High Speed, Low-Cost Key Encryption Function Implementation and Authentication Protocol Design for WSNs

Kavana A M[1] Kavya S G[2] Priyanka M K[3] Sindhu G N[4] Prashantha N C[5]
[1,2,3,4]Student [5]Assistant Professor
[1,2,3,4,5]Department of Electronics & Communication Engineering
[1,2,3,4,5]B.G.S. Institute of Technology Mandya, Karnataka

Abstract— Wireless Sensor Networking is one of the most exciting and challenging research domains, WSN's have been widely used in real-time traffic monitoring and military sensing and tracking, however, WSN applications could suffer from threats and hazard. Because of this the user authentication is an important concern to protect data access from unauthorized users. This paper presents a lightweight mutual authentication protocol for WSN applications. The data protection with a low computational cost, and the proposed key encryption function requires simple exclusive-or (X-OR) arithmetic operations. The advantage of this system also allow legitimate users can freely change own passwords and the data leakage can be eliminated.
Key words: WSN Applications, WSNs

I. INTRODUCTION

Wireless sensor networks (WSNs) play an important role in information transmission and have a wide range of applications such as real-time traffic monitoring, building safety monitoring, military sensing and tracking, etc. They are composed of many little and low-cost sensor nodes with limited energy and computation ability to cooperatively monitor physical environmental intimation. It is well known that WSNs are hugely censurable and become a threat, thereby endangering the applications if the accurate security is not taken into working out. Because of it, how to secure WSNs has been becoming a challenging issue as it presents a resource-constrained environment.

User authentication is one of the most determinant security mechanisms to prevent the illicit or malicious entities from accessing the WSNs. In the past 10-15 years, various authentication schemes in WSNs have been proposed. Benenson et al. presented an authentication protocol where users can successfully authenticate with any suborder of sensors out of a set of n sensors. Benenson et al. further utilized public key cryptography (PKC) and elliptic curve cryptography (ECC) to design a new authentication mechanism. It is established on the self-certified key cryptosystem, which is an innovation of ECC. The advanced user authentication scheme proposed by Butun was based on the fact that it employed both the PKC and symmetric key cryptography schemes. This approach provides higher energy efficiency. Where the PKC- or ECC-based scheme suffers from a high computational cost for WSNs.

To minify the computational cost, Wong et al. proposed a dynamic password-based authentication scheme. Although this scheme only requires one-way hash functions and simple XOR operations, it is censurable to many attacks such as replay (rematch) attacks, forgery (bogus) attacks, and so on. The "two-factor user authentication," can be used to avoid multiple users with the same login-id and stolen-verifier attacks. However, Das's scheme has no ability to resist gateway node bypass attack and privileged-insider attack. There is no provision of users to change or update their passwords. Consider a wireless heterogeneous sensor network that consists of two types of devices, for example, low-resource devices (TelosB, MICAz) and high-resource devices (Star gate), also known as the GW. The high-resource devices are tamper-resistant, but the low-resource devices are vulnerable to tampering. These devices are appropriately distributed in a confined area. A user can access (on demand) the sensor data within the network using their personal digital assistant, mobile phone, or laptop. To query the sensor data, a user must register with the GW node and get a smartcard. Upon registration, he/she can query the sensor data within the network in a secure manner. This scheme allows the user to choose and change his password frequently and An improved scheme presented is gives various advantages, including elimination of password leakage risk, capability of auto changeable password, and better efficiency and The use of the scheme in [9] is considered for the (24,12) Golay code, Which has an efficient parallel decoder capable of correcting the single and double-adjacent errors. The decoder exploits the properties of the Golay code to reduce the implementation cost. This results in a decoder that is simpler than a Personal use is permitted, but traditional SEC decoder but that can also correct all double-adjacent errors and some triple-adjacent errors.

In this paper, we present a mutual authentication mechanism for data protection. The remainder of this paper is organized as follows. The key generation function and algorithm are discussed in Part II. Part III presents the mutual authentication protocol for WSNs. The security and demonstration of the proposed scheme are discussed and analyzed in Part IV. Part V shows the simulation and verification results of proposed system.

