International Journal of Mobile Network Communications & Telematics (IJMNCT) Vol. 4, No.3, June 2014

OPTIMIZED RATIONALIZE SECURITY AND EFFICIENT DATA GATHERING IN WIRELESS SENSOR NETWORKS

S.Karthik[1], E. Lalitha[2]

[1]PG Student/M. Tech, KSR College of Engineering,
[2]PG Student/M. Tech, KSR College of Engineering,

ABSTRACT

Wireless reprogramming during a wireless detector net- work (WSN) is that the method of propagating a replacement code image or relevant commands to detector nodes. As a WSN is sometimes deployed in hostile environments, secure reprogramming is and can continue to be a significant concern. Whereas all existing insecure/secure reprogramming protocols square measure supported the centralized approach, it\'s necessary to support distributed reprogramming during which multiple licensed network users will at the same time and directly reprogram detector nodes while not involving the bottom station. Terribly recently, a novel secure rationalize and distributed reprogramming protocol named SRDRP has been planned, that is that the initial work of its kind. However, during this paper, we have a tendency to establish associate inherent style weakness within the increased signature verification of SRDRP associated demonstrate that it\'s at risk of associate impersonation attack by that an resister will simply impersonate any licensed user to hold out reprogramming. Later on, we have a tendency to propose a straightforward modification to mend the known security drawback while not losing any options of SRDRP. Our experimental results demonstrate that it\'s able to eliminate the planning weakness by adding one-B redundant information which the execution time of the prompt answer during a 1.6-GHz laptop personal computer is not any quite one ms. Therefore, our answer is possible and secure for real-world applications. Moreover, we have a tendency to show that, so as to additional improve the safety and potency of SRDRP; any higher established identity-based position formula will be directly utilized in SRDRP. Supported implementation results, we have a tendency to demonstrate potency improvement over the initial SRDRP.

KEYWORDS

Security Reprogramming, security, sensor networks, system verification, random delays, rationalize.

1. INTRODUCTION

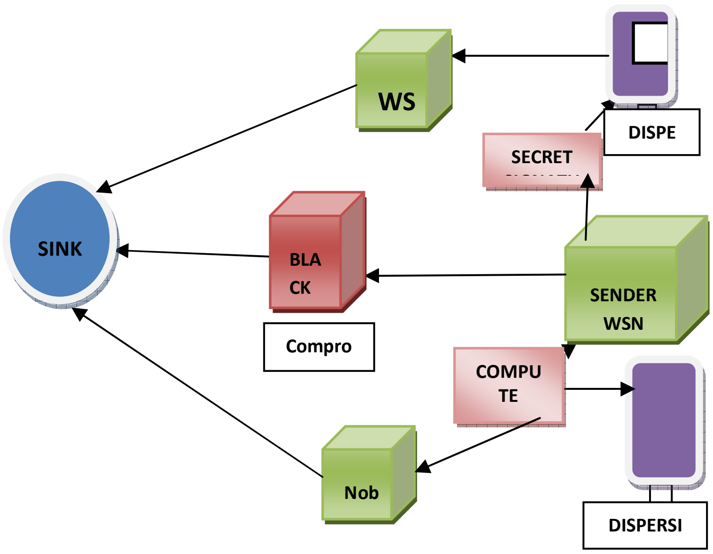
WIRELESS device reprogramming is that the method of propagating a new code image or relevant commands to device nodes through wireless links when a wireless device network (WSN) is deployed. Owing to the necessity of removing bugs and adding new functionalities, reprogramming is Associate in nursing necessary operation operate of WSNs. As a WSN is sometimes deployed in hostile environments like the battleground, Associate in nursing person might exploit the reprogramming mechanism to launch varied attacks. Thus, secure programming is and can still be a serious concern. There has been lots of analysis specializing in secure reprogramming, and lots of attention-grabbing protocols are projected in recent years. However, all of them or supported the centralized approach that assumes the existence of a base station, and solely the bottom station has the authority to reprogram device nodes, as shown within the higher figure in Fig. 1 sadly, the centralized approach isn't reliable as a result of, once the bottom station

DOI : 10.5121/ijmnct.2014.4305

fails or once some device nodes lose connections to the bottom station, it's not possible to hold out reprogramming. Moreover, there WSNs having no base station in any respect, and hence, the centralized approach isn't applicable. Also, the centralized approach is inefficient, decrepit scalable, and at risk of some potential attacks on the long communication path.

