KEIO 150
Design the Future

Privacy Protection for Low-Cost RFID Tags in Iot System

Ye Li, Fumio Teraoka
Keio University,
Graduate School of Science and Technology

Contents

KEIO UNIVERSITY
CALAMVS GLADIO FORTIOR
1858

I. Introduction

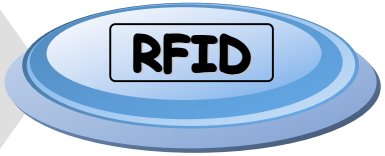
- RFID: an enabler of IoT

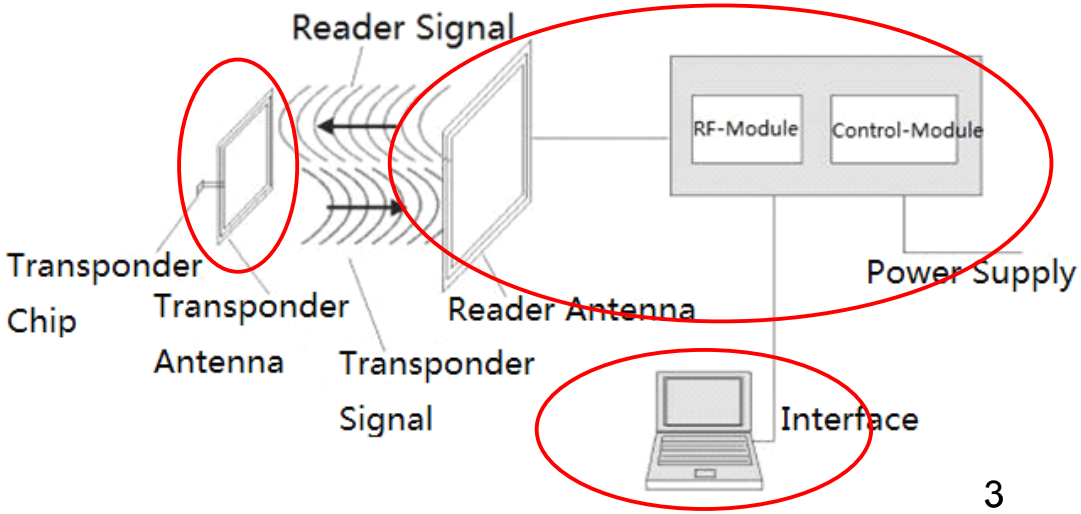
Being attached almost to everything making it intelligent. If all objects were equipped with RFID tags, they could be identified by the PC.

RFID *tag*, or *transponder*

RFID *reader*, or *transceiver*

back-end database





I. Introduction

- Low-cost RFID tags:

Lacking resources to perform true cryptographic operations.

- Research challenges:

The communication channel between the tag and the reader is insecure. Hence, the low security performance may result in leakage of personal information.

II. Security Requirements for low-cost Tags

- Confidentiality

All of the information in the protocol is securely transmitted.

- Indistinguishability

The sent information from the tag or the reader should not be different from the sent information of other tags.

- Forward Security

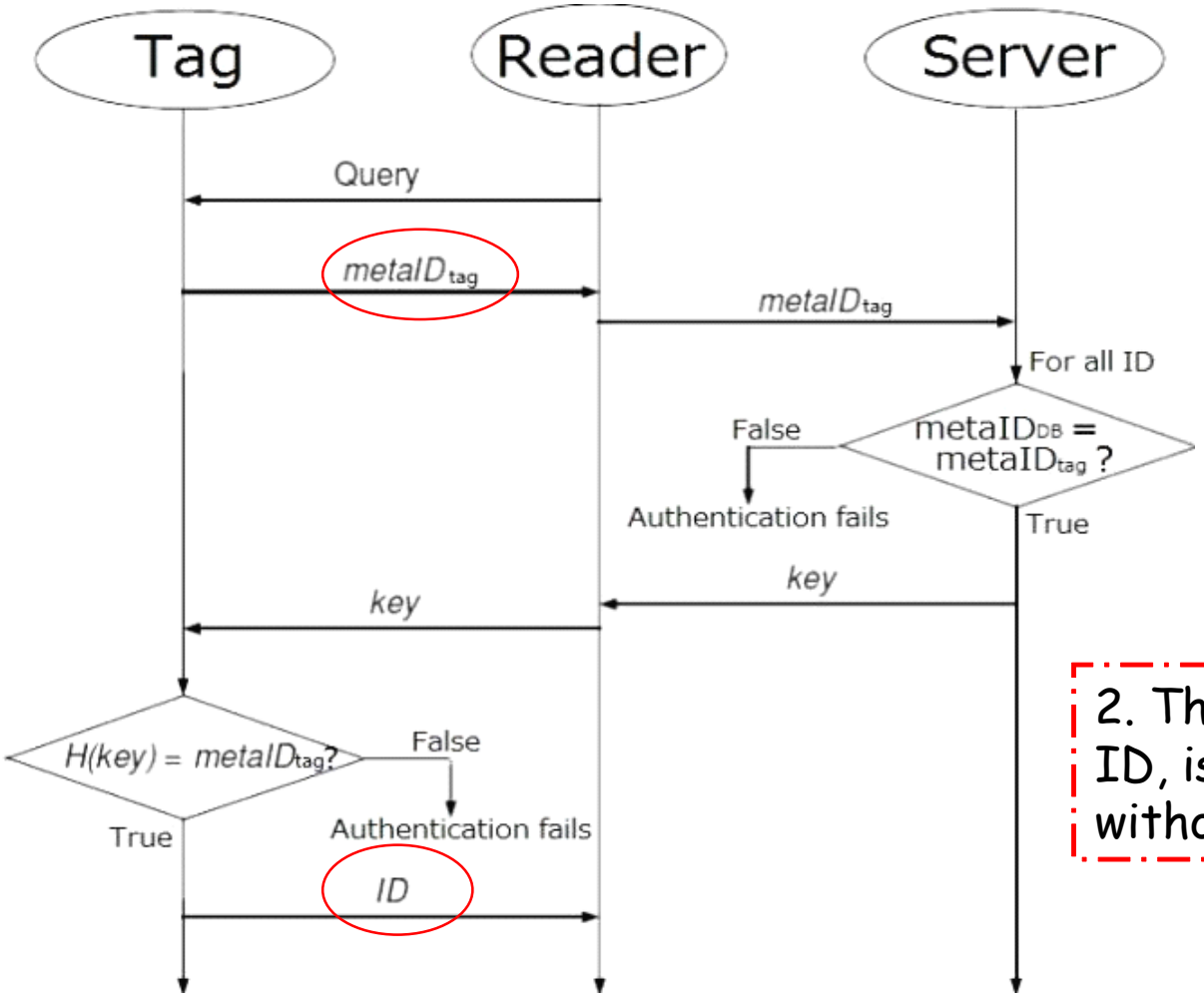
The previously sent information cannot be tracked using the present information of the tag.

