World Applied Sciences Journal 29 (Computer Sciences, Engineering and Its Applications): 292-298, 2014
ISSN 1818-4952

DOI: 10.5829/idosi.wasj.2014.29.csea.2248

Efficient Broadcast Authentication with Maximum Lifetime in Wireless Sensor Networks Using X-TESLA

K.P. Thooyamani, R. Udayakumar and V. Khanaa

School of Computing Science, Bharath University, Chennai-73, India

Abstract: We consider the problem of Authenticated broadcasting messages in a wireless energy-limited network so as to maximize network lifetime. It enabling the base station to send commands and requests to low- powered sensor nodes in an authentic manner is one of the important challenges for securing wireless sensor networks. X-TESLA is a mechanics, which is considering the problem arising from sleep modes, network failures, idle sessions and DoS attacks. But the problem of power energy should not be considered. The lifetime of the sensor nodes may be ended when the broadcasting process. In proposed system we will implement algorithm namely greedy heuristics with X-TESLA for maximize the life time of the sensor networks. The proposed implementation improves network lifetime significantly when compared with network lifetime using the base greedy heuristics in isolation.

Key words: Wireless Networks · Network Lifetime · Security · Broadcast Authentication

INTRODUCTION

Technological advancement in large-scale distributed networking and small sensor devices has led to the development of wireless sensor networks with numerous applications [1]. Sensor nodes are usually constrained in their computation, communication, storage and energy resources for economical reasons, but need security functions since they are deployed in unattended or even hostile environments. The high risk of physical attacks and the limited capabilities of sensor nodes make it difficult to apply traditional security techniques to wireless sensor networks, posing new challenges [7]. Authenticated broadcast, enabling a base station to send authentic messages to multiple sensor nodes, is one of the core challenges [3], while even the broadcast by nodes is an important topic in wireless sensor networks [2, 6, 4]. For the purpose, digital signatures (public key) are not very useful in a resource-limited environment, while naý¨ve use of HMAC (secret key) does not work either, as node capture can lead to a key compromise.

Battery-operated wireless sensor networks may be deployed in environments in which it is impractical to recharge/replace the battery of a sensor (e.g., in a battle field or at the bottom of an ocean). Hence, these networks must operate subject to the constraint that the energy available to a sensor isn't replenish able. In other wireless network applications, even though it is possible to recharge a node's battery (or replenish its energy supply), it is desirable to operate in an energy frugal manner so as to minimize the need for this recharge. With this need to conserve energy in many wireless network applications, several authors have developed energy-efficient algorithms for point-to-point communication, multicasting and broadcasting. The overall objective of these algorithms is to either maximize the lifetime (number of successful communications before first communication that cannot be done) or the capacity of the network (amount of data traffic carried by the network over some fixed period of time). Lifetime maximization is considered for wireless sensor networks.

In the most common model used for power attenuation in wireless broadcast, signal power attenuates at the rate a/r^d, where a is a media dependent constant, r is the distance from the signal source and d is another constant between 2 and 4 [5]. So, for this model, w(i, j) w(j, i) = c ◎ $r(i, j)^d$, where r(i, j) is the Euclidean distance between nodes i and j and c is a constant. In practice,

Corresponding Author: K.P. Thooyamani, School of Computing Science, Bharath University, Chennai-73, India.

