy permitting that the authorized LBS users can successfully decrypt the

r*l,j,x*, r*l,j,y* , r*l,j,t* and r*l,j,x*, r*l,j,y* , r*l,j,t* are respectively

adopted in the re-encryption to achieve the interest pattern

privacy of vehicular users (i.e. please refer to Sec. 6.2 for the security proof).

**LBS Message Filtering**: The roadside unit RSU*s*(s = 1, 2, · · · , n*d*) predicts the rating in the encrypted domain for each vehicular user U*i*(i = 1, 2, · · · , n*u*) currently travelling in district s on LBS message m*k,j′* which U*i* has not rated before.

1. For each LBS message m*k,j′* (k = 1, 2, · · · , n*u*; j*′* = 1, 2, , n*k*) as the j*′* -th message generated by user U*k*, for each tuple in T *′* sent by user U*i*, RSU*s* computes the function

*i*

· · ·

F*dist*(Inf o*k,j′* , Inf o*l,j*)

= (Loc*k,j′ ,x* − Loc*l,j,x*) + (Loc*k,j′ ,y* − Loc*l,j,y*)

2 2

+(t*k,j′* − t*l,j*)2 (19)

in the encrypted domain by exploiting the algorithm

underlying LBS message m*i,j*, and broadcasts Inf o*i,j* = (U*i*, U*i*, Index*i,j* , C*i,j* , σ*i,j* , C*Loci,j,x* , C*Loci,j,y* , C*ti,j* ) in

C*Dist*

*k,j ,l,j*

*′*

= M SOC.Eval(P P R, F*dist*, sk*f,RSUs* ,

sk*f,csp*, T *′* , T *′* ), (20)

its neighborhood. *k i*

*k*

1. Each vehicular user U*i* initializes and updates a table T*i* by storing the encrypted tuples Inf o*l,j* =

*i*

where T *′* is the table maintained and updated by user U*k*. Note that to evaluate on the re-encryption ciphertexts (i.e.

(U*i*, U*l*, Index*l,j*, C*Locl,j,x* , C*Locl,j,y* , C*tl,j* , C*Ri,l,j* )(l =

*i,′*

*Locl,j,x*

we take C

in the updated table T *′* for example), the

1, 2, · · · , n*u*; j = 1, 2, · · · , n*l*) associated to m*l,j* as the

CSP firstly deciphers r*i,′*

= f *−*1

(C*i,′*

), computes

j-th LBS message it generated itself if l = i or accepted

*l,j,x*

*′*

C

= ((r

)*−* C

*skf,csp l,j,x*

as the j-th generated LBS message from other vehicles U*l*

*Locl,j,x q*

*i,csp i,* 1 *i,′*

*Locl,j,x l,j,x Locl,j,x*

*Locl,j,x q*

*l,j,x*

) mod N*l*, and C*i,ser,bld* =

if l i, where

*Locl,j,x*

p*−*1p(C*i,csp*

)*q* + q*−*1q(C*i,csp* )*q* + r*i,csp*N mod T ,

where r*i,csp* ∈*R* {0, 1}*λ*. Afterwards the RSU*s* is also

*l,j,x*

C*Ri,l,j*

(C

= M SOC.Enc(P P R, pbk*i*, pvk*i*, R*i,l,j*) (17)

required to compute r*i*

= f *−*1

*f,RSUs*

*sk*

*i*

*l,j,x*

) and exe-

cute an additional deblinding operation that C*i,ser* =

*l,j,x*

*Locl,j,x*

is the ciphertext of its rating R*i,l,j* on each LB-

r*−*1(r*i*

)*−*1C*i,ser,bld*

where r*l* is the blinding factor

S message m*l,j* located in table T*i*. Then, for *l*

*l,j,x*

*Locl,j,x*

each tuple in table T , user U

randomly select-

adopted to encrypt Loc*l,j,x* by the algorithm M SOC.Enc

*i*

s r

*l,j,x*

*i,′*

*l,j,x*

, r

*i*

*l,j,y*

, r

*i,′*

*l,j,y*

, r

*i*

*i*

, r

*l,j,t*

*i,′ l,j,t*

, r

*i*

∈*R* {0, 1}2*λ* with

, r

*i*

*k*

in Eqn. (16) and can be decrypted by RSU*s* using its

secret key sk*f,RSUs* . The same operations are required to

the conditions that r*i*

*l,j,x*

*i*

*l,j,y*

, r

*i*

*l,j,t*

∈ Z*∗T* and

be performed on other re-encryption ciphertexts both in the

r*i,′*

*l,j,x*

*l,j,y*

*l,j,t*

*l*

, r*i,′*

, r*i,′*

∈ Z*T* , where T, T*l* are the temporary

updated table T *′* and T *′* .