- Mutual Authentication

Unlike the more common RFID authentication protocols where only one side (either the reader or the tag) authenticate the other.

III. Related work

- Hash Lock protocol



Challenge

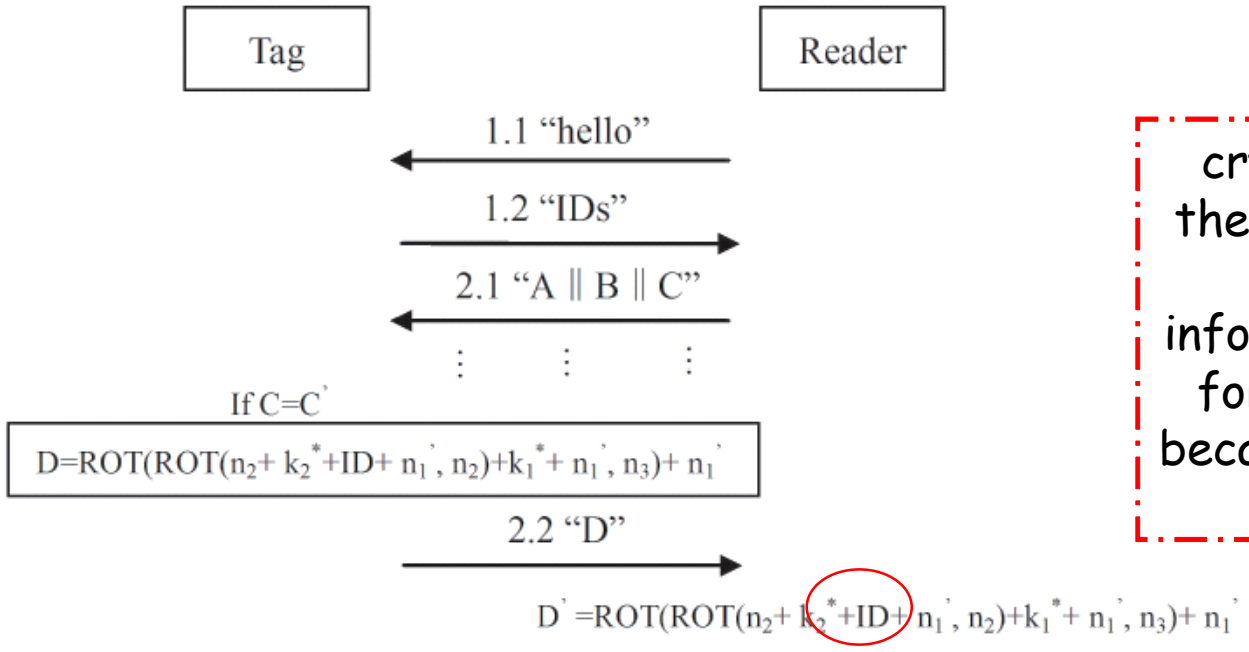
1. The $metaID_{tag}$ keeps same in each authentication process.

2. The tag's identifier, ID, is directly sent without any encryption.

III. Related work

Gossamer Protocol

The protocol is made up of three stages: (1) tag identification, (2) mutual authentication, and (3) index pseudonym (IDS) and key updating.



Challenge

cryptanalysis shows that the attack is possible when the tag transmits information D to the reader for authenticating itself, because it contains the tag's information ID,

III. Related work

- security issues of the existing protocols

	Hash Lock	Randomized Lock	one time password	Gossamer
Confidentiality	○	○	○	△
Forward security	×	×	×	○
Mutual authentication	×	×	○	○
Eavesdropping	×	×	△	△
Track attacking prevent	×	○	○	○

- A new protocol based on the ideas of hash locker and mutual authentication mechanism is proposed.