II. PROPOSED KEY GENERATION SCHEME

- In proposed key generation scheme, a new key generation function (KeyG), which is a main component for data encryption (protection), is introduced.
- The password and data leakage complication can be reduced by exploiting the Key Generation (KeyG) function and XOR arithmetic operation for cover-coding.
- The Key Generation (KeyG) function and algorithm can be represented as follows.
- The 32-bit message (M_g) and password (P_w) in binary (base 2) can be represented as
  - Mg= m0, m1, m2... m31
  - P = p0, p1, p2... p31.
- The 32-bit random number R is represented by
  - $R_X = R_{MSB} \parallel R_{LSB}$
    Where,
    1) R_X=Random number

2) R_{MSB} and R_{LSB} are denoted as 16 most significant bits (MSBs) and 16 least significant bits (LSBs), respectively.
3) || denotes the bitwise concatenation operation.

Now, let R_{MSB} and R_{LSB} be in hexadecimal (base 16), respectively.

$$R_{MSB} = dt1\ dt2\ dt3\ dt4$$
$$R_{LSB} = ds1\ ds2\ ds3\ ds4$$

– Each digit of R_{MSB} and R_{LSB} is used to indicate a each bit location in Message(M_g) and these each bits will concatenates to form a 16-bit output in hexadecimal (base 16) can be represented as

$$M_g{-}KeyG\ (R_{MSB}, R_{SB}) = mdt1mdt2mdt3mdt4||mdt1+16mdt2+16mdt3+16mdt4+16||mds1mds2mds3mds4||mds1+16mds2+16mds3+16mds4+16= dv1dv2dv3dv4 \equiv R_{VX}$$

Where, dv1 dv2 dv3 dv4 is the hexadecimal (base 16) notation.

– $P_w{-}KeyG\ (R_{VX}, R_{MSB})$ denotes the Key Generation output. And it performed over Password (P_w) using the previously generated R_{VX} and R_{MSB}.
– It is indicating a bit location in Password (P_w) and each bits will concatenates to form a 16-bit Key.
– The resulting of these Key can be represented as
– $P_w{-}KeyG(R_{VX},R_{MSB}) = pdv1pdv2pdv3pdv4 || pdv1+16pdv2+16pdv3+16pdv4+16 || pdt1pdt2pdt3pdt4||pdt1+16pdt2+16pdt3+16pdt4+16=hw1hw2hw3hw4 \equiv Key$

Where,

– hw1 hw2 hw3 hw4 is the hexadecimal (base 16) notation.
– The key generation requires two steps. and it is represented by

$$Key \equiv P_w{-}M_g{-}KeyG\ (R_X).$$

A. *Encoding Procedure of the Message*

1) The 32-bit message (Mg) can be represented as

$$M_g = M_{MSB} \,||\, M_{LSB}$$

Where

M_{MSB} and M_{LSB} are 16 MSBs and 16 LSBs. By utilizing these Key, we perform the XOR operation for cover coding M_{MSB} and M_{LSB}, respectively. i.e.,

$$CCM_{gMSB} = M_{gMSB} \oplus P_w{-}M_g{-}KeyG\ (R_X)$$
$$CCM_{gLSB} = M_{gLSB} \oplus P_w{-}M_g{-}KeyG\ (R_X).$$

Consequently, the cover-coded message is

$$CCM_g = CCM_{gMSB} \,||\, CCM_{gLSB}.$$

For convenience sake, the steps are briefly expressed as

1) $CCM_g = M_g \oplus P_w{-}M_g{-}KeyG\ (R_X)$. Finally, the original message can be decoded and recovered through the XOR operation as follows:
2) $M_g = CCM_g \oplus P_w{-}M_g{-}KeyG\ (R)$.
3) The security is therefore significantly enhanced by this low complexity technique.

