World Applied Sciences Journal 29 (Computer Sciences, Engineering and Its Applications): 304-308, 2014
ISSN 1818-4952

DOI: 10.5829/idosi.wasj.2014.29.csea.2260

# A Secure and Efficient Authentication System for Distributed Wireless Sensor Network

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

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

**Abstract:** X-TESLA, a new member of the TESLA family uses two levels of chains that have distinct intervals and cross-authenticate each other are used. This allows the short key chains to continue indefinitely and makes new interesting strategies and management methods possible, significantly reducing unnecessary computation and buffer occupation. We propose a technique X-TESLA, can efficiently be combined with public-key techniques. In this, data are splitted into two halves and that data's are encrypted. So that dos attack and other attacks are completely prevented. This encryption is done by RSA algorithm.

**Key words:** X-Tesla · Cross-authentication · Public key technique · Short key chain · RSA algorithm

## INTRODUCTION

A wireless sensor network is a collection of nodes organized into a cooperative network. Sensors integrated into structures, machinery and the environment, coupled with the efficient delivery of sensed information, could provide tremendous benefits to society. Potential benefits include: fewer catastrophic failures, conservation of natural resources, improved manufacturing productivity, improved emergency response and enhanced homeland security [1]. However, barriers to the widespread use of sensors in structures and machines remain. Bundles of lead wires and fiber optic "tails" are subject to breakage and connector failures. Long wire bundles represent a significant installation and long term maintenance cost, limiting the number of sensors that may be deployed and therefore reducing the overall quality of the data reported. Wireless sensing networks can eliminate these costs, easing installation and eliminating connectors.

The ideal wireless sensor is networked and scalable, consumes very little power, is smart and software programmable, capable of fast data acquisition, reliable and accurate over the long term, costs little to purchase and install and requires no real maintenance. Selecting the optimum sensors and wireless communications link requires knowledge of the application and problem definition. Battery life, sensor update rates and size are all major design considerations.

Examples of low data rate sensors include temperature, humidity and peak strain captured passively. Examples of high data rate sensors include strain, acceleration and vibration.

Recent advances have resulted in the ability to integrate sensors, radio communications and digital electronics into a single integrated circuit (IC) package. This capability is enabling networks of very low cost sensors that are able to communicate with each other using low power wireless data routing protocols. A wireless sensor network (WSN) generally consists of a base station (or "gateway") that can communicate with a number of wireless sensors via a radio link. Data is collected at the wireless sensor node, compressed and transmitted to the gateway directly or, if required, uses other wireless sensor nodes to forward data to the gateway. The transmitted data is then presented to the system by the gateway connection.

**Authentication Protocol:** One of the main challenges of securing broadcast communication is source authentication, or enabling receivers of broadcast data to verify that the received data really originates from the claimed source and was not modified en route. This problem is complicated by mutually untrusted receivers and unreliable communication environments where the sender does not retransmit lost packets.

**Corresponding Author:** V. Khanaa, School of Computing Science, Bharath University, Chennai-73, India.

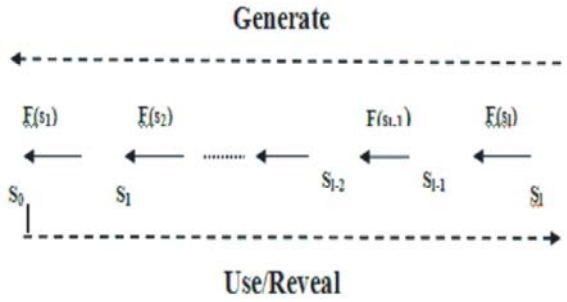


Fig. 1: Example of One-way chain.

**Tesla:** TESLA (Timed Efficient Stream Loss-tolerant Authentication) broadcast authentication protocol, an efficient protocol with low communication and computation overhead, which scales to large numbers of receivers and tolerates packet loss. TESLA is based on loose time synchronization between the sender and the receivers. Despite using purely symmetric cryptographic functions (MAC functions), TESLA achieves asymmetric properties.

**Background and Assumptions:** TESLA requires that the receivers are loosely time synchronized with the sender. TESLA also needs an efficient mechanism to authenticate keys at the receiver. Let us first review one-way chains for this purpose.

**One-Way Chains:** Many protocols need to commit to a sequence of random values. For this purpose, repeatedly use a one way hash function to generate a one-way chain. One way chains are a widely-used cryptographic primitive. One of the first uses of one-way chains was for one time passwords by Lamport [2]. Haller later used the same approach for the S/KEY one-time password system [3]. One-way chains are also used in many other applications.

The sender generates this chain by randomly selecting $s_l$ and repeatedly applying the one-way function F. The sender then reveals the values in the opposite order. Figure 1 shows the one-way chain construction. To generate a chain of length l we randomly pick the last element of the chain $s_l$, so generate the chain by repeatedly applying a one-way function F. Finally, $s_0$ is a commitment to the entire one-way chain and can verify any element of the chain through $s_0$, e.g. to verify that element $s_i$ is indeed the element with index i of the hash chain, we check that

$F^i(s_i) = s_0$.

