ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization) Vol.2, Special Issue 1, March 2014

Proceedings of International Conference On Global Innovations In Computing Technology (ICGICT'14)
Organized by
Department of CSE, JayShriram Group of Institutions, Tirupur, Tamilnadu, India on 6th & 7th March 2014

An Enhanced Rationalize Security and Efficient Data Gathering In Wireless Sensor Networks

Karthick.S, Senthil Kumar.V
KSR College of Engineering, Tiruchengode, Namakkal Dist, Tamilnadu, India

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 doable 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 [1]–[9]. 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 [10]–[15]. 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 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

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization) Vol.2, Special Issue 1, March 2014

Proceedings of International Conference On Global Innovations In Computing Technology (ICGICT'14)

Organized by

Department of CSE, JayShriram Group of Institutions, Tirupur, Tamilnadu, India on 6th & 7th March 2014

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 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.

Quite recently, He et al. have projected a secure and distributed reprogramming protocol named SRDRP [16], that is that the initial work of its kind.

Since a completely unique identity-based signature theme is used in generating public/private key combine of every licensed user, SRDRP is economical for resource-limited device nodes and mobile devices in terms of communication and storage needs. what is more, SRDRP will deliver the goods all needs of distributed reprogramming listed in [12], whereas keeping the deserves of the well-known mechanisms like Deluge and Seluge [17]. Also, SRDRP has been implemented in a very network of resource-limited device nodes to point out its high potency.

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.

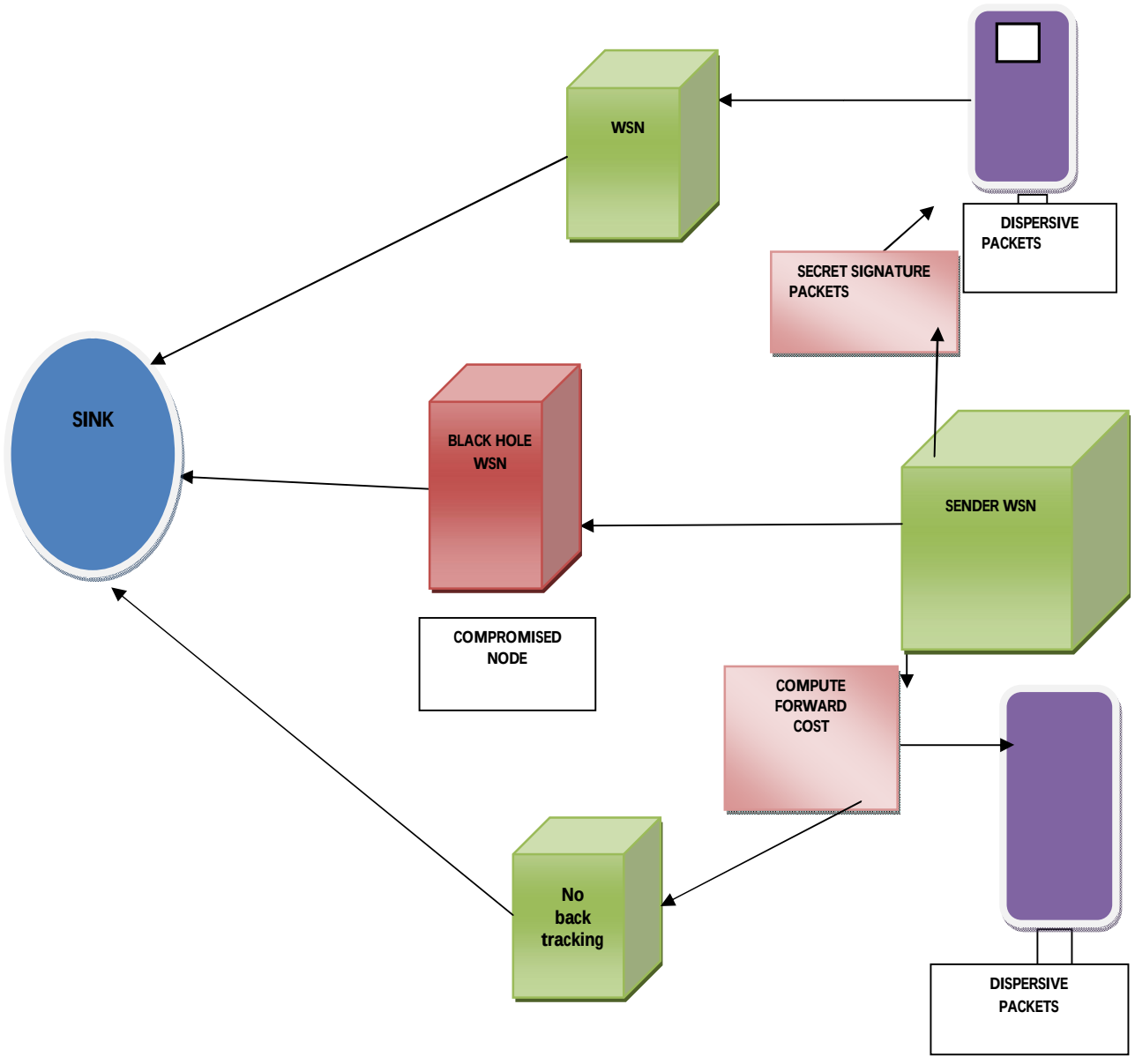
ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization) Vol.2, Special Issue 1, March 2014