B. *Security Analysis of the Key Generation Function*

To investigate the possibility of the key decryption, we consider the scenarios as follows.

1) Scenario1: R_{MSB} is hacked and modified as 0000h (base16).
2) In this scenario, we have
3) $R_{VX} = M_g{-}KeyG\ (0000h, R_{LSB}) = m0m0m0m0 || m16m16m16m16 || mds1mds2mds3mds4 || mds1+16mds2+16mds3+16mds4+16.$

It is observed that $R_{VX} \in \{00XYh, 0FXYh, F0XYh, FFXYh\}$

Each case has an equal probability of 1/4.

1) Case 1: R_{VX} = 00XYh: The generated key is computed as

Key = $P_w{-}KeyG\ (00XYh, 0000h) = p0p0pXpY || p16p16pX+16\ pY+16 || p0p0p0p0 || p16p16p16p16.$

We consider the following conditions of X and Y.

1) When X =0and Y =0, the probability of key decryption is $1/16\times1/16\times1/2^2=1/2^{10}$. The probability of X =0(Y = 0) is 1/16.
2) When X = 0and Y ≠ 0(X ≠ 0and Y =0), the probability of key decryption is $1/16\times15/16\times 1/2^4 = 15/2^{12}$.
3) When X ≠ 0 and Y ≠ 0, the probability of key decryption is $15/16\times15/16\times1/2^6 = 225/2^{14}$.

– The probability that the key is decoded in case 1 is Pr (Case1) $=1/4\times (1/2^{10}+2\times15/2^{12}+ 225/2^{14}) = 0.0055$.

2) Case 2: R_{VX} =0FXYh

Key =$P_w{-}KeyG\ (0FXYh, 0000h) = p0p15pXpY || p16p31pX+16pY +16 || p0p0p0p0 || p16p16p16p16.$

– The conditions of X and Y are considered as follows.

1) When X =0andY =0(X = F and Y = F), the probability of key decryption is $1/16\times1/16\times1/2^4 =1/2^{12}$.
2) When X = 0and Y = F (X = F and Y =0), the probability of key decryption is $1/16\times1/16\times1/2^4 =1/2^{12}$.
3) When X =0 and Y ≠ 0, F (X = F and Y ≠ 0, F), the probability of key decryption is $1/16\times14/16\times1/2^6 =7/2^{13}$.
4) When X ≠ 0, F and Y =0(X ≠ 0, F and Y = F), the probability of key decryption is $1/16\times14/16\times1/2^6 =7/2^{13}$.
5) When X ≠ 0, F and Y ≠ F, the probability of key decryption is $14/16\times 14/16\times1/2^8 = 49/2^{14}$.
6) The probability of key decryption in case 2 is Pr (Case 2) $= 1/4\times (4\times1/2^{12}+4\times7/2^{13} + 49/2^{14}) = 0.0019$.

3) Case 3: R_{VX}= F0XYh

– Key = $P_w{-}KeyG\ (F0XYh, 0000h) = p15p0pXpY || p31p16pX+16pY +16\ p0p0p0p0 || p16p16p16p16.$
– In case 2, the probability of key decryption as Pr (Case 3) = 0.0019.

4) Case 4: R_{VX} = FFXYh

– Key = $P_w{-}KeyG\ (FFXYh, 0000h) = p15p15pXpY || p31p31pX+16pY +16\ p0p0p0p0 || p16p16p16p16.$
– Similarly, the probability of key decryption in case 4 is Pr (Case 4) = 0.0019.

According to cases 1 to4, we have the probability that the key is decoded in,

– Scenario 1 is 0.0055+ 3×0.0019 = 0.0112.
– Scenario 2: R_M and R_L are both hacked and changed as 0000h (base 16).
– $R_{VX} = M_g{-}KeyG\ (0000h, 0000h) = m0m0m0m0 ||m16m16m16m16||m0m0m0m0|| m16m16m16m16.$

It implies that $R_{VX} \in \{0000h, 0F0Fh, F0F0h, FFFFh\}$.