Alternatively, as shown within the lower figure in Fig. 1, a distributed approach is utilized for reprogramming in WSNs. It permits multiple licensed network users to simultaneously and directly update code pictures on totally different nodes while not involving the bottom station. Another advantage of distributed reprogramming is that {different totally different completely different} licensed users could also be appointed different privileges of reprogramming sensor nodes. this is often notably necessary in large-scale WSNs in hand by Associate in Nursing owner and employed by totally different users from each public and personal sectors. Also, SRDRP has been implemented in a very network of resource-limited device nodes to point out its high potency in follow.

1.1 SYSTEM ARCHITECTURE



However, during this paper, we tend to demonstrate that a style weakness exists in the user pre-processing section of SRDRP, associate degreed an antagonist will simply impersonate any approved user to hold out reprogramming. Let P be a generator of G. Let $\hat{e} : G \times G \rightarrow GT$ be a additive map. To eliminate the known security vulnerability, we propose

- Decide random $s \in Z*$as the passe-partout, and figure a straightforward modification on SRDRP while not losing any feature public key $P\ Kowner= s \bullet P$.(such as distributed reprogramming, supporting completely different user privileges, dynamic participation, quantifiability, high potency, and sturdy security) of the first protocol.
- Select 2 secure science hash functions H1 and H2 , where

$$H1: * \rightarrow G \quad \text{and } H2 : * \rightarrow Z* .$$

Then, the public parameters, any economical identity-based signature algorithmic program that has survived a few years of public scrutiny are often directly utilized in SRDRP. This paper additionally reports the experimental results of the improved SRDRP in laptop computer PCs and resource- restricted sensing element nodes, that show its potency in apply. The remainder of this paper is organized as follows. we tend to shortly review SRDRP in Section II so determine its security weakness in Section III. Section IV presents a modification that remedies the known weakness. Section V provides associate degree approach to additional improve the protection and potency of SRDRP. Section VI concludes this paper and points out future analysis directions.

2. TRANSIENT OVERVIEW OF SRDRP

The SRDRP consists of 3 phases: system data format, user preprocessing, and sensing element node verification. Within the system data format section, the network owner creates its public non-public and personal} keys so assigns the reprogramming privilege and also the corresponding private key to the approved user(s). Solely the general public parameters ar loaded on every sensing element node before preparation. Within the user preprocessing section, if a network user enters the WSN and incorporates a new code image, it'll have to be compelled to construct the reprogramming packets so send them to the sensing element nodes. Within the sensing element node verification section, if the packet verification passes, then the nodes settle for the code image.

2.1 System data format section

The network owner executes the subsequent steps.

1. Let G be a cyclic additive cluster and GT be a cyclic increasing cluster of an equivalent primer order alphabetic character. sensing element node before preparation.
2. For a user U_j with identity $UID_j \in *$, the network owner sets U_j's public key as $PK_j = H1 (UID_j \; Pr_{ij}) \in G$, computes the personal key $SK_j = s \bullet PK_j$, so sends back to U_j employing a secure channel, like the wired transport layer security protocol. Here, Pr_{ij} denotes the amount of user privilege (e.g., the sensing element node set among a particular region that user U_j is allowed to reprogram) and subscription amount.