Then, RSU*s* computes

C*Td* = M SOC.Enc(P P R, pbk*s*, pvk*s*, T*d*), (21)

where T*d* is the threshold to decide whether an LBS mes- sage is redundant, and for each tuple in table T *′* associated

U*i* discards LBS message m*k,j′* and stop it from further transmission; otherwise, U*i* accepts and recovers m*k,j′* by performing the ABE decryption algorithm m*k,j′* = ABE.Dec(P P AR*ABE*, SK*ABE,i*, C*k,j′* ). Final- ly, U*i* gives a rating R*i,k,j′* on LBS message m*k,j′* and adds

to user ’s

*i*

, com-

the tuple (U , U , j*′* , C

, C , C , C )

U*l* j-th LBS message accepted by user U*i*

*′*

*i k Loc*

*k,j ,x*

*′*

*′*

*′*

*′*

*Loc*

*k,j ,y*

*t*

*k,j*

*R*

*i,k,j*

pares Dist*k,j′ ,l,j* (l = 1, 2, · · · , n*u*; j = 1, 2, · · · , n*l*; j = 1, 2, · · · , n*k*) and T*d*, namely evaluating the judging poly- nomial f*jud*(Dist*k,j′ ,l,j* , T*d*) in the encrypted domain by exploiting the algorithm

into its table T*i* by performing Step 2) in the algorithm

# LBS Message Generation.

**Remark:**(Aggregated LBS bundles) It is noted that the techniques of aggregate signature and multi-signature

C*f ′*

= LSCP.Comp(P P R, f*jud*, sk*f,RSUs* ,

*jud,k,j ,l,j*

sk*f,csp*, C*Dist* , C*T* ). (22)

*k,j ,l,j*

*′ d*

[6] can be exploited by each vehicular user to compress all her/his accepted LBS messages originally generated

2) RSU*s* computes the similarity S(U*i*, U*t*)(i, t = 1, 2, · · · , n*u*) between users U*i* and U*t*

∑∑*nu nl* 2

(

*l*=1

*j*=1

R*i,l,j* R*t,l,j* )

S(U , U ) =

(23)

R

R

and signed by different users, into a single bundle (i.e. the useless but not duplicate LBS messages can also be aggregated into a single bundle in the same way). Then, the specific vehicular user rates on both the accepted

and the useless but not redundant LBS bundle, and each

*i t nu*

∑∑ *nl*

∑∑ *nl*

*l*=1

*j*=1

2

*i,l,j*

*nu l*=1

*j*=1

2

*t,l,j*

(23)

and the useless but not redundant LBS bundle, and each

RSU*s*(s = 1, 2, · · · , n*d*) can predict the ratings on the

in the encrypted domain, by performing the algorithm

C*S*(*Ui,Ut*) = M SOC.Evl(P P R, S(U*i*, U*t*), sk*f,RSUs* ,

sk*f,csp*, T *′* , T *′* ), (24)

bundles for other users based on their similarities in Eqn. (25). The communication cost on each vehicular user would be further reduced (i.e. please refer to Sec. 7.2 for performance evaluation).

*i t*

where T *′* is the table maintained and updated by user U*t*. Then, it predicts user U*i*’s rating on LBS message m*k,j′* , by computing

*t*

∑

**6 SECURITY PROOF**

In this section, we firstly give the formal security proof of our proposed multi-key secure outsourced computation

*nu*

*′ ′ t*=1*,t̸*=*i*

R*t,k,j*

*i*

*′* S(U*i*, U*t*)

*i*

*t*

(25)

scheme MSOC. Then, based on the primitive of MSOC, we

P R*i,k,j*

*t*=1*,t*

= RED*k,j*

∑*nu*

S(U , U )

elaborate that our proposed lightweight privacy-preserving

authentication protocol LPPA for location-based services in

in the encrypted domain through performing the algorithm

C*P R ′* = M SOC.Evl(P P R, P R*i,k,j′* , sk*f,RSUs* ,

*i,k,j*

VANETs achieves the security goals.

# 6.1 Security for the Proposed MSOC

sk*f,csp*, C*RED* , T *′* , C*S*(*U ,U* )), (26)

*k,j′ t i t*

Before giving the security proof, we present the correctness

where

C*RED*

∏∑*nu nl*

of our MSOC that serves the primitive of LPPA. In Eqn.

is (7), by exploiting Chinese Remainder Theorem and Euler’s

*k,j′*

=

C*f*

the encrypted redundancy

*j*=1

factor.

*l*=1

*jud,k,j′ ,l,j*

Finally, RSU*s*

Theorem [29], we have

transmits all the encrypted prediction ratings *CCSP,*1 = *Cbld mod N*

*F F*

C*P R ′* (k = 1, 2, · · · , n*u*; j*′* = 1, 2, · · · , n*k*) to vehicular *K n*

*i,k,j*

user .