Proceedings of International Conference On Global Innovations In Computing Technology (ICGICT'14)
Organized by
Department of CSE, JayShriram Group of Institutions, Tirupur, Tamilnadu, India on 6th & 7th March 2014

1.1 SYSTEM ARCHITECTURE



ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization) Vol.2, Special Issue 1, March 2014

Proceedings of International Conference On Global Innovations In Computing Technology (ICGICT'14)

Organized by

Department of CSE, JayShriram Group of Institutions, Tirupur, Tamilnadu, India on 6th & 7th March 2014

However, during this paper, we tend to demonstrate that a style weakness exists in the user 1) preprocessing section of SRDRP, associate degrade an antagonist will simply impersonate any approved user to hold out reprogramming. Let P be a generator of G. Let eˆ : G × G → GT be a additive map. To eliminate the known security vulnerability, we propose 2) decide random s ∈ Z∗ as the passes part out, and figure a straightforward modification on SRDRP while not losing any feature public key PK owner= s • P .(such as distributed reprogramming, supporting completely different user privileges, dynamic participation, quantifiability, high potency, and sturdy security) of the first protocol. 3) select 2 secure science hash functions H1 and H2 , where H1 : ∗ → G and H2 : ∗ → 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.

.

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 or 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 Uj with identity UI Dj ∈ ∗ , the network owner sets Uj 's public key as P Kj = H1 (UI Dj P rij) ∈ G, computes the personal key SKj = s • P Kj , so sends back to Uj employing a secure channel, like the wired transport layer security protocol. Here, P rij denotes the amount of user privilege (e.g., the sensing element node set among a particular region that user Uj is allowed to reprogram) and subscription amount.

2.2 User Pre-processing section

User Uj takes the subsequent actions. 1) Uj partitions the code image to Y fixed-size pages, de- noted as page one through page Y . Uj splits page i ($1 \leq i \leq Y$) into N fixed-size packets, denoted as P kti,1 through P kti,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 P ktY,1 h(P ktY,1) is enclosed in packet P ktY −1,1 . Here, P ktY,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 Uj 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 [13] 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

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization) Vol.2, Special Issue 1, March 2014

Proceedings of International Conference On Global Innovations In Computing Technology (ICGICT'14)
Organized by
Department of CSE, JayShriram Group of Institutions, Tirupur, Tamilnadu, India on 6th & 7th March 2014

regarding the code image (e.g., version variety, targeted node identity set, and code image size), and a signature over all of them or enclosed in an exceedingly signature message. The careful info are often stated in [17]. 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.

That is, υ = (H2 (m))−2 (mod q) = u (mod q), wherever u • H2 (m) ≡ 2(mod q). It ought to be noted that image. Then, in order to make sure the credibleness and any number in Z∗ has a reciprocal if and

integrity of the new code image, Uj takes the subsequent actions to construct the signature message.

2) With the personal key SKj , Uj will figure the signature

σj of the message m, wherever σj = H2 (m) • 2S1Kj .