IV. The proposed protocol

- Assumptions and Notations

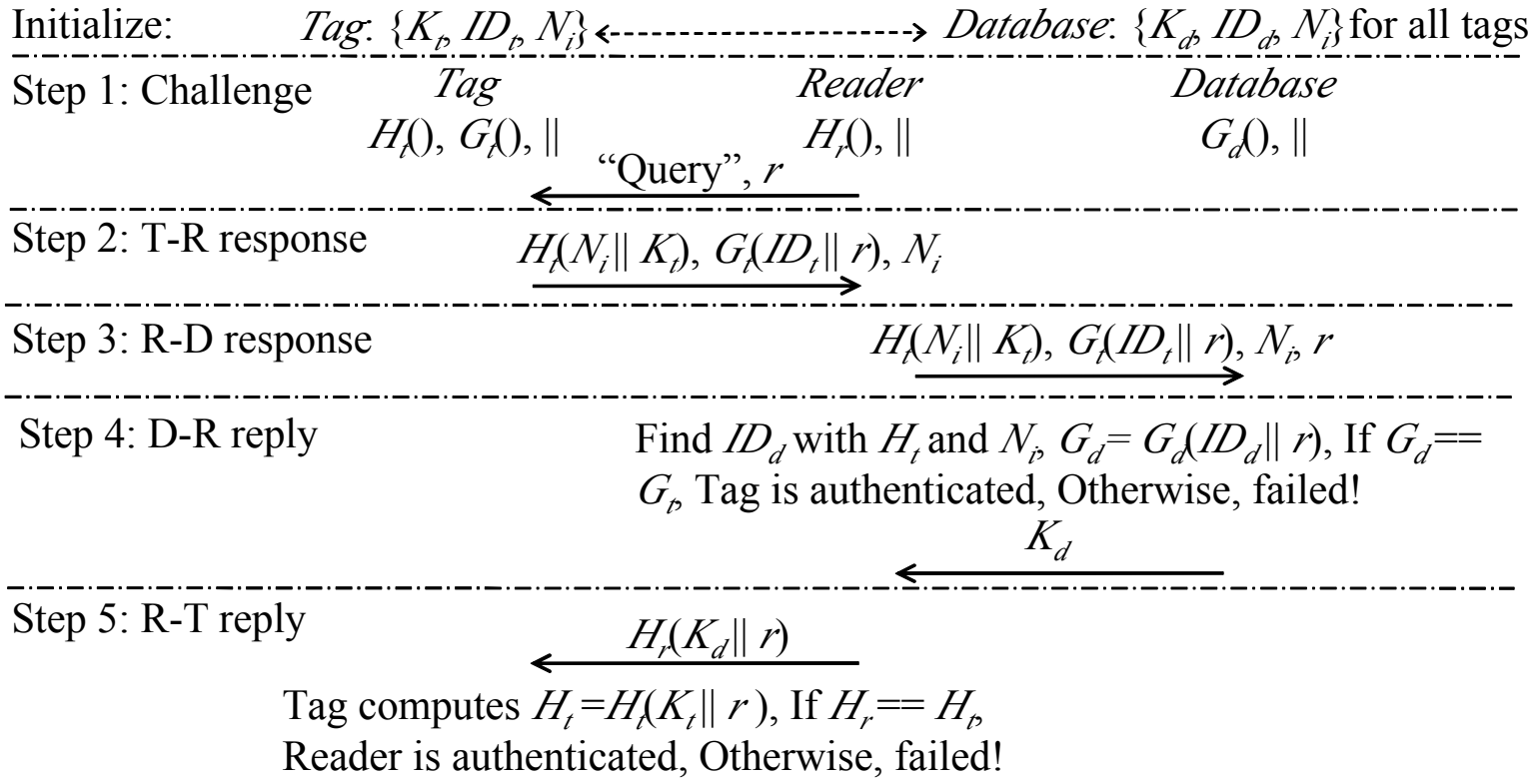
1. A tag is passive and has a rewritable memory such as EEPROM with reasonable size.

2. The communication channel between the reader and the back-end database is secure.

3. The cryptographic hash function in the protocol requires security of preimage resistance, 2nd-preimage resistance, collision avoidance.

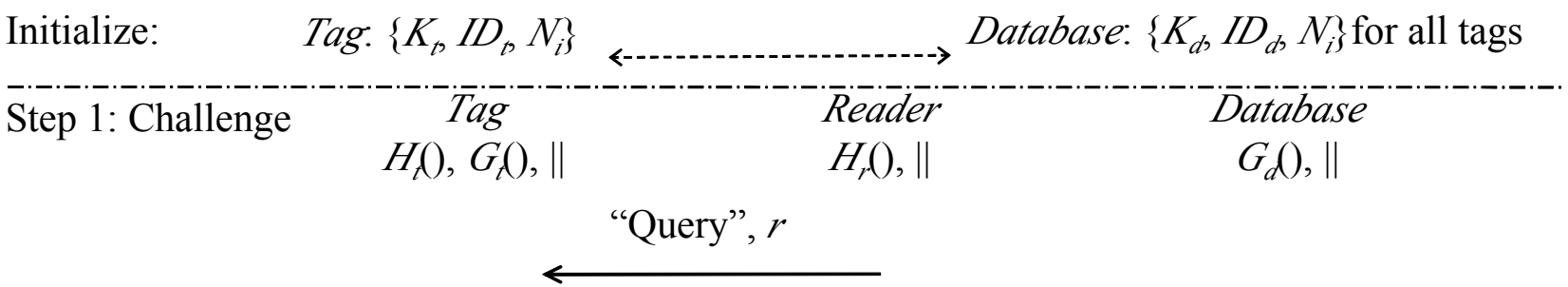
KEIO UNIVERSITY
1858
CALAMVS GLADIO FORTIOR