*deg*

*−deg* ∑ ∏ ” *t*

U*i*

# LBS Message Decryption and Verification: While

= *r F*

*j aj*

*j*=1

*K n*

*l,q*

*l*=1

(*Cl,SER*) *l,j mod N*

receiving a newly-arriving LBS message m*k,j′* , vehicular

= *rdegF* ∑ *aj* ∏(*p−*1*pmq*

+ *q−*1*mp*

+ *ri,cspN* )*tl,j*

user U*i* firstly recovers the prediction rating by performing

*l,p*

*j*=1 *l*=1

*mod N*

P R*i,k,j′* = M SOC.Dec(P P R, sk*f,Ui* , C*P R ′* ), (27)

*i,k,j*

= *rdegF* ∑ *a* (*p−*1*p*(∏ *mtl,j* )*q* + *q−*1*q*(∏ *mtl,j* )*p*)

*i,k,j*

and compares it to the predefined threshold T*a*. If

P R*i,k,j′* = 0, U*i* considers LBS message m*k,j′* to

*K*

*j*

*j*=1

*mod N*

*K*

*n*

*l q*

*l*=1

*n*

*n*

*l p*

*l*=1

*K n*

be duplicate to the messages she/he has accepted,

= *rdegF* (*p−*1*p*(∑ *a*

∏ *mtl,j* )*q* + *q−*1*q*(∑ *a*

∏ *mtl,j* )*p*)

discards and prevents it from being further broadcasted in the neighborhood; if 0 < P R*i,k,j′* < T*a*, user

*mod N*

*K*

*n*

*ml*

*j*=1

*j l q*

*l*=1

*j*=1

*j l p*

*l*=1

U*i* considers LBS message m

not redundant one without further authentication and transmit it to other vehicles in her/his neighborhood;

*k,j*

*′* as a useless but

= *rdegF* ∑ *a* ∏

*j*=1

*l*=1

*tl,j* = *rdegF F* (*m , m , · · · , m* )*.*

*j*=1

*l*=1

otherwise, U*i* decides LBS message m*k,j′* as a useful one and verifies the signature by performing

*j*

1

2

*n*

Therefore, the authorized receiver possessing the secret key

sk*f,rec* can successfully recover F (m1, m2, · · · , m*n*) =

*F*

the algorithm Λ.V erif y(pkΛ*,k*, C

*k,j*

*′* , σ

*k,j*

*′* ). If it fails,

r*−degF* C*CSP,*1.

We firstly give the security proof of the data privacy of the subset of uncorrupted senders N*Sen* T in our proposed MSOC against the collusion attack between the CSP, the

\

the encryption of m*i,i′ ,β* as the challenge ciphertext c*∗i,i′* . After receiving c*∗ ′* , A can continue to make queries to the

oracles O*H*0 and O*Dec* with the restriction that c*∗ ′* cannot

*i,i*

subset of corrupted senders T and malicious receivers. We also take input privacy to detail the security proof and the

be queried to the decryption oracle query phase.

O*Dec*

*i,i*

in the adaptive

proofs for the output privacy and the collusion with the

cloud server can be similarly derived.

*Theorem 1:* (**Data Privacy for MSOC**) Let be a malicious adversary defeating the CCA2 security for data privacy of our proposed MSOC with a nonnegligible ad-

A

To explain the interaction perfectly simulates the real environment of the adversary running with its oracles, we study the following events. Let S be the event that for some ciphertext s*′* ∥ d*′* ∥ h*′* , A made some query r*i* ∥ d*′*

to the oracle O*H*0 satisfying f (r ) = s*′* . Then, we further

A

vantage defined as ϵ*′,ploy*(*λ*), where poly(λ) refers to the

total number of queries made to the oracles and λ is the

let

*i*

R be the event that A made some query s*′*

* d*′*
* h*′*

to the decryption oracle O*Dec* where h*′* = H0(r*i* ∥ d*′* )

security parameter. There exists a simulator B who can use

holds without making any query (f *−*1

*f,ser*

*sk*

(s*′* ) ∥ d*′* ) to the

to invert the one-way trapdoor permutation f with the nonnegligible probability ϵ that:

A

*′,poly*(*λ*) p oly(λ)

ϵ ϵ . (28)

≥ − 2*λ−*1

*Proof:* We take input privacy to detail the security proof. Intuitively, although the adversary considered as the collusion between the CSP, the malicious receiver and a subset of corrupted senders holds secret key sk*f,csp*

A

H0-oracle *H*0 . Let poly(λ) be the total number of oracle queries made by the adversary . Then, we can conclude that

*Suc*

P r[A ]

A

O

= P r[A |R]P r[R] + P r[A |R¯ ∧ S]P r[R¯ ∧ S]