2.2 User Pre-processing section

User U_j takes the subsequent actions. U_j partitions the code image to Y fixed-size pages, denoted as page one through page Y. U_j splits page i ($1 \leq i \leq Y$) into N fixed-size packets, denoted as $Pkt_{i,1}$ through $Pkt_{i,N}$. The hash worth of every packet in page Y is appended to the corresponding packet in page $Y -$ one. for instance, the hash worth of packet $Pkt_{Y,1}$ $h(Pkt_{Y,1})$ is enclosed in packet $Pkt_{Y-1,1}$. Here, $Pkt_{Y,1}$ presents the primary packet of page Y. Similarly, the hash worth of every packet in page $Y -$ one is enclosed within the corresponding packet in page $Y -$ two. This method continues till U_j finishes hashing all the packets in page two and together with their hash values within the corresponding packets in page one. Then, a Merkle hash tree is employed to facilitate the authentication of the hash values of the packets in page one. We tend to discuss with the packets associated with this Merkle hash tree conjointly as page zero. The foundation of the Merkle hash tree, the information regarding the code image (e.g., version variety, targeted node identity set, and code image size), and a signature over all of them are enclosed in an exceedingly signature message. Assume that the message m represents the foundation of the Merkle hash tree and also the information regarding the code u of $H2$ (m) modulo alphabetic character victimization the extended geometrician algorithm.

2.3 sensing element Node Verification section

Upon receiving a signature message, every sensing element node verifies it as follows.

1. The sensing element node initial pays attention to the lawfulness of the programming privilege P_{rij} and also the message m. given that they're valid, the verification procedure goes to successive step.
2. Given the general public parameters, the sensing element node performs the subsequent verification:
   $\hat{e}(\sigma_j, P) = \hat{e}(H_2(m) \bullet H_1(UID_j \| P_{rij}), P_{Kowner})$.
3. If the aforesaid verification passes, the sensing element node believes that the message m and also the privilege P_{rij} ar from a licensed user with identity UID_j.

Hence, the sensing element node accepts the foundation of the Merkle hash tree made for page zero. Thus, the nodes will manifest the hash packets in page zero once they receive such packets, supported the protection of the Merkle hash tree. The hash packets embody the hash values of the information packets in page one. Therefore, once collateral the hash packets, a node will simply verify the information packets in page one supported the unidirectional property of hash functions. Likewise, once the information packets in page I actually have been verified, a sensing element node will simply manifest the information packets in page i + one, wherever i = one, 2,... ,$Y - 1$. given that all verification procedures represented antecedent pass, the sensing element node accepts the code image.

3. SECURITY WEAKNESS O F SRDRP

Recall that, within the user preprocessing section of SRDRP, the signature σ_i is computed as $H_2(m) \bullet SK_j$. This is, however, a style weakness as a result of it allows associate degree antagonist A to get the personal key SK_j of user U_j as shown within the following. whereas U_j transmits to the targeted nodes the signature message , the antagonist A will get by eavesdropping. With the public parameter H_2, the antagonist A will figure $\upsilon = (H_2(m))^{-1} \pmod q$, wherever $H_2(m) \in Z^*$ .Note that the aforesaid equation involves modular mathematical process with a negative exponent, which might be performed by finding the standard increasing inverse only if that number is comparatively prime to alphabetic character. That is, the reciprocal u of $H_2(m)$ modulo alphabetic character exists if and given that $H_2(m)$ and alphabetic character ar coprime (i.e., gcd($H_2(m)$, q)

1. In SDRP, since alphabetic character is prime, all of the nonzero integers in Z^* are comparatively prime to alphabetic character, and so, there exists a reciprocal for all of the nonzero integers in Z^* .
2. The antagonist A computes the personal key $SK_j = \upsilon \bullet \sigma_i$.

Consequently, the antagonist A will impersonate user U_j to inject fake code pictures to require over the management of the entire WSN. Of course, the injury that the adversary A will build is in keeping with the reprogramming privilege of user U_j .

4. SECURITY IMPROVEMENT OF SRDRP

Clearly, if $H_2(m)$ and alphabetic character don't seem to be coprime, associate degree antagonist cannot figure the personal key SK_j . Therefore, the design weakness of the user preprocessing section doesn't exist, and the ensuing attack is invalid. to realize this goal, the subsequent step is usually recommended to be superimposed into SDRP.