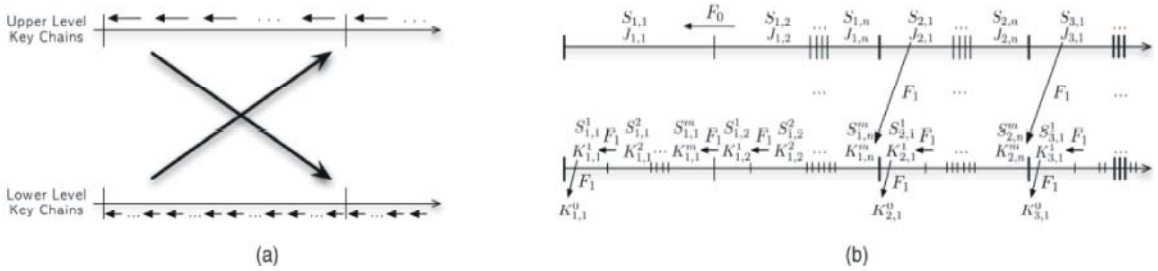

Fig. 1: Basic concept of X-TESLA. (a) Cross authentication. (b) Basic flows

however, this nice relationship between w(i, j) and r(i, j) may not apply. This may, for example, be due to obstructions between the nodes that may cause the attenuations to be larger than predicted. Also, the transmission properties of the media may be asymmetric resulting in w(i, j) ≠ w(j, i). In this paper, we assume the most general case in which edge weights may be asymmetric and may reflect the presence of obstructions in the broadcast path.

Our basic idea starts from the extendable management of short key chains. In essence, we make two levels of chains having distinct time intervals cross-authenticate each other (Fig. 1a) to provide permanently extendable chains. Our protocol X-TESLA, read either as eks TESLA or cross TESLA, stands for extendable TESLA. As with other TESLA variants, X-TESLA provides broadcast authentication, under the assumption that the base station and sensor nodes are loosely time synchronized with a known maximum synchronization discrepancy.

Problem Definition:

64-Bit Key Chain: A short 64-bit key chain is desirable for efficiency in resource limited sensor nodes, but care must be taken, even with short time intervals. As we show, if the chain is generated in a straight forward manner, Time-Memory-Data (TMD) tradeoff techniques can be applicable, leading to discovery of future keys.

Sleep Mode or Network Failure: If sensor nodes go into a sleep mode or key disclosure messages are lost frequently, X TESLA may force heavy key computation to be done at once on sensor nodes for chain verification, during which incoming packets get dropped.

If Commitment Distribution Messages (CDMs) are missing, multilevel X TESLA makes nodes wait and buffer for the long interval of upper levels, during which incoming packets are dropped due to the buffer limit.

Idle Sessions: Even for idle sessions with no broadcasts, X TESLA forces chain computation for sensor nodes. Key disclosure messages should be broadcast constantly or heavy computation needs to be done later. Multilevel X TESLA needs CDMs to be broadcast for higher levels, with the number of CDMs increasing with the number of levels.

Extended Lifetime: With node malfunctions and premature power exhaustion, there are needs for node additions [1] or rechargeable sensor nodes [8]. Thus, the lifetime of a network may extend beyond that of each node. As noted in [5], [19], lifetime extension was not clearly considered in X TESLA. Multilevel X TESLA should also fix the lifetime.

DoS Attacks: To resist DoS attacks, multilevel X TESLA requires many CDMs to be distributed for longer intervals. Its DoS tolerant version needs sufficiently large buffers on sensor nodes for random selection of received CDMs. The DoS-resistant version requires CDMs to be received stably along with a larger packet and additional hash function.

Let T be a broadcast tree. Following a broadcast using the broadcast tree T, the residual energy, re(i, T), at node i is:

$$re(i, T) = ce(i) - \max\{w(i, j) | j \text{ is a child of } i \text{ in } T\} \geq 0$$

The critical energy, CE(T), following the broadcast is defined to be:

$$CE(T) = \min\{re(i, T) | 1 \leq i \leq n\}$$

In the maximum critical energy problem (MCEP), we are given a network and a source node s and are to find a broadcast tree T rooted at s such that CE(T) is maximum. This maximum value of CE(T) is called the maximum critical energy and is denoted MCE(G, s).

Intuitively, by using a broadcast tree T that maximizes CE(T), we maximize our chances of being able to complete the next broadcast request. Hence, we expect to prolong network lifetime by maximizing CE(T) following each broadcast.

Algorithm for MCE (G, s): Given a source node s, our strategy to determine MCE(G, s) is to first obtain a sorted list, L, of candidate values for MCE. Next, we perform a binary search on the values in L to determine MCE(G, s).

Determine Candidate List L: For each node i of G, define the set a(i) of residual energy values as below.