IV. The proposed protocol



T	Tag, or transponder	Kt	The secret key stored in the tag
R	Reader, or transceiver	Kd	The secret key stored in the database
D	Database	r	Random number generated
IDt	Identification value stored in the tag	||	Link operation
IDd	dentification value stored in the database		
Ni	The ith nickname, i=1,…,n. n is the number of nicknames stored in the tag		

IV. The proposed protocol



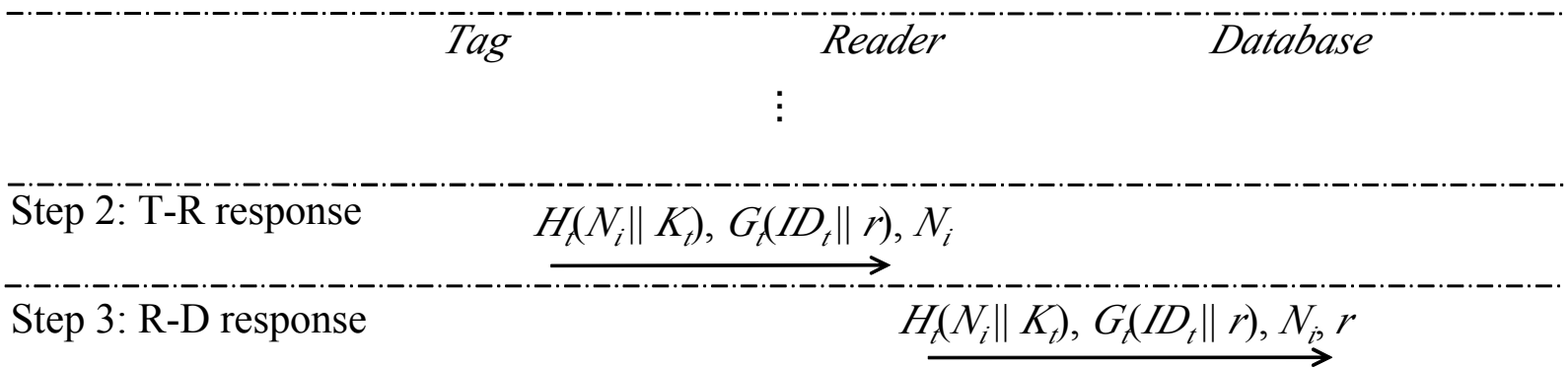
Initial setup:

Each tag stores its identifier, ID_t, secret key, K_t, and several nicknames, N_i. And shared within the back-end database.
Each tag has 2 hash functions, $H_t()$ and $G_t()$, and link operation. And the reader has a random number generator.

Step 1 (Challenge):

The reader generates a fresh random nonce, r, and sends it with query to the tag.

IV. The proposed protocol



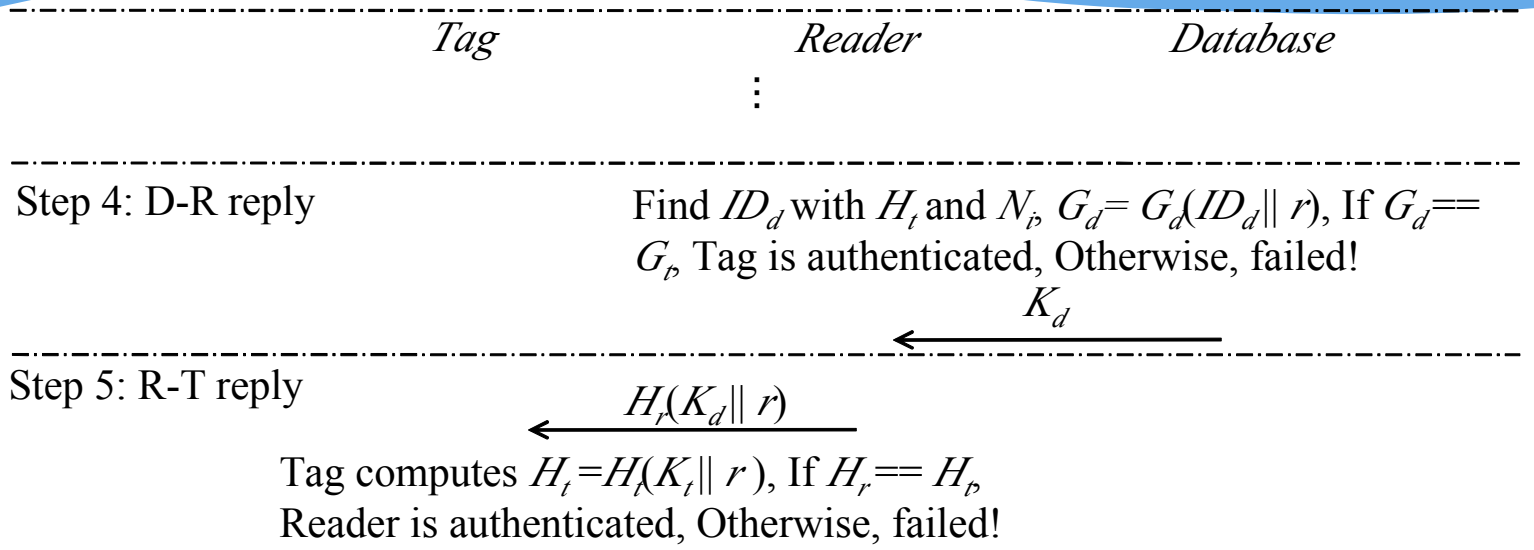
Step 2 (T-R response)

After being queried, the 2 hash values, H_t and G_t, are calculated and sent to the reader with the picked nickname in this step.