In the system initiation section, the order alphabetic character of cyclic additive cluster G and cyclic increasing cluster GT ought to be set to an oversized number. Note that Boneh et al. have introduced composite-order additive teams [24], that are wont to with success solve several difficult issues in cryptography. within the user preprocessing section, once user Uj computes m, it will check whether or not H2 (m) and alphabetic character ar coprime. If yes, before a signature on m is computed, redundant bits ar appended into m specified H2 (m) and alphabetic character don't seem to be coprime; otherwise, as represented in Section II-B, user Uj directly computes a signature on m. On the opposite hand, the sensing element node verification section remains an equivalent. That is, compared to the first SDRP, the advised modification doesn't incur any overhead on the sensing element node aspect. In the style of SDRP, the length of m is twenty nine B. additionally assume that the hash perform H2 is enforced victimization SHA-1 with a 20-B output. Taking alphabetic character as a 160-b random number.

we carry out experiments of coprime checking on laptop computer PCs with completely different procedure powers. In every experiment, alphabetic character is willy-nilly generated for a thousand times. For each q, m is willy-nilly generated for a thousand times. Thus, every experiment has one mil- lion measurements. The experimental results show that, while not the addition of any redundant bit, the chance that H2 (m) and alphabetic character don't seem to be coprime is fifty eight.0212%. Also, our implementation results regarding the common search time of acceptable redundant information and also the failure rate with the addition of 1 or 2 redundant bytes ar summarized in Table I. Here, we tend to take into account a 1.6-GHz processor and also the addition of 1 redundant computer memory unit as associate degree example. The failure rate for looking acceptable redundant information is zero.4597% for this experiment (i.e., the chance that H2 (m) and alphabetic character don't seem to be coprime is one − zero.4597% = 99.5403%), and also the search of acceptable redundant information is extremely quick (i.e., the common execution time is sixty eight.12 μs). Clearly, failure rates depend upon the bit length of the superimposed redundant information however not on processor speed.

Table 1

Failure rates and search time f or the addition of one or 2 redundant bytes once alphabetic character is a composite variety.

	Experiment 1		Experiment 2		Experiment 3		Experiment 4		Experiment 5	
processor speed (GHz)	1.6		2		2.4		2.6		3.1	
redundancy (byte)	1	2	1	2	1	2	1	2	1	2
search time (μs)	68.12	608.91	50.82	598.01	49.34	529.85	48.18	368.77	40.02	300.65
failure rate (%)	0.4597	0.0691	0.7380	0.1851	0.7550	0.1780	0.9752	0.1566	0.9916	0.2772

Furthermore, taking alphabetic character as a 160-b random even variety, we tend to repeat the aforesaid experiments of coprime checking. The experimental results show that, while not the addition of any redundant bit, the chance that H2 (m) and alphabetic character don't seem to be coprime is fifty nine.4491%. Also, with the addition of 1 or 2 redundant bytes, the failure rates for looking acceptable redundant information ar all zero for every experiment (i.e., the chance that H2 (m) and alphabetic character don't seem to be coprime is 100%). On the opposite hand, our results regarding the common search time of acceptable redundant information of one or two B ar summarized in Table II. It are often seen that the search of acceptable redundant information is extremely quick. for instance, with the addition of 1 redundant computer memory unit, the common execution times ar forty.38 and 36.50 μs on one.6- and 1.8-GHz laptop computer PCs, severally. Here, it's advised to solely use one redundant computer memory unit once alphabetic character could be a 160-b random even variety. With this setting, not solely zero failure rate is achieved however additionally several benefits within the user pre-processing procedure are obtained in terms of computation, memory usage, and transmission and reception powers. As shown in Tables I and II, our experimental results demon- strate that the search time for 2-B redundant information isn't any over one ms in an exceedingly one.6-GHz notebook computer.

Considering that the clock frequencies of typical mobile devices (e.g., iPhones or laptop computer PCs) are over one GHz, the suggested modification is efficient for many of mobile devices. in step with the aforementioned analysis, the advised resolution is possible and secure for real-world applications.