– Each case has an equal probability of 1/4.

Case 1: $R_{VX} = 0000h$.

– The generated key is computed as

Key = $P_w{-}KeyG\ (0000h, 0000h) = p0p0p0p0 || p16p16p16p16 || p0p0p0p0 || p16p16p16p16.$

– The probability of key decryption in case 1 is Pr (Case 1) $= 1/4\times1/2^2 =1/2^4$

Case 2: R_{VX}=0F0Fh.

- The generated key is computed as
  Key = P_w−KeyG (0F0Fh, 0000h) = p0p15p0p15 || p16p31p16p31 || p0p0p0p0 || p16p16p16p16.
- The probability of key decryption in case 2 is Pr (Case 2) = $1/4\times1/2^4 = 1/2^6$.

Case 3:- R_{VX}= F0F0h.

- The generated key is computed as
  Key = P_w−KeyG (F0F0h, 0000h) = p15 p0p15p0||p31p16p31p16||p0p0p0p0||p16p16p16p16.
- The probability of key decryption in case 3 is Pr (Case 3) = $1/4\times1/2^4=1/2^6$.

Case 4: R_{VX} = FFFFh.

- The generated key is computed as
  Key = P_w−KeyG (FFFFh, 0000h) = p15p15p15p15||p31p31p31p31||p0p0p0p0||p16p16p16p16.
- The probability of key decryption in case 4 is Pr (Case 4) = $1/4\times1/2^4 =1/2^6$.

The probability that the key is successfully decoded in scenario 2 is $1/2^4+3\times1/2^6$ =0 .1094.

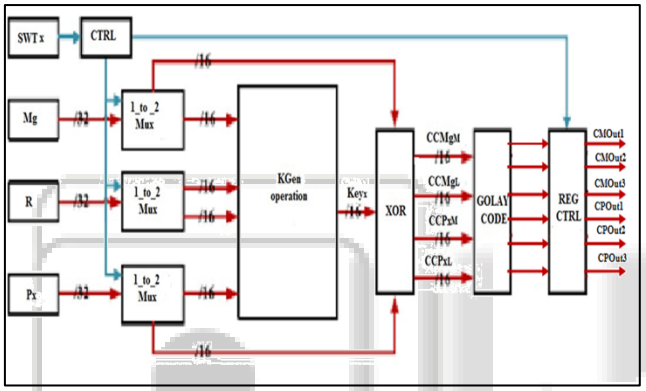

Fig. 1: Proposed key generation block diagram

III. PROPOSED MUTUAL AUTHENTICATION PROTOCOL

The scheme can be divided into 4 phases they are: registration phase, login phase, Authentication phase, password change phase.

A. *Registration Phase*

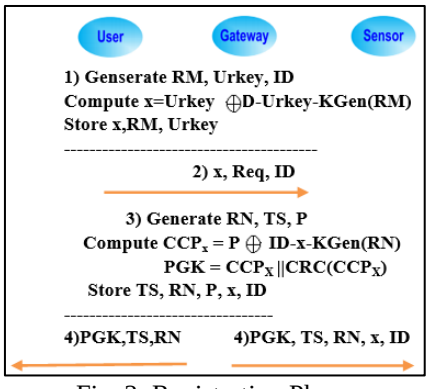

Fig. 2: Registration Phase

- In this phase generates user's mobile device, user ID, and select the client password (urkey) and then computes x by the KeyG function and Xor operation.
- Users sends x, ID, and the requests to the gateway node.
- Gateway node can produce random number RN, password ID, time stamp TS, and access password Pw. The gateway using ID, x, and the KeyG algorithm and cover code passwords CCPwx. Finally, this technique generated by using cyclic redundancy code CRC.
- User can receive PGK, TS, RN from gateway node. The sensor node can decode with P to store the access password information.