More generally, $s_i$ commits to $s_j$ if $i < j$ (to verify that $s_j$ is part of the chain if we know that $s_i$ is the ith element of the chain, we check that $F^{j-i}(s_j) = s_i$).

We reveal the elements of the chain in this order $s_0$, $s_1$,. . . , $s_{l-1}$, $s_l$.

How to store this chain? Just either create it all at once and store each element of the chain, or store $s_l$ and compute any other element on demand. In practice, a hybrid approach helps to reduce storage with a small recomputation penalty. Jakobsson [4] and Coppersmith and Jakobsson [5] propose a storage efficient mechanism for one-way chains: a one-way chain with N elements only requires log(N) storage and log(N) computation to access an element. In TESLA, the elements of the one-way chain are keys, so call the chain a one-way key chain. Furthermore, any key of the one-way key chain commits to all following keys, so call such a key a one-way key chain commitment, or simply key chain commitment.

**Time Synchronization:** TESLA does not need the strong time synchronization properties that sophisticated time synchronization protocols provide, but only requires loose time synchronization and that the receiver knows an upper bound on the sender's local time. Let us outline a simple and secure time synchronization protocol that achieves this requirement. For simplicity, assume the clock drift of both sender and receiver is negligible (otherwise the receiver can periodically resynchronize the time with the sender). Denote the real difference between the sender and the receiver's time with $\Delta$. In loose time synchronization, the receiver does not need to know the exact $\Delta$ but only an upper bound on it, which also refer to as the maximum time synchronization error. This approach does not require any extra infrastructure to perform time synchronization. A simple two round time synchronization protocol that satisfies the requirement for TESLA, which is that the receiver knows an upper bound on the sender's clock. Reiter previously describes this protocol [6, 7]. Figure 2 shows a sample time synchronization between the receiver and the sender. In the protocol, the receiver first records its local time $t_R$ and sends a time synchronization request containing a nonce to the sender. 1 Upon receiving the time synchronization request, the sender records its local time $t_S$ and replies with a signed response packet containing $t_S$ and the nonce.

The receiver issues a time synchronization request at time $t_R$, at which time the sender's clock is at time t1. The sender responds to the request at its local time $t_S$. In TESLA, the receiver is only interested in an upper

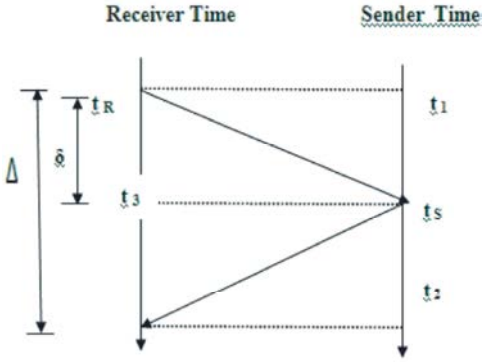


Fig. 2: Direct time synchronization between the sender and the receiver.

bound on the sender's time. When the receiver has its current time $t_r$, it computes the upper bound on the current sender's time as $t_S - t_r - t_R + t_S$. The real synchronization error after this protocol is Δ. The receiver, however, does not know the propagation delay of the time synchronization request packet, so it must assume that the time synchronization error is Δ (or the full round-trip time (RTT)).

**Related Work**

**The TESLA Broadcast Authentication Protocol:** A viable broadcast authentication protocol has the following requirements:

- Low computation overhead for generation and verification of authentication information.
- Low communication overhead.
- Limited buffering required for the sender and the receiver, hence timely authentication for each individual packet.
- Robustness to packet loss.
- Scales to a large number of receivers.

The TESLA protocol meets all these requirements with low cost - and it has the following special requirements:

- The sender and the receivers must be at least loosely time-synchronized.
- Either the receiver or the sender must buffer some messages.

Despite the buffering, TESLA has a low authentication delay. In typical configurations, the authentication delay is on the order of one round-trip delay between the sender and receiver.

**Sketch of TESLA Protocol:** The main ideas behind TESLA. Broadcast authentication requires a source of asymmetry, such that the receivers can only verify the authentication information, but not generate valid authentication information. TESLA uses time for asymmetry. Assume that receivers are all loosely time synchronized with the sender — up to sometime synchronization error Δ, all parties agree on the current time. Here is a sketch of the basic approach:

- The sender attaches a MAC to each packet. The MAC is computed over the content of the packet. For each packet, the sender uses the current key from the one-way chain as a cryptographic key to compute the MAC.
- The sender discloses a key from the one-way chain after some pre-defined time delay
- Each receiver receives the packet. Each receiver knows the schedule for disclosing keys and, since it has an upper bound on the local time at the sender, it can check that the key used to compute the MAC was not yet disclosed by the sender. If it was not, then the receiver buffers the packet. Otherwise the packet is dropped due to inability to authenticate. Note that we do not know for sure whether a "late packet" is a bogus one or simply a delayed packet. Drop the packet because we are unable authenticate it.
- Each receiver checks that the disclosed key belongs to the hash-chain and then checks the correctness of the MAC. If the MAC is correct, the receiver accepts the packet.