*Suc Suc*

+P r[A |R¯ ∧ S¯]P r[R¯ ∧ S¯]

*Suc*

≤ poly(λ)2*−λ* + P r[S] + ,

1

that can be used to compute N*i* = f *−*1

(C*i,csp*) and 2

*skf,csp*

and P r[A

2

C*′* = C*i,i′* mod N*i* = r*i*m*i,i′* mod N*i*, the input m*i,i′*

*i,i′*

since

P r[R] ≤

*p oly*(*λ*) 2*λ*

*Suc*|R¯ ∧ S¯] = 1 can

cannot be derived without the knowledge of r*i* encrypted in C*i,ser* = f*pkf,ser* (r*i*). Therefore, we can reduce the CCA2 security for input privacy to the inverse of one-way trapdoor permutation f without secret key sk*f,ser* and the proof

be straightforwardly derived. Finally, it is observed that

the probability of simulator to fail in behaving like the adversary in inverting the one-way trapdoor permutation f can be bounded by P r[R]. Therefore,

A

B

is given by contradiction. In the initialization phase, the system performs (f, f *−*1) ← G(1*λ*), y*i* = f*pk* (r*i*) and

*f,ser*

ϵ ≥ ϵ

*′,poly*(*λ*)

poly(λ)

,

−

2*λ−*1

the simulator B tries to solve r*i* = f *−*1 (y*i*). The adver-

*skf,ser*

which is also non-negligible. Therefore, theorem 1 holds.

sary is given the public parameter P P R, the secret keys sk*f,csp*, sk*f,rec* of the corrupted CSP and the corrupted re- ceiver and all the temporary secret key pvk*CSP* , pvk*i* of the corrupted CSP and all corrupted senders Sen*i* ∈ T . There

A

*Theorem 2:* (**Security for the Whole Protocol**) The proposed MSOC securely implements the functionality

F un, namely for every real world adversary A, there exists

are two oracles, namely O*H*0 and O*Dec*. B can perform the

an ideal world adversary with access to in a black-

simulations by answering the queries from the adversary as follows. For the collusion between the CSP, malicious receivers and a subset of corrupted senders, we mainly focus on the ciphertext components C*i,ser*, C*i,i′* , C*′* in

*i,ser*

C*Seni* .

*H*0 **Query**. If a query r*i* C*i,*1 C*i,n* to *H*0 satisfies f (r*i*) = y*i*, then outputs r*i* and halts; otherwise, it returns a random element Str0 *R* 0, 1 2*λ* as the response to the adversary and remains the triple

∈ { }

O B

*i*

O ∥ ∥ · · · ∥

(r*i*, C*i,*1 ∥ · · · ∥ C*i,ni* , Str0) in the H0-list.

O*Dec* **Query**. To answer the query s*′* ∥ d*′* ∥ h*′* to O*Dec*

where d*′* = ∪*ni* C *′* , the simulator B firstly checks if

*i′* =1 *i,i*

box manner such that forSall input vectorsAm⃗ , we have

Ideal*F un,S* (m⃗ ) *c* Real*MSOC,A*(m⃗ ).

≈

*Proof:* Based on the data privacy (indistinguishability)

proved in Theorem 1, we prove this theorem when the server is corrupted via a series of hybrid games, by using an ideal/real paradigm. The proofs for other cases of a corrupted CSP, corrupted sender, corrupted receiver and their collusion can be analogously derived.

**Game 0**. This is the real world execution of our proposed MSOC.

**Game 1**. Instead of executing **MSOC.Dec** where the

honest receiver uses its secret key, we run the simulator

there exists a triple (r*i*, d*′* , h*′* ) in the H0-list. If it does, S

interacting with the adversary A. Owing to

the simulator B further checks whether s*′* = f*pk* (r*i*)

*f,ser*

holds. If not, the simulator B returns invalid; otherwise it

*M SOC.Dec*

the data privacy (i.e. we mean output privacy here) of

our proposed MSOC, if the ideal decryption functionality

is correctly emulated, the joint output is computationally

computes and returns C*′ ′* = r*−*1d*′* to the adversary A and

*i,i*

*i*

A computes m*i* = C*′* mod N*i*.

*i*

Then, the adversary A submits two challenge plaintexts m*i,i′ ,*0, m*i,i′ ,*1 associated to an uncorrupted sender Sen*i* ∈ N*Sen* \ T of the same size |m*i,i′ ,*0| = |m*i,i′ ,*1| = 2λ, and the simulator B randomly selects β ∈ {0, 1} and returns

indistinguishable in a real world execution of our proposed MSOC with the adversary , and in a ideal world execution with the adversary *MSOC.Dec*.

**Game 2**. In this game, by replacing computing

S

A

yˆ = M SOC.Dec(P P R, sˆk*f,rec*, C*F* ), the joint output is