B. *Login Phase*

- Login phase submits the ID, random number RN, and access password PW to generate PUK by using crc technique.
- Users transmit PUK, and current time t1 to the gateway node.
- Gateway node use the current time t2 and checks t2-t1<ΔT. The time interval within the range, gateway node generates PGK and If PGK=PUK and x*=x. Finally, PGK and t1 are stored.
- The sensor node can record the time t4 and checkst4-t3<ΔT. The time interval within the range. If PSK=PGK and x* = x. Finally, PSK and t1 are stored.

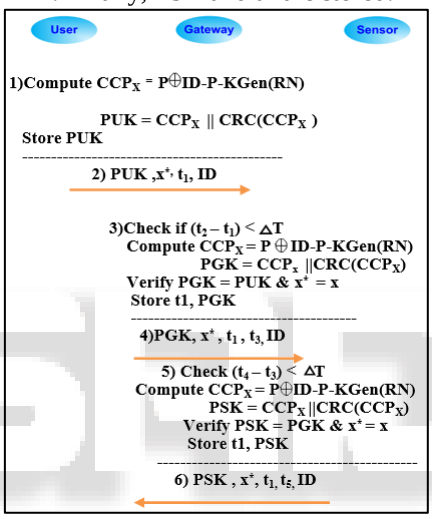

Fig. 3: Registration Phase

C. *Authentication Phase*

- The gateway node records the current time $t6$, checks if $t6 - t5 < \Delta T$, and examines if the time $t1$ had been stored. If either of the aforementioned conditions is violated, the authentication phase ends. Otherwise, the gateway further examines if $PGK = PSK$ and $x = x*$.
- The gateway node transmits $PGK*$, $x*$, $t1$, and the current time $t7$ to the sensor node.
- The sensor node records the current time $t8$, checks if $t8 - t7 < \Delta T$, and examines if the time $t1$ had been stored. Similarly, if either of the aforementioned conditions is not satisfied, the authentication phase ends.
- The sensor node transmits $PSK*$, $x*$, $t1$, and the current time $t9$ to the user.
- The user records the current time $t10$, checks if $t10 - t9 < \Delta T$, and examines if the time $t1$ had been stored Gate node receives ID, PSK, t1 and t5 from sensor node..

D. *Password Change Phase*

In the proposed mechanism, the function of password change is presented in and operates as follows.

- The user generates a new random number RMr, UrKeyr, and new access password PWrand then computes CCRMxr, PWUK, and xr. p_{wr}, xr, RMr, and PWUK need to be stored.

- The user transmits CCRMxr, PWUK, RMr, and ID to the gateway node.
- The gateway node obtains Rmrby the XOR operation of CCRMxr⊕ ID. Furthermore, Pwr can be recovered via CCP Wxr. Finally, P_{wr}, xr, and Rmrhave been updated.
- The sensor node receives ID, P W ttKr, RM r, and xr from the gateway node and then decodes P W ttKrand\ stores PWr. The password changes phase end.