Step 3 (R-D response)

The received information H_t, G_t, N_i and r, is sent to the database to find the corresponding secret key, K_d, stored in the database.

IV. The proposed protocol

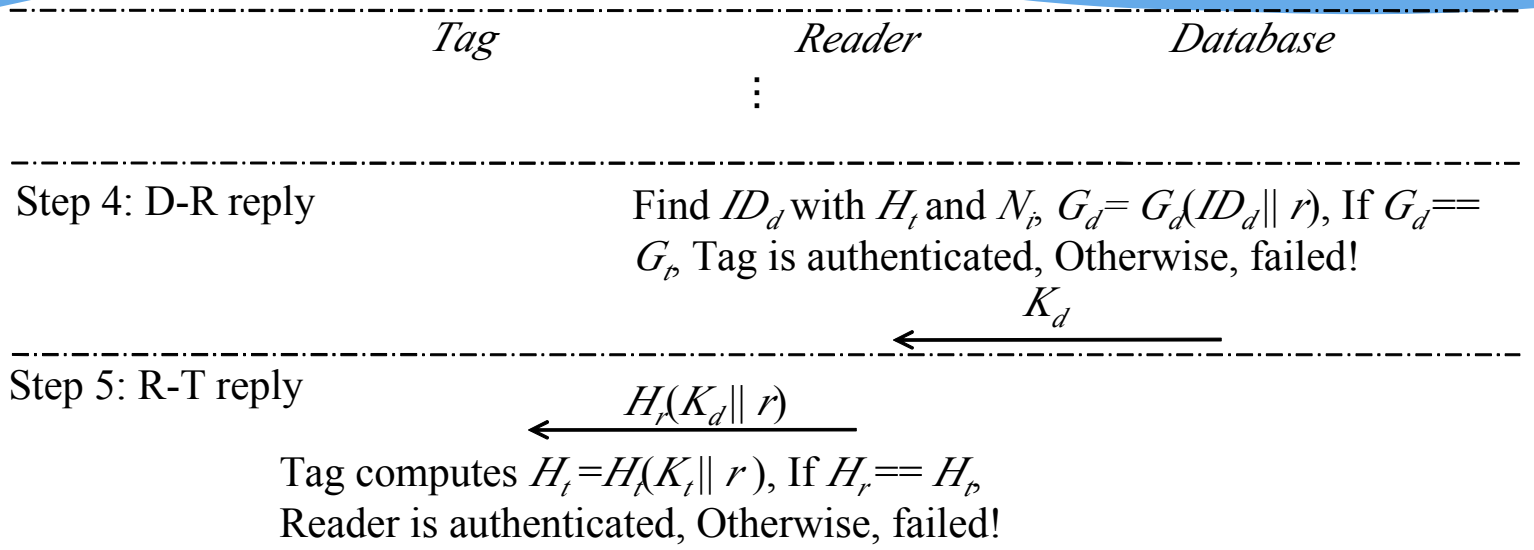


Step 4 (D-R reply)

The database will look for ID_d due to the received N_i and the initially-stored hash values. The database computes the hash value G_d and compares with the received G_t; if G_d equals G_t, the tag is authenticated by the reade.

If succeeds, K_d is sent to the reader.

IV. The proposed protocol



Step 5 (R-T reply)

After receiving K_d, the reader computes the hash value, H_r and then sends to the tag. On the tag side, it computes H_t and compares with H_r. If H_t equals H_r, the reader is successfully authenticated;
Otherwise the protocol fails to execute.

V. Analyses

Security Analyses

Protocol	HL	RHL	Gossamer	Proposed
Confidentiality	OK	OK	moderate	OK
Indistinguishablity	ng	ng	OK	OK
Forward security	ng	ng	OK	OK
Mutual authentication	ng	ng	OK	OK
Eavesdropping attack prevent	ng	ng	moderate	OK
Spoofing attack prevent	ng	ng	OK	OK
Replay attack prevent	ng	ng	OK	OK
Track attacking prevent	ng	OK	OK	OK