5. ANY IMPROVEMENT OF SRDRP

Designing a secure rationalize reprogramming protocol may be a tough task, as a result of there square measure such a lot of details concerned (e.g., the complicated interactions with the environment) that the designer will solely strive his/her best to create certain his/her protocol is inerrant. this is notwithstanding whether or not security proofs square measure supported by heuristic arguments or formal ways in which. In reality, the degree of confidence incidental a security mechanism will increase with time providing the underlying algorithms will survive a few years of public scrutiny. SDRP primarily based is predicated relies on a unique and new designed identity- based signature formula. the easy modification bestowed in Section III will fix the known security drawback of this signature formula, however it's still unsure whether or not there's the other security weakness during this changed identity-based signature formula. to handle this issue, it's recommended that, rather than this novel identity-based signature formula, some economical identity-based signature algorithms that have survived a few years of public scrutiny will be directly utilized in SDRP. for instance, we will select the demonstrably secure identity-based signature projected by Barreto et al. other than providing higher security, the strategy by Barreto et al. conjointly improves the potency of SDRP because of the subsequent 2 reasons. First, its signature verification operation solely wants one pairing computation and, hence, is among the foremost economical ones. Second, the length of its signature is reduced because of linear pairing.

5.1 Improved SRDRP

1) System low-level formatting Phase: The network owner exe- cutes the subsequent steps.

1. Key setup: Generate the public parameters params $=$, and cargo them in every device node before readying, where $(G1, G2, G3)$ represents linear teams of massive prime order p with generators $g2 \in G2$, $g1 = \psi(g2) \in G1$, and $g = \hat{e}(g1, g2)$. The network owner picks a random variety $s \in Zp$ because the passkey and computes public key $Qpub = s \bullet g2 \in G2$. H3 and H4 square measure cryptological hash functions, wherever $H3 : * \rightarrow Zp$ and $H4 : * \times G3 \rightarrow Z*$.
2. User public/private key generation: For a user with identity $UIDj \in *$, the network owner sets Uj 's public key as $Pj = H3(UIDj\ Prij) \in Zp$, computes the non-public key $Sj = (1/(Pj + s)) \bullet g1 = (1/(H3(UIDj\ Prij) + s)) \bullet g1$, and then sends back to Uj through a secure channel. Here, Prij denotes the amount of user privilege and subscription amount.

2) User Preprocessing Phase: User Uj takes the subsequent actions.

1. This step is that the same as step 1) of the user preprocessing part of the initial SDRP.
2. With the non-public key Sj, Uj will cipher the signature σj of the message m as represented in the following. decide a random variety $x \in Z*$, and cipher $r = gx$.

Table 2

Running time for each part of the improved SDRP (except the device node verification particle part)

	Key setup	User public/private key generation	User signing
Time (CPU = 1.6 GHz) (μs)	5709.5	1216.5	6617.5
Time (CPU = 1.8 GHz) (μs)	5094.5	1050	5909.5
Time (CPU = 2 GHz) (μs)	4595.5	995.5	5211
Time (CPU = 2.2 GHz) (μs)	4153	852.5	4841
Time (CPU = 2.4 GHz) (μs)	3813	801	4437
Time (CPU = 2.6 GHz) (μs)	3505	720.5	4099
Time (CPU = 3.1 GHz) (μs)	2933	621	3414.5

Set $h = H4\ (m, r) \in Z*$, and cipher $W = (x + h)$ •TABLE II Sj . The signature is the pair(h, W) $\in Z* \times G1$ .CODE SIZES OF VERIFICAT particle IMPLEMENTATION OF SIGNATURE σjp MESSAGES I N T HE ORIGINA L SRDRP AND THE IMPROVED SRDRP

3) This step is that the same as step 3) of the user pre-processing phase of the initial SRDRP. Device Node Verification Phase: Upon receiving a signature message , every device node verifies it as follows.

1. This step is identical as step 1) of the device node verification part of the initial SRDRP.
2. Given the general public parameters, the device node computes $h* = H4\ m, e\ (W, H3\ (UI\ Dj\ \ P\ rij\ ) \bullet g2 + Qpub\ )\ g{-}h$ and then sees whether or not $h*$ is adequate h or not, wherever h is from σj . If the result's positive, the signature σj is valid; otherwise, the node merely drops the signature.
3. This step is identical as step 3) of the device node verification part of the initial SRDRP.