**Proposed Work**

**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. 3) 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 [8] provides broadcast authentication, under the assumption that the base station and sensor nodes are loosely time synchronized with a known maximum synchronization discrepancy.

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

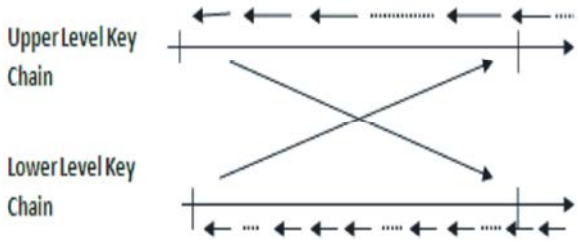


Fig. 3: Basic concept of X-TESLA.

- Multiple distinct keys in the upper level chain authenticate the initial commitment of the next lower level chain repeatedly.
- The repeated authentication will help in resolving problems from DoS attacks, sleeping nodes and idle sessions.

**X-TESLA Details:** An extendable broadcast authentication scheme called X-TESLA, as a new member of the TESLA family, to remedy the fact that previous scheme do not consider problems arising from sleep modes, network failure, idle sessions and to reduce their high cost of countering DOS attacks. In X-TESLA, uses two levels of chains that have distinct intervals and cross-authenticate each other are used. This allows the short key chains to continue indefinitely and makes new interesting strategies and management methods possible, significantly reducing unnecessary computation and buffer occupation.

**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.

**Implementation of X-TESLA using RSA Algorithm:** The proposed work is a new approach which uses the technique of X-TESLA. It can cope with the problems coming from sleep modes, network failures and idle sessions. DoS attacks are resisted without requiring large buffers or strict commitment delivery guarantee, both of which were required for multilevel TESLA. To achieve the goal, the proposed system X-TESLA can efficiently be combined with public-key techniques. In this data are splitted in two halves and that data's are encrypted. So that dos attack and other attacks are minimized. This encryption is done by RSA algorithm. To reduce the size of the packet, the given packet is compressed into a small size compared to original packets. zip compression mechanism is used.

**Constructing the Network:** We are going to construct a network for broadcasting the message in a secure and authenticated way in wireless sensor networks.

**Applying x-tesla:** An extendable broadcast authentication scheme called X-TESLA, as a new member of the TESLA family, to remedy the fact that previous schemes do not consider problems arising from sleep modes, network failures, idle sessions, as well as the time-memory-data tradeoff risk and to reduce their high cost of countering DoS attacks. In X-TESLA, two levels of chains that have distinct intervals and cross-authenticate each other are used. This allows the short key chains to continue indefinitely and makes new interesting strategies and management methods possible, significantly reducing unnecessary computation and buffer occupation and leads to efficient solutions to the raised problems.

**Compression Technique:** This module is used to reduce the size of the packet; the given packet is compressed into a small size compared to original packets. This compression of packets holds two short key chains modes. This mode help us the avoid packet loss or failure at any stage. And we are going to send the data in encrypted format in key chains.

**Scheduling the Sensors in the Networks:** In this module sensors are scheduled for the efficient battery consumption. Here the data's are sent through the active state of the sensors. Sleep node sensors we are going to avoid the buffer overflow by increasing the size of the buffer queue. By using the active state of the sensors commitment distribution messages are passed.

**Creation of Key for the Public Level in Short Key Chains:** In this the public level key is generated. Using that key the two level key chain data's are distributed in the chain for the authorized user. For discovering the authorized user private key will be generated for the each user, using that key broadcasted messages have been delivered.

## CONCLUSION

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. And a enhancement zip compression mechanism is used that is to reduce the size of the packet; the given packet is compressed into a small size compared to original packets. This compression of packets holds two short key chains modes. This mode help us the avoid packet loss or failure at any stage. And we are going to send the data in encrypted format in key chains. And the 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.

## REFERENCES

1. Lewis, F.L., 2004. Wireless Sensor Networks, Smart Environments: Technologies, Protocols and Applications, ed. D.J. Cook and S.K. Das, John Wiley, New York.
2. Lamport, L., 1981. Password authentication with insecure communication. Communications of the ACM, 24(11): 770-772.
3. Haller, N., 1994. The S/Key one-time password system. In Proceedings of the Symposium on Network and Distributed Systems Security, pp: 151-157. Internet Society.
4. Jakobsson, M., 2002. Fractal hash sequence representation and traversal. In Proceedings of the 2002 IEEE International Symposium on Information Theory (ISIT ’02), pp: 437-444.