3) Uj transmits to the targeted nodes the signature message, that is the notification of the new code image. SDRP depends on the underlying Deluge protocol to distribute packets for a given code image.

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:

eˆ(σj ,P) = eˆ (H2 (m) • H1 (UI Dj |P rij),P K owner) . (1) If the equation holds, the signature σj is valid.

3) If the aforesaid verification passes, the sensing element node

believes that the message m and also the privilege P rij or from a licensed user with identity UI Dj . 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 antecedently pass, the sensing element node accepts the code image.

3. SECURITY WEAKNESS OF SRDRP

Recall that, within the user preprocessing section of SRDRP, the signature σi is computed as H2 (m) • SKj . This is, however, a style weakness as a result of it allows associate degree antagonist A to get the personal key SKj of user Uj as shown within the following.

1) whereas Uj transmits to the targeted nodes the signature message , the antagonist A will get by eavesdropping. With the public parameter H2 , the antagonist A will figure υ = (H2 (m))−1 (mod q), wherever H2 (m) ∈ 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 H2 (m) modulo alphabetic character exists if and given that H2 (m) and alphabetic character or coprime (i.e., gcd(H2 (m), q) =

3). In SDRP, since alphabetic character is prime, all of the nonzero integers in Z∗ or comparatively prime to alphabetic character, and so, there exists a reciprocal for all of the nonzero integers in Z∗ .

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization) Vol.2, Special Issue 1, March 2014

Proceedings of International Conference On Global Innovations In Computing Technology (ICGICT'14)

Organized by

Department of CSE, JayShriram Group of Institutions, Tirupur, Tamilnadu, India on 6th & 7th March 2014

4) The antagonist A computes the personal key SKj = υ • σi .

Consequently, the antagonist A will impersonate user

Uj 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 Uj .

4. SECURITY IMPROVEMENT OF SRDRP

Clearly, if H2 (m) and alphabetic character don't seem to be coprime, associate degree antagonist cannot figure the personal key SKj . 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 , 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 or coprime.

If yes, before a signature on m is computed, redundant bits or 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 SRDRP, the advised modification doesn't incur any overhead on the sensing element node aspect.

In the style of SRDRP, 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 q, m is willy-nilly generated for a thousand times. Thus, every experiment has one million 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 or 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.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization) Vol.2, Special Issue 1, March 2014

Proceedings of International Conference On Global Innovations In Computing Technology (ICGICT'14)
Organized by
Department of CSE, JayShriram Group of Institutions, Tirupur, Tamilnadu, India on 6th & 7th March 2014

TABLE 1
FAILURE RATES AND SEARCH TIME F OR THE ADDITION OF ONE OR 2 REDUNDANT BYTES ONCE ALPHATEIC CHARAACTER 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	*68.12*	*608.91*	*50.82*	*598.01*	*49.34*	*529.85*	*48.18*	*368.77*	*40.02*	*300.65*
Failure rate(%)	*0.45970*	*0.0691*	*0.7380*	*0.1851*	*0.7550*	*0.1780*	*0.9752*	*0.156*	*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 or 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 implement thinking results regarding the common search time of acceptable redundant information of one or two B or summarized. 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 or 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 preprocessing procedure or obtained in terms of computation, memory usage, and transmission and reception powers. As shown in Tables I, our experimental results demonstrate 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) or 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. PROVABLE SECURITY AGAINST ATTACKS

Here we modify the forwarding phase of PLGP to provably avoid the above-mentioned attacks. First we introduce the no- backtracking property, satisfied for a given packet if and only

```
Function forward_packet(p)

  s ← extract_source_address(p);
  c ← closest_next_node(s);
  if is_neighbor(c) then forward(p, c);
```

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization) Vol.2, Special Issue 1, March 2014

Proceedings of International Conference On Global Innovations In Computing Technology (ICGICT'14)

Organized by

Department of CSE, JayShriram Group of Institutions, Tirupur, Tamilnadu, India on 6th & 7th March 2014

```
else
    r ← next_hop_to_non_neighbor(c);
    forward(p, r);
```