V. Analyses

- Performance analyses

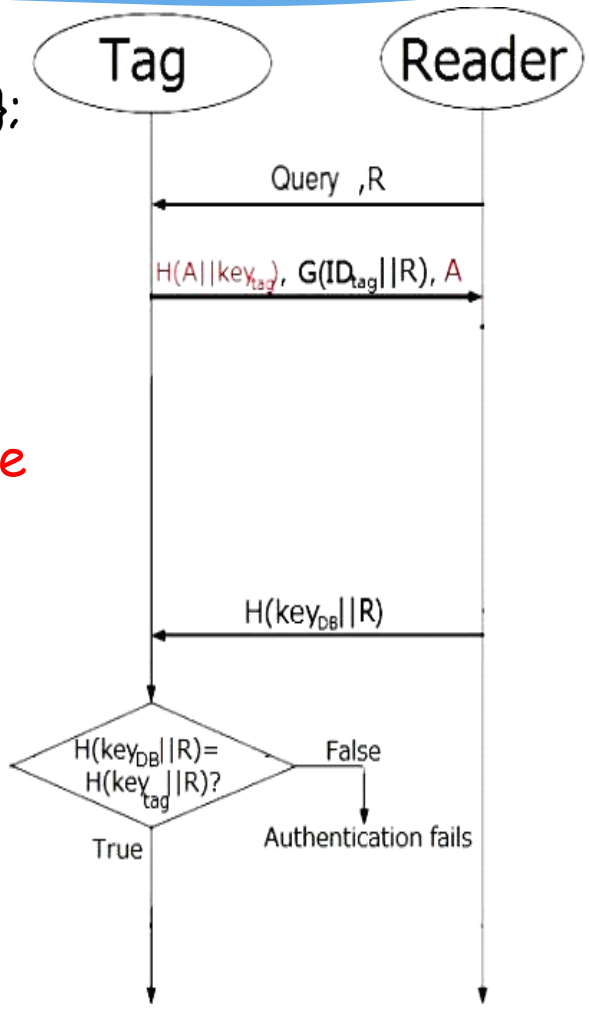
Protocol		HL	RHL	Gossamer	Proposed
No. of hash	Tag	1	2	–	3
	Reader	–	–	–	1
	DB	–	N	–	N
No. of RNG	Tag	–	1	3	1
	Reader	–	–	–	1
	DB	–	–	–	–
No. of ROT	Tag	–	–	8	–
	Reader	–	–	10	–
	DB	–	–	12	–
No. of auth steps		6	5	5	5
Required memory	Tag	$2L$	L	$7L$	$2L + nL$
	DB	$4LN$	LN	$4LN$	$(2L + nL)N$

VI. Simulation

- Pseudo-code in tag(1)

```
enum sysState ={ s_ini, s_check, s_pass, s_fail };

int AutoRun_Authentication ( )
{
        char * s, *Hr ,* r, *Ni;
        int i;
        sysState = s_ini;          // Initialize
        waitQuery(&r);
        i = Random(1,n);
        switch(i)
        {
                case 1: Ni = N1;break;
                case 2: Ni = N2;break;
                                ⋮
               case n: Ni = Nn;break;
               default: Ni = N1;
         }
```



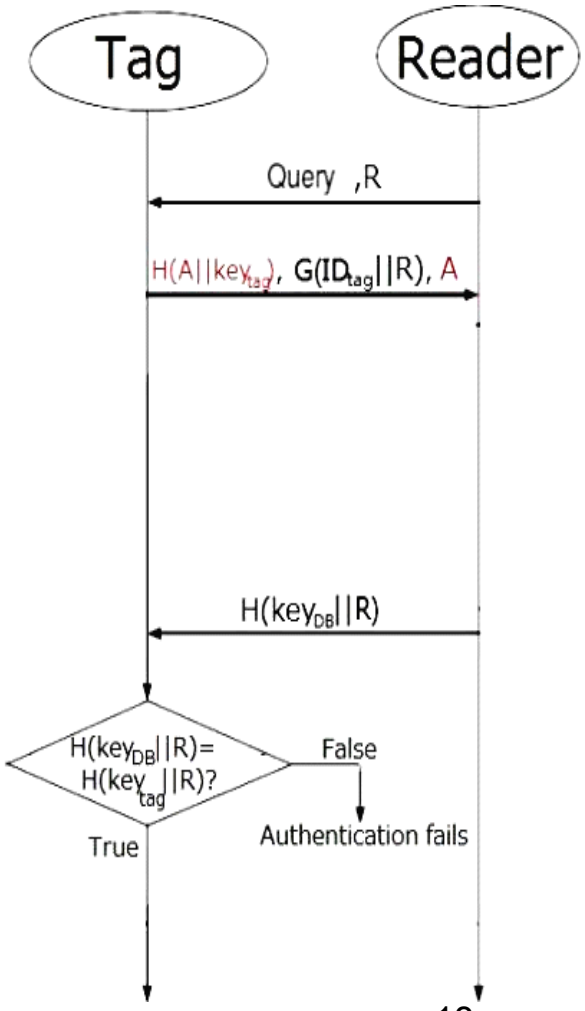
VI. Simulation

Pseudo-code in tag(1)

```
enum sysState ={ s_ini, s_check, s_pass, s_fail };

int AutoRun_Authentication ( )
{
        char * s, *Hr ,* r, *Ni;
        int i;
        sysState = s_ini;
        waitQuery(&r);
        i = Random(1,n);
        switch(i)
        {
                case 1: Ni = N1;break;
                case 2: Ni = N2;break;
                        ⋮
               case n: Ni = Nn;break;
               default: Ni = N1;
        }
```