$a(i) = \{ce(i) - w(i, j) | (i, j) \text{ is an edge of G and } ce(i) \geq w(i, j)\}$

Let l(i) denote the set of all possible values for re(i) following the broadcast. We see that:

$$l(i) = \begin{cases} a(i) & \text{if } i = s \\ a(i) \cup \{ce(i)\} & \text{otherwise} \end{cases}$$

Consequently, the sorted list of all possible values for MCE(G, s) is given by

$$L = sort(\cup_{1 \leq i \leq n} l(i))$$

We assume that G is represented using adjacency lists [3]. Since each l(i) may be computed in $O(di^{out})$ time, where di^{out} is the out-degree of node i, all l(i) may be computed in O(n + e) = O(e) time. From the l(i) s, L may be computed in O(e log e) time using a sort method such as merge sort [3]. So, the total time needed to compute L is O(e log e). Note that |L| = O(e). Let compute L(G, s) denote an algorithm that determines the sorted list L using the strategy just described.

X-TESLA: Secure and Efficient Broadcast Authentication Protocol

Overview of X-TESLA: Our basic idea starts from the extendable management of short key chains. In essence, we make two levels of chains having distinct time intervals cross-authenticate each other (Fig. 1a) to provide permanently extendable chains. Our protocol X-TESLA, read either as eks TESLA or cross TESLA, stands for eXtendable TESLA. As with other TESLA variants, X-TESLA provides broadcast authentication, under the assumption that the base station and sensor nodes are loosely time synchronized with a known maximum Synchronization discrepancy.

The crossing of Fig. 1a illustrates the followings:

- The lower level chain naturally authenticates the next upper level chain, as they are connected in a single chain by construction. Commitment of the next lower level chain repeatedly.

Multiple distinct keys in the upper level chain authenticate the initial value. The repeated authentication will help in resolving problems from DoS attacks, sleeping nodes and idle sessions.

X-TESLA Details

Initialization: We assume a base station broadcasts authenticated messages to sensor nodes. A method to choose salt values is fixed at system design phase. The base station generates the first upper level chain by choosing seed key $J_{1,n} \in K$ at random and also generates the first lower level chain together with the second upper level chain by choosing another seed key J2,n $\in$ K randomly. The values $J_{1,0} = F_1(J_{1,1}, S_{1,1})$ and $K^0_{1,1}$ are stored in each sensor node as initial upper and lower level commitments, respectively. Depending on the way salt is chosen, some extra information may also need to be stored. It would be advisable to keep these values secret until just before deployment. Generation of the second lower level chain together with the third upper level key chain should soon follow, so as to be ready for commitment distribution. When the initialized nodes are deployed, they are to be loosely time synchronized with the base station, as assumed in X TESLA.

Broadcast Authentication: During an $I^w_{u,v}$, the base station uses $K^w_{u,v}$ as the MAC key for Types 1, 2 and 3 packets being sent out and reveals $K^w_{u,v}$ after a wait of time δ from the end of $I^w_{u,v}$, in Type 2 or 3 packets. We shall abuse interval indices, setting The following is a Type 2 packet for use with "δ=one time interval." Here, j denotes:

$$I_{u,n+l} = I_{u+1,1}, I^{m+1}_{u,v} = I^1_{u,v+1}, I_{u,0} = I_{u-1,n}, and\ I^0_{u,v} = I^m_{u,v-1}.$$

Concatenation and signifies the index and data portion.

$$T2P^w_{u,v} = \langle u, v, w \rangle \,||\, data \,||\, MAC_{K^w_{u,v}}(*) \,||\, K^{w-2}_{u,v}.$$

Commitment Hopping With TESLA variants, there are at least two situations in which verification of a newly disclosed key places heavy computational load on a

sensor node, resulting in many message drops, for the duration of this computation. First, if a sensor node falls into sleep mode or turns off its radio power to save energy, it may not be able to listen to the key disclosure messages during that period. Second, if there are long idle periods with no broadcast, it would be wasteful to disclose keys on schedule and a base station might minimize the key disclosures for those periods. As a result, there could be a large gap between the current commitment and the key to be verified. Type 3 packets can resolve this problem, by providing commitment hopping. Let $I_{u',v'}$ be an interval appearing after $I_{u,v}$. The distance between the two intervals depends on the application needs. We set:

$$T3P_{u,v}^{w} = \langle u, v, w \rangle \,||\, K_{u',v'}^{m} \oplus Ju', v' \,||\, MAC_{K_{u,v}^{w}}(*) \,||\, K_{u,v}^{w-2}.$$