Function secure_forward_packet(p)

```
s ← extract_source_address(p);
a ← extract_attestation(p);
if (not verify_source_sig(p)) or (empty(a) and
 not is_neighbor(s)) or (not saowf_verify(a)) then
    return ;                                  /* drop(p) */
foreach node in a do
    prevnode ← node;
    if (not are_neighbors(node, prevnode)) or
    (not making_progress(prevnode, node)) then
        return ;                              /* drop(p) */
c ← closest_next_node(s);
p' ← saowf_append(p);
if is_neighbor(c) then forward(p', c);
else forward(p',
 next_hop_to_non_neighbor(c));
```

if it consistently makes progress toward its destination in the logical network address space. More formally:

Definition 1. No-backtracking is satisfied if every packet p traverses the same number of hops whether or not an adversary is present in the network. (Maliciously-induced route stretch is bounded to a factor of 1.)

This does not imply that every packet in the network must travel the same number of hops regardless of source or destination, but rather that a packet sent to node D by a malicious node at location L will traverse the same number of hops as a packet sent to D by a node at location L that is honest. If we think of this in terms of protocol execution traces, no-backtracking implies that for each packet in the trace, the number of intermediate honest nodes traversed by the packet between source and destination is independent of the actions of malicious nodes. Equivalently, traces that include malicious nodes should show the same network-wide energy utilization by honest nodes as traces of a network with no malicious actors. The only notable exceptions are when adversaries drop or mangle packets en route, but since we are only concerned with packets initiated by adversaries, we can safely ignore this situation: "pre-mangled" packets achieve the same result — they will be dropped by an honest intermediary or destination. No-backtracking implies Vampire resistance. It is not immediately obvious why no-backtracking prevents Vampire attacks in the forwarding phase. Recall the reason for the success of the stretch attack: intermediate nodes in a source route cannot check whether the source-defined route is optimal, or even that it makes progress toward the destination.

When nodes make independent routing decisions such as in link-state, distance-vector, coordinate-based, or beacon- based protocols, packets cannot contain maliciously composed routes. This already means the adversary cannot perform carousel or stretch attacks — no node may unilaterally specify a suboptimal path through the network. However, a sufficiently clever adversary may still influence packet progress. We can prevent this interference by independently checking on packet progress: if nodes keep track of route "cost" or metric and, when forwarding a packet,

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization) Vol.2, Special Issue 1, March 2014

Proceedings of International Conference On Global Innovations In Computing Technology (ICGICT'14)
Organized by
Department of CSE, JayShriram Group of Institutions, Tirupur, Tamilnadu, India on 6th & 7th March 2014

communicate the local cost to the next hop, that next hop can verify that the remaining route cost is lower than before, and therefore the packet is making progress toward its destination. (Otherwise we suspect malicious intervention and drop the packet.) If we can guarantee that a packet is closer to its destination with every hop, we can bound the potential damage from an attacker as a function of network size. (A more desirable property is to guarantee good progress, such as logarithmic path length, but both allow us to obtain an upper bound on attack success.)

PLGP does not satisfy no-backtracking. In non-source routing protocols, routes are dynamically composed of forwarding decisions made independently by each node. PLGP differs from other protocols in that packets paths are further bounded by a tree, forwarding packets along the shortest route through the tree that is allowed by the physical topology. In other words, packet paths are constrained both by physical neighbor relationships and the routing tree. Since the tree implicitly mirrors the topology (two nodes have the same parent if and only if they are physical neighbors, and two nodes sharing an ancestor have a network path to each other), and since every node holds an identical copy of the address tree, every node can verify the optimal next logical hop. However, this is not sufficient for no-backtracking to hold, since nodes cannot be certain of the path previously traversed by a packet. Communicating a local view of route cost is not as easy as it seems, since adversaries can always lie about their local metric, and so PLGP is still vulnerable to directional antenna/wormhole attacks, which allow adversaries to divert packets to any part of the network.

The resulting protocol, PLGP with attestations (PLGPa) uses this packet history together with PLGP's tree routing structure so every node can securely verify progress, preventing any significant adversarial influence on the path taken by any packet which traverses at least one honest node. Whenever node n forwards packet p, it this by attaching a non replay able attestation (signature). These signatures form a chain attached to every packet, allowing any node receiving it to validate its path. Every forwarding node verifies the attestation chain to ensure that the packet has never travelled away from its destination in the logical address space. See Function secure_forward_packet for the modified protocol.