5.2 Implementation and Performance analysis

First, we have a tendency to measure the performance of the improved SRDRP with relation to the protocols operated by the network owner and user. In our experiment, the network owner and device network user aspect programs are enforced in C++ (using the number arithmetic within the publically accessible crypto logical library A Multiprecision number and Rational Arithmetic C/C++ Library (MIRACL) and dead in portable computer PCs (with2-GB RAM) below Ubuntu eleven.04 surroundings with totally different machine powers. The ηT pairing formula over the sphere F397 is employed (using Barreto's ASCII text file code). The overhead is |UI Dj | + |P rij | + |m| + |σj | = a pair of + 16+ twenty nine + 20+20 = 87 B. Obviously, the transmission overhead of the im- established SRDRP is extremely low, that is extremely appropriate for low- information measure WSNs. The memory overhead measures the precise quantity of information area needed in the real implementation. Similarly, the execution time measures the time length for the signature verification operation of the 2 protocols. The device node aspect programs square measure written in nesC and run on 2 types of resource-limited device nodes, i.e., MicaZ and TelosB motes. Our motes run TinyOS version a pair of x.

Note that the time for signature verification on device nodes will be reduced through the employment of the optimized cryptological operations. for instance, in our implementation, the exponentiation operation will be optimized.

6. FURTHER IMPROVEMENT OF SRDRP

Designing a secure reprogramming protocol is a difficult task, because there are so many details involved (e.g., the complicated interactions with the environment) that the designer can only try his/her best to make sure his/her protocol is infallible. This holds regardless of whether security proofs are supported by heuristic arguments or formal ways. In reality, the degree of confidence accompanying a security mechanism increases with time only if the underlying algorithms can survive many years of public scrutiny.

SRDRP is based on a novel and newly designed identity- based signature algorithm. The simple modification presented in Section III can fix the identified security problem of this signature algorithm, but it is still uncertain whether there is any other security weakness in this modified identity-based signature algorithm. To address this issue, it is suggested that, instead of this novel identity-based signature algorithm, some efficient identity-based signature algorithms which have survived many years of public scrutiny can be directly employed in SRDRP. For example, we can choose the provably secure identity-based signature proposed by Barreto *et al.*Aside from providing better security, the method by Barreto *et al.* also improves the efficiency of SRDRP due to the following two reasons. First, its signature verification operation only needs one pairing computation and, hence, is among the most efficient ones. Second, the length of its signature is reduced due to bilinear pairing.

7. CONCLUSION AND FUTURE WORK

In this paper, we\'ve got noted Associate in Nursing inherent style weak- terra firma within the user preprocessing part of SRDRP. The known style weakness permits Associate in nursing person to impersonate any authorized user to reprogram sensing element nodes. we\'ve got additionally bestowed a modification to repair the matter while not sacrificing any desire in a position feature of SRDRP. Moreover, we\'ve got chosen the identity- based mostly signature formula by Barreto et al. as Associate in Nursing example to point out that, for security and potency thought, any economical identity-based signature formula and position based mostly routing that has survived a few years of public scrutiny is directly used in SDRP. Though Deluge could be a factual commonplace and has been enclosed in Tiny OS distributions, many additional economical central- ized reprogramming protocols have recently been planned for WSNs, like Rate less Deluge and Dissemination Protocol (DIP). For instance, compared to Deluge, Rate- less Deluge has several blessings like reducing latency at moderate levels of packet loss, being additional ascendible to dense networks, and usually intense way less energy, a premium resource in WSNs. Thus, so as to more improve the re- programming potency of SRDRP, future work ought to target a way to integrate SDRP with a far additional economical reprogramming protocol like Rate less Deluge, resulting in additional secure and economical distributed reprogramming.

In some applications, the network owner and users area unit different entities. A user might want to multiple channel reprogramming privacy from anyone else, together with the network owner. In future work, we\'ll study a way to support user privacy preservation in distributed reprogramming.