The future lower level key $K^{m}_{u',v'}$ is masked by the future upper level key $J_{u',v'}$ and distributed in $I^{w}_{u,v}$. A node can authenticate it quickly within a few lower intervals, but Unmask it afterwards only when $J_{u',v'}$ is obtained. After unmasking, it may replace the current lower level commitment, should it be older. In the opposite direction, $K^{m}_{u',v'}$ can be used to reveal $J_{u',v'}$ .If v' is close to n, $K^{m}_{u',v'}$ an be used as the next upper level commitment.

Cross Authentication: With X-TESLA, keys of the upper level chain can be authenticated by the previous lower level chain since they are connected in a single chain by construction and since the latest commitment key of the previous lower level is available to sensor nodes. Type 3 packets further help in making this available. After any verification, the commitment for the upper level can be updated. For authentication of a new lower level chain, the upper level chain is used. The following is a Type 4 packet. It distributes the commitment of the next lower level chain while disclosing a previous upper level key.

$$T4P_{u,v} = \langle u, v \rangle \,||\, K_{u+1,1}^{0} \,||\, MAC_{J_{u,v}}(*) \,||\, J_{u,v-1}.$$

Flexible Constructions: We now place more flexibility, in addition to the choice of chain lengths, into the X-TESLA construction. This will resolve even the most extreme situation that could occur with Type 4 packets. Starting from the basic flow of Fig. 3a, we extend the upper level chain over a number of lower level chains for better survivability against high communication faults and long idle sessions, as depicted in Fig. 3b. Even a short extension of the upper level chain with only small bits allows many lower level chains to be attached and these may be generated on the fly. The extension increases stability of chain verification in both levels. The change also provides longer periods in which to distribute the next chain commitments for both levels through Type 3 and Type 4 packet variants The reverse flow depicted in Fig. 3c allows reduction of Type 4 packets for environments in which authenticated messages are broadcast very frequently. Since an upper level chain serves as commitments for the next lower level chain, Type 4 packets distribute $J_{u+1,0} := F_0(J_{u+1,1}, S_{u+1,1})$ instead of $K^{0}_{u+1,1}$ in this version. But the dependence on Type 4 packets is smaller, because the upper level keys can be recovered stably from Type 3 packets if the authenticated broadcasts of the lower level are very frequent. The hybrid flow of Fig. 3d, offers extreme durability.

Sleep Mode Management: Energy efficiency is mandatory for sensor networks since tiny nodes are operated on batteries. Various types of sleep modes11 that stop CPUs or radio functions are commonly used but care must be taken [16], as nodes that have been inactive for a long time may need to do much computation for key verification or lose commitment.

Let T_u denote the starting time of interval I_u and set $\Phi = T_{u+1} - T_u$ to the length of one upper level chain. As depicted in Fig. 4, a sensor node shall not be allowed to go into a long-term sleep or, at the least, not be allowed to stop radio functions for a long-term period unless it has obtained the next lower level commitment, while short-term sleeps are always allowed. More specifically, we fix some threshold value è that takes the clock discrepancy of nodes into account and for a node that has verified a Type 4 packet at time T, we allow it to set the maximum sleep length timer () to the duration of up to Φ only if $T < T_u + \theta$ (as in node A of Fig. 4) and to the duration of up to $T_{u+1} + \theta - T$ if otherwise (as in nodes B and C of Fig. 4).

Two New Heuristics for Broadcast Trees: We describe two new greedy heuristics–BIPLA (broadcast incremental power with look ahead) and MEN (maximum energy node)–to construct broadcast trees. The first of these (BIPWLA) is an adaptation of the look ahead heuristic proposed by Guha and Khuller [12] for the connected dominating set problem. This heuristic, which also may be viewed as an adaptation of BIPPN, attempts to construct broadcast trees with smaller energy requirement by doing a limited look ahead. The second heuristic (MEN) doesn't explicitly attempt to construct broadcast trees with low energy requirement. Rather, it favors the use of high-energy nodes as relays (non-leaf nodes) of the broadcast tree so as to preserve the energy of low-energy nodes. This strategy is expected to increase lifetime.