- causing honest node I to forward p with non-null attestation, over a route that backtracked, violating the assumption that honest nodes correctly follow PLGPa
- causing honest node I to forward p with a non-null attestation, from source S who is I's direct neighbor, violating the assumption that honest nodes correctly follow PLGPa;
- truncating the route attestation, violating the security of chain signatures.

PLGPa satisfies no-backtracking. To show that our modified protocol preserves the no-backtracking property, we define a network as a collection of nodes, a topology, connectivity properties, and node identities. Honest nodes can broadcast and receive messages, while malicious nodes can also use directional antennas to transmit to (or receive from) any node

in the network without being overheard by any other node. Honest nodes can compose, forward, accept, or drop messages, and malicious nodes can also arbitrarily transform them. Our adversary is assumed to control m nodes in an N-node network (with their corresponding identity certificates and other secret cryptographic material) and has perfect knowledge of the network topology. Finally, the adversary cannot affect connectivity between any two honest nodes.

Since all messages are signed by their originator, messages from honest nodes cannot be arbitrarily modified by malicious nodes wishing to remain undetected. Rather, the adversary can only alter packet fields that are changed en route (and so are not authenticated), so only the route attestation field can be altered, shortened, or removed entirely. To prevent truncation, For the purposes of Vampire attacks, we are unconcerned about packets with arbitrary hop counts that are never received by honest nodes but rather are routed between adversaries only, so we define the hop count of a packet as follows:

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization) Vol.2, Special Issue 1, March 2014

Proceedings of International Conference On Global Innovations In Computing Technology (ICGICT'14)
Organized by
Department of CSE, JayShriram Group of Institutions, Tirupur, Tamilnadu, India on 6th & 7th March 2014

6. 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 [17].

SRDRP 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 SRDRP. 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 SRDRP 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.

6.1 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 ∈ G2 , g1 = ψ(g2) ∈ G1 , and g = eˆ(g1 , g2). The network owner picks a random variety s ∈ Zp because the passkey and computes public key Qpub = s • g2 ∈ G2 . H3 and H4 square measure cryptological hash functions, wherever H3 : ∗ → Zp and H4 : ∗ × G3 → Z∗ .

2) User public/private key generation: For a user

with identity UI Dj ∈ ∗ , the network owner sets Uj 's public key as Pj = H3 (UI Dj P rij) ∈ Zp , computes the non-public key Sj = (1/(Pj + s)) • g1 = (1/(H3 (UI Dj P rij)+ s)) • g1 , and then sends

back to Uj through a secure channel. Here, P rij denotes the amount of user privilege (e.g., the device node set at intervals a selected region that user Uj is allowed to reprogram) and subscription amount.

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

4) This step is that the same as step 1) of the user preprocessing part of the initial SDRP.

5) 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 ∈ Z∗ , and cipher r = gx .

Sj . The signature is the pair(h, W) ∈ Z∗ × G1.

6) This step is that the same as step 3) of the user preprocessing phase of the initial SRDRP.

7) device Node Verification Phase: Upon receiving a signature message , every device node verifies it as follows.

8) This step is identical as step 1) of the device node verification part of the initial SRDRP.

9) Given the general public parameters, the device node computes h∗ = H4 m, e (W, H3 (UI Dj P rij) • 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.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization) Vol.2, Special Issue 1, March 2014

Proceedings of International Conference On Global Innovations In Computing Technology (ICGICT'14)
Organized by
Department of CSE, JayShriram Group of Institutions, Tirupur, Tamilnadu, India on 6th & 7th March 2014

10) This step is identical as step 3) of the device node verification part of the initial SRDRP.

6.2 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 cryptological library A Multiprecision number and Rational Arithmetic C/C++ Library (MIRACL) [26]) and dead in portable computer PCs (with2-GB RAM) below Ubuntu eleven.04 surroundings with totally different machine powers. The ηT [27] pairing formula over the sphere F397 is employed (using Barreto's ASCII text file code [28]). Thus, the pairing is bilateral, the length of every component of G1 is twenty B, and also the length of every component of Z∗ is additionally twenty B. because the original SDRP, in our implementation, the computer memory unit lengths of UI Dj , P rij , and m square measure set to a pair of, 16 provides the execution time of some major operations within the improved SRDRP. we've dead every operation for twenty 000 times and

taken a median over them. for instance, the execution times of the network owner for the key setup part and generating public/private key operation (for every network user) square measure three.505 and 0.7205 ms on a a pair of.6-GHz notebook computer, severally. Also, the signature generation time for a network user is vi.6175 ms on a 1.6-GHz notebook computer.