REFERENCES

[1] D. He, C. Chen, S. Chan, and J. Bu, "SDRP: A secure and economical reprogramming protocol for wireless sensing element networks," IEEE Trans. Ind. Electron., vol. 59, no. 11, pp. 4155–4163, Nov. 2012.
[2] V. C. Gungor, B. Lu, and G. P. Hancke, "Opportunities and challenges of wireless sensing element networks in good grid," IEEE Trans. Ind. Electron., vol. 57, no. 10, pp. 3557–3564, Oct. 2010.
[3] J. Chen, X. Cao, P. Cheng, Y. Xiao, and Y. Sun, "Distributed collabora- tive management for industrial automation with wireless sensing element and mechanism networks," IEEE Trans. Ind. Electron., vol. 57, no. 12, pp. 4219–4230, Dec. 2010.
[4] X. Cao, J. Chen, Y. Xiao, and Y. Sun, "Building-environment management with wireless sensing element and mechanism networks: Centralized versus distributed," IEEE Trans. Ind. Electron., vol. 57, no. 11, pp. 3596–3604, Nov. 2010.
[5] J. Carmo, P. Mendes, C. Couto, and J. Correia, "A 2.4-GHz CMOS short- vary wireless-sensor-network interface for automotive applications," IEEE Trans. Ind. Electron., vol. 57, no. 5, pp. 1764–1771, May 2010.
[6] V. Naik, A. Arora, P. Sinha, and H. Zhang, "Sprinkler: A reliable and energy economical information dissemination service for extreme scale wireless networks of embedded devices," IEEE Trans. Mobile Comput., vol. 6, no. 7, pp. 762–776, Jul. 2007.
[7] L. Mottola and G. Picco, "Programming wireless sensing element networks: Funda-mental ideas and state of the art," ACM Comput. Surv., vol. 43, no. 3, pp. 1–51, Apr. 2011.
[8] H. Song, V. Shin, and M. Jeon, "Mobile node localization victimization fusion prediction-based interacting multiple model in cricket sensing element network," IEEE Trans. Ind. Electron., vol. 59, no. 11, pp. 4349–4359, Nov. 2010.
[9] R. C. Luo and O. Chen, "Mobile sensing element node preparation and asyn- chronous power management for wireless sensing element networks," IEEE Trans. Ind. Electron., vol. 59, no. 5, pp. 2377–2385, May 2012.
[10] H. Tan, J. Zic, S. Jha, and D. Ostry, "Secure multihop network program- dynasty with multiple unidirectional key chains," IEEE Trans. Mobile Comput., vol. 10, no. 1, pp. 16–31, Jan. 2011.
[11] P. K. Dutta, J. W. Hui, D. C. Chu, and D. E. Culler, "Securing the deluge schedule system," in Proc. IPSN, 2006, pp. 326–333.
[12] C. Lim, "Secure code dissemination and remote image management victimization ephemeral signatures in WSNs," IEEE Commun. Lett., vol. 15, no. 4, pp. 362–364, Apr. 2011.
[13] Doh, J. Lim, and K. Chae, "Code updates supported minimal backbone and cluster key management for secure sensing element networks," Math. Comput. Model., 2012, to be printed.
[14] Y. Law, Y. Zhang, J. Jin, M. Palaniswami, and P. Havinga, "Secure rate- less deluge: Pollution-resistant reprogramming and information dissemination for wireless sensing element networks," EURASIP J. Wireless Commun. Netw., vol. 2011, no. 1, pp. 1–21, Jan. 2011.
[15] C. Parra and J. Garcia-Macias, "A protocol for secure and energy-aware reprogramming in WSN," in Proc. IWCMC, 2009, pp. 292–297.
[16] N. Bui, O. Ugus, M. Dissegna, M. Rossi, and M. Zorzi, "An integrated system for secure code distribution in wireless sensing element networks," in Proc. PERCOM, 2010, pp. 575–581.
[17] S. Hyun, P. Ning, A. Liu, and W. Du, "Seluge: Secure and DoS-resistant code dissemination in wireless sensing element networks," in Proc. IPSN, 2008, pp. 445–456

BIOGRAPHIES

E. Lalitha doing M.Tech multimedia Technology in K.S.R. College of Engineering, she completed B.Tech Information Technology, Her current research includes VANET, Sensor Network