// pick up nickname



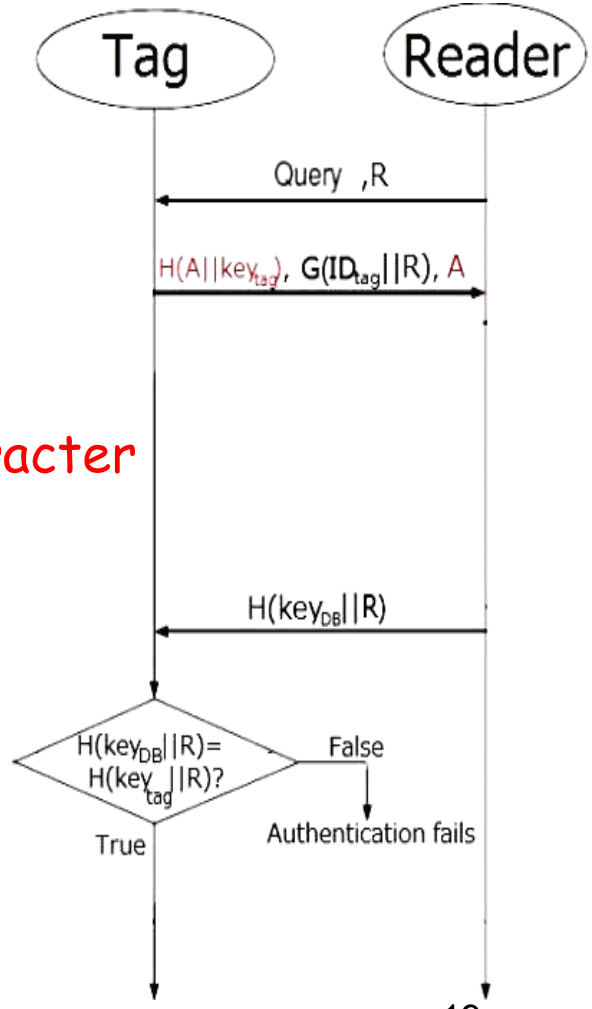
VI. Simulation

- Pseudo-code in tag(2)

```
s = connectchar( H( Ni || Kt) , G( IDt || r ) );
s = connectchar( s , Ni );
send(s);

sysState = s_check;
waitHashString( & Hr );
s = comparechar( Ht( Ni || Kt) , Hr );
if ( s == 0 ) sysState =s_pass
else sysState = s_fail;
}
```

// connect character & send



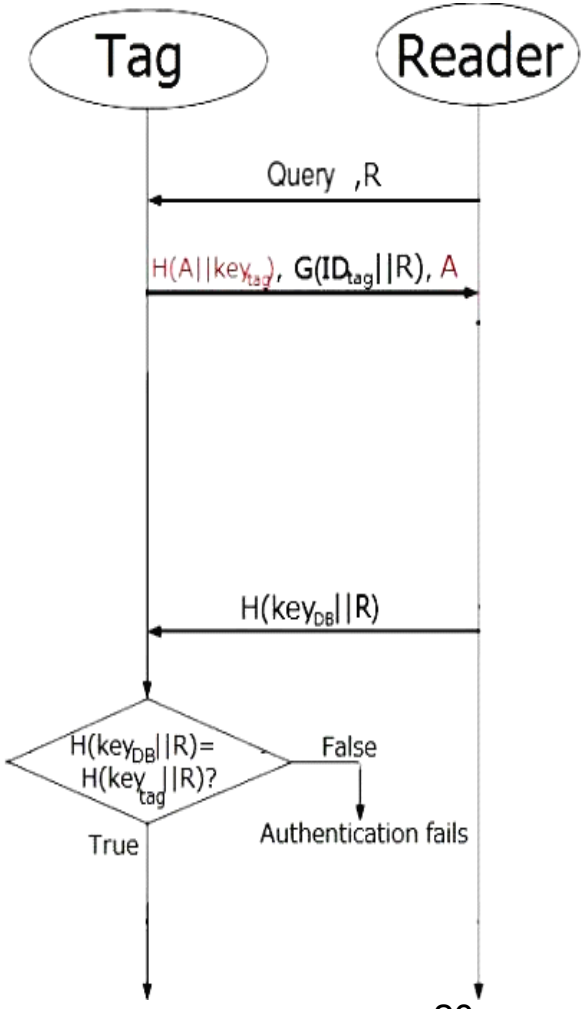
VI. Simulation

- Pseudo-code in tag(2)

```
    s = connectchar( H( Ni || Kt) , G( IDt || r ) );
    s = connectchar( s , Ni );
    send(s);

    sysState = s_check;
    waitHashString( & Hr );
    s = comparechar( Ht( Ni || Kt) , Hr );
    if ( s == 0 ) sysState =s_pass
    else sysState = s_fail;
}
```