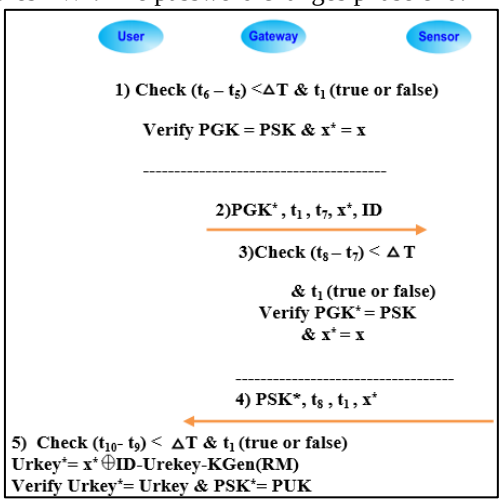

Fig. 4: Authentication phase

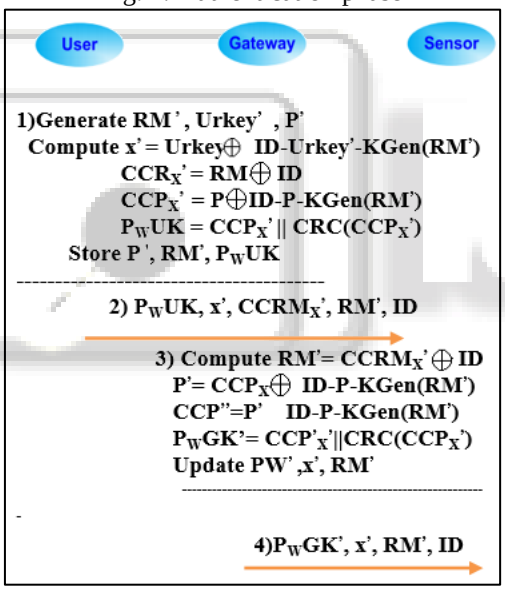

Fig. 5: Password change phase

IV. Analysis of the Proposed Mutual Authentication Protocol

In this section, we analyze the security of the proposed mutual authentication mechanism.

A. *Replay Attack*

If the intruder gets ID, x∗, and PUK to login the gateway node, the verification will fail because (t_1− t1) < ΔT is checked. Note that t_1 is the system time of the gateway node when receiving the replayed message. The proposed framework is therefore secure against the replay attack. In a replay attack, Adversary may collect messages over a public network and try to replay them to the GW node, Sn, and user

B. *Impersonation Attack*

An adversary cannot impersonate the user when he captures the message ID, x∗, PU, and timestamp t1 in the login phase. It is infeasible for the attacker to get the original UrKey and PW because they are protected by the key encryption function. Hence, the proposed scheme is strong against user impersonation attacks and message alteration attacks.

C. *Node Compromise Attack*

The impact of the node compromise attack is high when the user is allowed for data access and authentication directly. In the proposed mechanism, the authentication procedure works via the gateway and sensor nodes. It may alleviate the impact of such an attack. However, there is still room for future improvement.

D. *Stolen-Verifier Attack*

When x∗ and PW stored in the gateway node are stolen, the insider still cannot login to the system. It is because the gateway node needs to identify if PGK = PUK, where PGK is computed by the key encryption function and PW. The proposed scheme is strong against the stolen-verifier since the GW does not maintain a password/verifier table.

E. *Guessing Attack*

In the presented mechanism, the passwords Urkey and PW are encrypted in the login phase, and it is hard to guess the two passwords in the limited time interval ΔT.

F. *Insider Attack*

In the registration phase, the user and the gateway node only send the cover coded UrKey and PW, where the proposed key encryption functions are unavailable to the attacker. This makes the insider attack impossible. However, in the registration phase, a user sends his/her password as a hashed value (h(b⊕PWk)) rather than a plain password. An insider cannot see the user's password.

V. The Simulation and Results of Proposed System

Simulations of the proposed system is done by using Xlink 14.2 software tool and simulation result of the key generation function are shown in Figure 6.

Assume that Msg = 000100100100000110011001001000011$_b$ and Pwd= 0001 0010001101000101010101010101$_b$. The random number is generated as Rx = 10101011100110101000011001000001$_b$. In Fig. 6, the key is produced as 0001000011010101$_b$ based on the KGen algorithm. They execute the XOR operations with Msg_M and Mg_L so as to generate $CCMsg_M$ = 0000001010010110 0$_b$ and $CCMsg_L$ =0001000101001011 0$_b$, respectively. Finally, the Golay (24, 12) codes are computed for CCMsg1, CCMsg2 and CCMsg3. When swt_1 turns high, the outputs are CCMsg1||Golay (CCMsg1), CCMsg2||Golay (CCMsg2) and CCMsg3||Golay(CCMsg3). Note that swt_1 is the control signal. Therefore, CMOut1 = 010010011001011001101101$_b$, CMOut2 = 110001100010110001011101$_b$ and CMOut3 = 010010101000000010011101$_b$. As shown in Fig. 6, Pwd_M and Pwd_L perform XOR operations to generate CCPwd1, CCPwd2 and CCPwd3. The Golay (24,12) codes for CCPwd1, CCPwd2 and CCPwd3 are then computed. When swt_1 turns high, the outputs are CCPwd1||Golay(CCPwd1), CCPwd2||Golay(CCPwd2) and CCPwd3||Golay(CCPwd3)

Accordingly, CPOut1 =$001000111100000000010101_b$, CPOut2 = $111011010000001101100101_b$ and CPOut3 = $010001010100000001001101_b$.