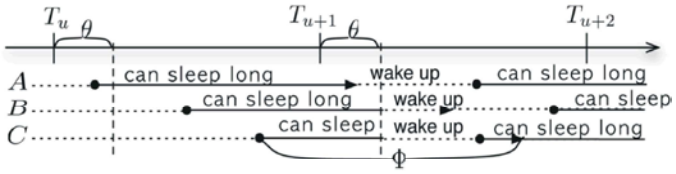

Fig. 4: Sleep mode in X-TESLA

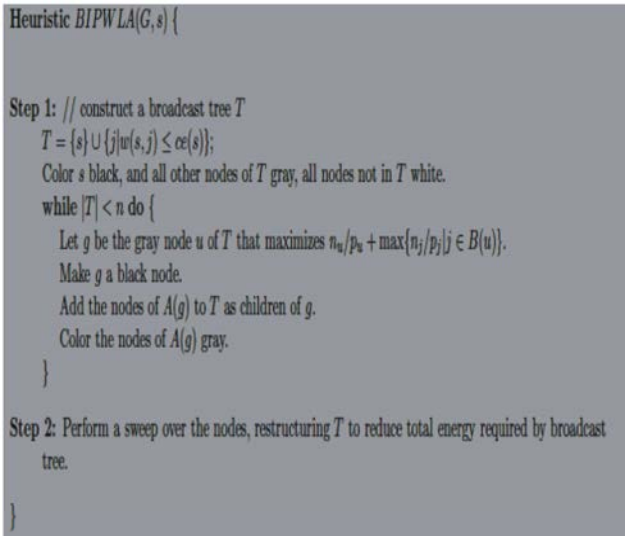

Fig. 5: BIPWLA minimum energy broadcast tree heuristic

The BIPLA Heuristic: Let ce (i) be the current energy at node i of the network before sending a message. In BIPWLA, we begin with a tree T that comprises the source node s together with all neighbors of s that are reachable from s using ce(s) energy. Initially, the source node s is colored black, all other nodes in T are gray and nodes not in T are white. Nodes not in T are added to T in rounds. In a round, one gray node will have its color changed to black and one or more white nodes will be added to T as gray nodes. It will always be the case that a node is gray if it is a leaf of T, it is black iff it is in T but not a leaf, it is white if it is not in T. In each round, we select one of the gray nodes g in T; color g black; and add to T all white neighbors of g that can be reached using ce (g) energy. The selection of g is done in the following manner. For each gray node u _ T, let nu be the number of white neighbors of u reachable from u by a broadcast the uses ce (u) energy. Let p_u be the minimum energy needed to reach these n_u nodes by a broadcast from u. Let,

$A(u) = \{j | w(u,j) \leq ce(u)$ and j is a white node

We see that $n_u = |A(u)|$ and $p_u = \max\{w(u, j) | j \in A(u)\}$.

For each $j \in A(u)$, we define the following analogous quantities.

Node g is selected to be the gray node u of T that maximizes:

$n_u/p_u + max\{n_j/p_j \mid j \_ B(u)\}$

Once the broadcast tree is constructed, a sweep is done to restructure the tree so as to reduce the required energy. The BIPWLA heuristic is summarized in Figure 5.

When T contains all nodes of G(i.e., when T is a broadcast tree). Finally, a sweep is done to restructure the tree so as to reduce the required energy. The MEN heuristic is summarized in Figure 6.

The MEN Heuristic: The MEN (maximum energy node) heuristic attempts to use nodes that have more available energy as non-leaf nodes of the broadcast tree thereby preserving the energy of low-energy nodes, which become,From Q, we select the node u that has maximum energy ce (u). All neighbors j of u not already in T and which satisfy w(u, j) = ce(u) are added to T as children of u. This process of adding nodes to T terminates leaves of the broadcast tree (recall that the leaves of broadcast tree expend no energy in our model). In MEN, we start with T = {s}. At each step, we determine Q such that:

Heuristic $MEN(G,s)$ {

Step 1: // construct a broadcast tree T
$T = \{s\}$;
while $|T| < n$ do {
Let Q be as defined in Equation 1.
Let $u \in Q$ be such that $ce(u)$ is maximum.
$T = T \cup \{$neighbors j of u not in T for which $w(u,j) \leq ce(u)\}$;
}

Step 2: Perform a sweep over the nodes, restructuring T to reduce total energy required by broadcast tree.