Next, we have a tendency to contemplate the signature message overhead of the improved SRDRP while not considering packet headers. The over- head 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.

In typical WSNs, device nodes square measure sometimes resource constrained and battery restricted, and that they might not be ready to execute high-ticket cryptological operations with efficiency and therefore become the bottleneck of a security protocol.

Compared to the signature verification formula of the initial SRDRP that in the main requires 2 pairing, one hash-to-point (with 18-B message (i.e., (UI Dj P rij)) as input), and one purpose scalar multiplication operations on a device node, the signature verification formula of the improved SRDRP in the main needs one pairing, one purpose scalar multiplication, and one mathematical process (over the sphere F(397)6) operations on a device node and, thus, is a lot of economical. We use the subsequent 2 metrics to match the initial SRDRP with the improved SRDRP, namely, memory overhead and execution time. 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 [29] version a pair of x.

The MicaZ atom options associate 8-b 8-MHz Atmel microcontroller with 4-kB RAM and 128-kB store. Also, the TelosB atom has associate 8-MHz CPU, 10-kB RAM, 48-kB ROM, one MB of flash memory, and an 802.15.4/ZigBee radio. Different from, here, the implementation of device node aspect programs is totally supported TinyPairing (a pairing-based cryptological library), and that we don't do any optimisation. for instance, the pairing, purpose scalar multiplication, and SHA-1 hash operations from TinyPairing square measure utilized. This implementation in the main involves Pairing, Base Field, ExtField2, SHA1, PointArith, and NN parts of TinyPairing. in keeping with Barreto's ASCII text file code for the ηT pairing operation, we have a tendency to implement the mathematical process operation over the sphere F(397)6 supported TinyPairing.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization) Vol.2, Special Issue 1, March 2014

Proceedings of International Conference On Global Innovations In Computing Technology (ICGICT'14)

Organized by

Department of CSE, JayShriram Group of Institutions, Tirupur, Tamilnadu, India on 6th & 7th March 2014

The ensuing size of our implementation for the improved SRDRP corresponds to solely nineteen.92% and 18.47% of the RAM and store capacities of MicaZ, severally. it's clear that the memory overhead of the improved SRDRP is adore that of the initial SRDRP. Provides the execution time for signature verification of the initial SRDRP and also the improved SRDRP. for instance, with relation to the improved SRDRP, the execution times for signature verification on a MicaZ atom and TelosB atom square measure concerning ten.65 and 24.85 s, severally. As incontestable in [21], the proportion of this signature verification time within the total reprogramming time is extremely tiny. Clearly, our experiment results show that, for the signature verification operation on device nodes, the improved SRDRP is a lot of economical than the initial SRDRP.

The execution time for every cryptological operation on MicaZ and TelosB motes. for instance, the ηT pairing operation takes five.3216 and thirteen.0137 s on a MicaZ atom and a TelosB atom, severally. For the leads we have a tendency to perform every operation two hundred times and take a median over them.

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.

7.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 [18].

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.* [17]. 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.

8. CONCLUSION AND FUTURE WORK

In this paper, we have 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 have got additionally bestowed a modification to repair the matter while not sacrificing any desiring a position feature of SRDRP. Moreover, we have 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 SRDRP. though Deluge could be a factual commonplace and has been enclosed

in TinyOS distributions, many additional economical centralized reprogramming protocols have recently been planned for WSNs, like Rateless Deluge and DIssemination Protocol (DIP). for instance, compared to Deluge, Rate- less Deluge has

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization) Vol.2, Special Issue 1, March 2014

Proceedings of International Conference On Global Innovations In Computing Technology (ICGICT'14)
Organized by
Department of CSE, JayShriram Group of Institutions, Tirupur, Tamilnadu, India on 6th & 7th March 2014

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 Rateless 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 have 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 collaborative 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 Computing., 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 asynchronous 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] I. 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