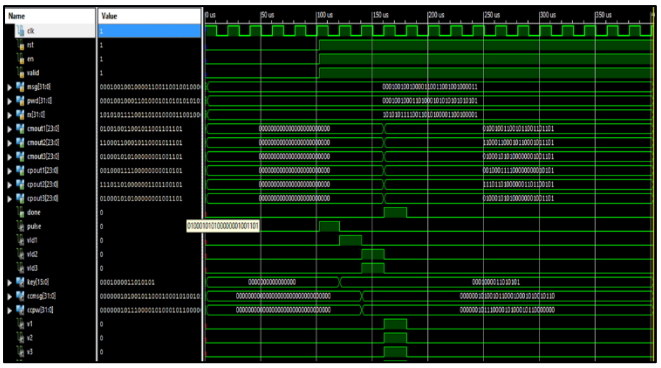

Fig. 6: Simulation result of Key generation function.

VI. CONCLUSION

A lightweight mutual authentication protocol over WSNs has been developed in this paper. In comparison with other recent schemes, the proposed approach provides a low computational cost with high speed, while the secure data transmission is given. The feasibility that the presented mutual authentication protocol can be used in WSN systems is successfully demonstrated.

VII. FUTURE SCOPE

We can implement the advanced version of this presented proposed scheme by adding the Turbo code, which is used in the decoding design of an advanced system and it produces less delay and higher frequency, which gives higher efficiency, and it has excellent error correcting capability.

REFERENCES

[1] Z. Benenson, F. Gartner, and D. Kesdogan, "User authentication in sensor networks," in Proc. Workshop Sensor Netw., Lecture Notes Informat. Proc. Informatik, 2004, pp. 1–5.
[2] Z. Benenson, N. Gedicke, and O. Raivio, "Realizing robust user authentication in sensor networks," in Proc. Workshop REALWSN, 2005, pp. 20–21.
[3] I. Butun, "Advanced two tier user authentication scheme for heterogeneous wireless sensor networks," in Proc. 2011 IEEE CCNC, Jan. 2011, pp. 169–171.
[4] K. H. M.Wong, Y. Zheng, J. Cao, and S. Wang, "A dynamic user authentication scheme for wireless sensor networks," in Proc. IEEE Int. Conf. SUTC, vol. 1, Jun. 2006, pp. 244–251.
[5] H. R. Tseng, R. H. Jan, and W. Yang, "An improved dynamic user authentication scheme for wireless sensor networks," in Proc. IEEE GLOBECOM, Nov. 2007, pp. 986–990.
[6] M. L. Das, "Two-factor user authentication in wireless sensor networks," IEEE Trans. Wireless Commun., vol. 8, no. 3, pp. 1086–1090, Mar. 2009.
[7] P. Kumar, M. Sain, and H. J. Lee, "An efficient two-factor user authentication framework for wireless sensor networks," in Proc. 2011 ICACT, Feb. 2011, pp. 574–578.
[8] F.Wang, Y. Zhang, Y. Xu, L.Wu, and B. Diao, "A DoS-resilient enhanced two-factor user authentication scheme in wireless sensor networks," in Proc. ICNC, Feb. 2014, pp. 1096–1102.
[9] K. Namba, S. Pontarelli, M. Ottavi, and F. Lombardi, "A single-bit and double-adjacent error correcting parallel decoder for multiple-bit error correcting BCH codes," IEEE Trans. Device Mater. Rel., vol. 14, no. 2, pp. 664–671, Jun. 2014.