}

Fig. 6: MEN minimum energy broadcast tree heuristic

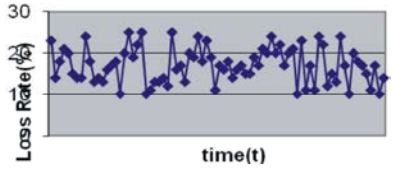

Fig. 7: Loss Rate

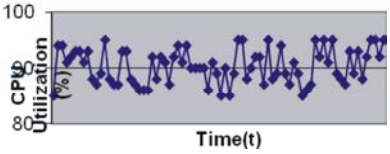

Fig. 8: CPUUtilization

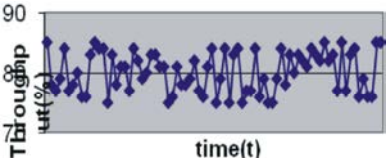

Fig. 9: Throughput

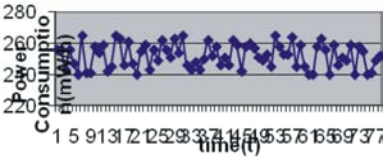

Fig. 10: Power Consumption

$Q = \{u | u$ is a leaf of T and u | has a neighbor $j, j \in T$, for which $w(u, j) \leq ce(u)\}$

significantly by incorporating the critical energy constraint into each minimum-energy broadcast tree heuristic. We have proposed X-TESLA, an efficient scheme which may continue indefinitely and securely, that addresses this and many other issues of the previous schemes. With the advent of more powerful sensor node commodities such as iMote2, the future of public-key technique application to broadcast authentication looks bright, but X-TESLA can efficiently be combined with public-key techniques also. For example, we could modify X-TESLA to use digital signatures on Type 4 packets, keeping everything else the same.

CONCLUSION

Our work shows that although minimum energy broadcast tree heuristics result in low-energy broad- cast trees, the use of these broadcast trees doesn't result in good network lifetime. Network lifetime is enhanced combined with public-key techniques also. For example, we could modify X-TESLA to use digital signatures on Type 4 packets, keeping everything else the same.

REFERENCES

1. Taekyoung kwon, Member, IEEE and Jin Hong "Secure and Efficient Broadcast Authentication in Wireless Sensor Network" IEE Transactions on Computers, 59(8) AUGUST 2010.
2. Akyildiz, I.F., W. Su, Y. Sankarasubramaniam and E. Cayirci, 2002. "ASurvey on Sensor Networks," IEEE Comm. Magazine, 40(8): 102-114.
3. Durresi, A., V. Paruchuri, S. Iyengar and R. Kannan, 2005. "Optimized Broadcast Protocol for Sensor Networks," IEEE Trans. Computers, 54(8): 1013-1024.
4. Luk, M., A. Perrig and B. Willock, 2006. "Seven Cardinal Properties of Sensor Network Broadcast Authentication," Proc. ACM Workshop Security of Ad Hoc and Sensor Networks (SASN).

5. Park J. and S. Sahni, 2005. "Maximum Lifetime Broadcasting in Wireless Networks," IEEE Trans. Computers, 54(9): 1081-1090.
6. Rappaport, T., 1996. Wireless communications: Principles and practices, Prentice Hall.
7. Luk, M., G. Mezzour, A. Perrig and V. Gligor, 2007. "MiniSec: A Secure Sensor Network Communication Architecture," Proc. ACM/IEEE Conf. Information Processing in Sensor Networks (IPSN).
8. Perrig, A., J. Stankovic and D. Wagner, 2004. "Security in Wireless Sensor Networks," Comm. ACM, 47(6): 53-57.
9. Kar, K., A. Krishnamurthy and N. Jaggi, 2006. "Dynamic Node Activation in Networks of Rechargeable Sensors," IEEE/ACM Trans. Networking, 14(1): 15-25.