// comparechar -> authenticate



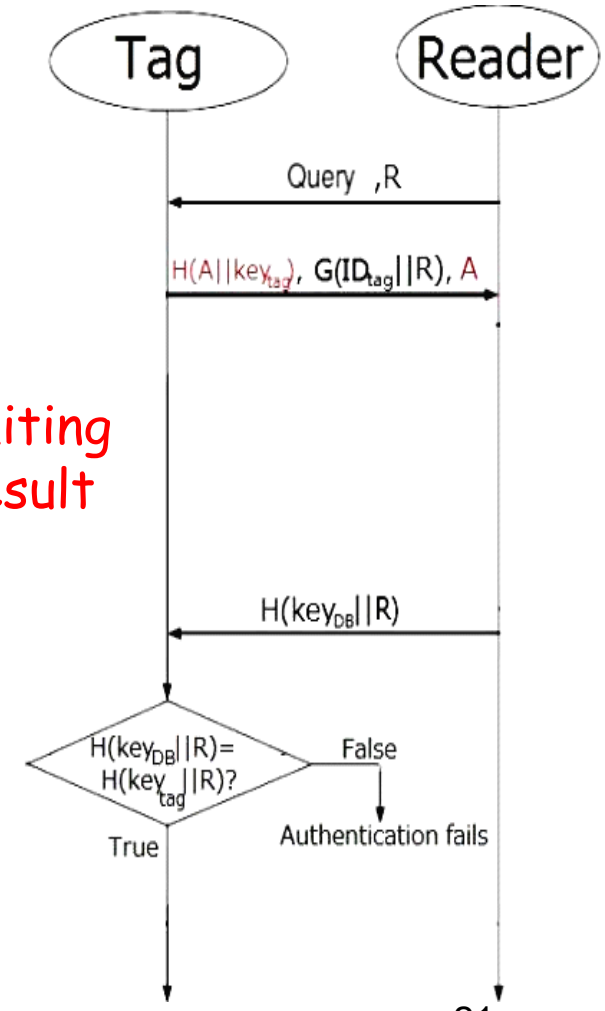
VI. Simulation

- Pseudo-code in reader

```
void Query( char* d_H_t ,d_ G_t , d_N_i , d_r )
{
        char* H_t, G_t, N_i; // buffer
        r = Random(1,n);
        send (r );
        waitResult( &H_t, &G_t, &N_i );
        d_H_t = H_t;
        d_ G_t = G_t;
        d_N_i = N_i;
        d_r = r;
}

Void AuthToCard( char* K_d )
{
        send (H_r(K_d || r));
}
```

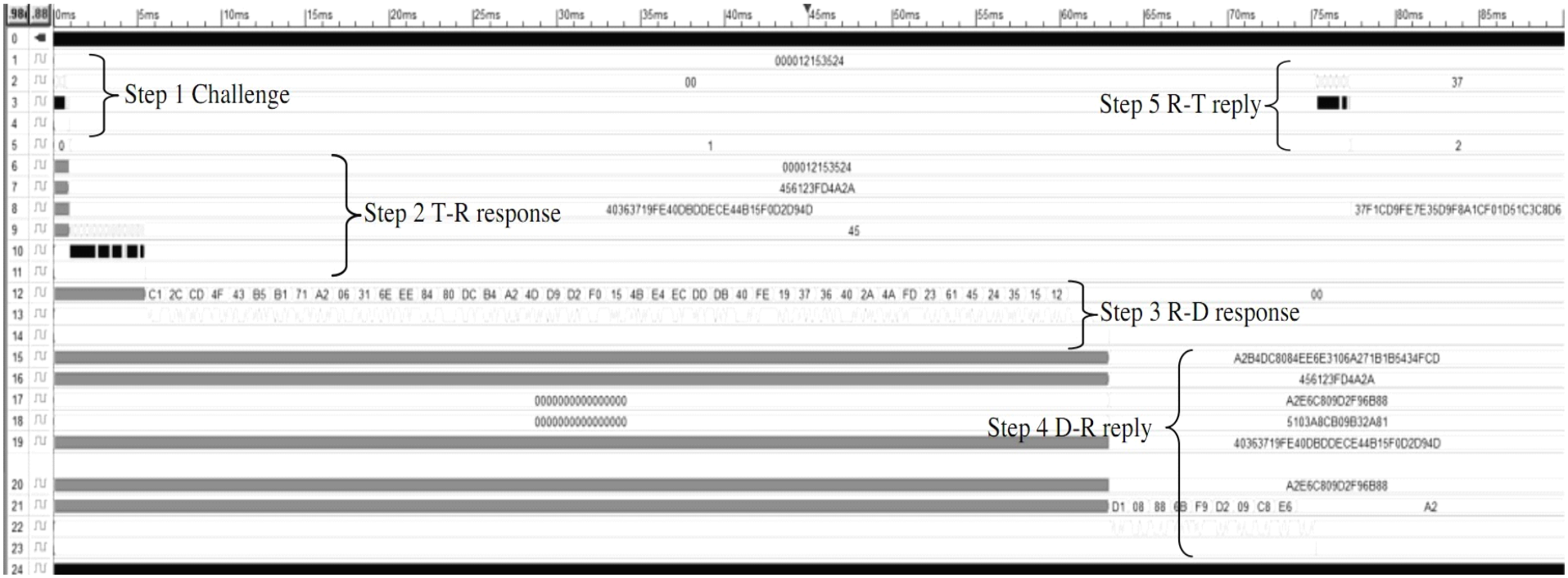
// waiting result



VI. Simulation

- Simulation environment
  - OS:Windows
  - Software：TestBencher Pro
  - Language:VERILOG
- Processing time
  - The proposed: 77.33ms
  - Gossamer: 120.4ms

Same of (rate of bottom layer, database, operation system)



Conclusions

- A new mutual authentication protocol based on the hash function and the nicknames is proposed and the efficiency of the proposal has been verified in the simulation.
- The security analyses and performance analyses show that the proposed protocol is secure against several types of attacks.
- The randomly-chosen nickname is utilized in authentication, during which the security level is assured due to the fuzziness of the picked nicknames and the usage of hash encryption.
- In conclusion, the proposed protocol has great potentials for low-cost RFID tags in the IoT system.

KEIO UNIVERSITY
1858
CALAMVS GLADIO FORTIOR

KEIO 150
Design the Future

Thank You !