# Enabling Security in the Transmission Power Grid using Wireless Sensor Networks

Amitabha Ghosh
Department of Electrical Engineering, University of Southern California
Los Angeles, CA 90007.
Email: amitabhg@usc.edu
Chellury Ram Sastry
Siemens Corporate Research, RFID & Sensor Networks Group
Princeton, NJ, 08540.
Email: chellury.sastry@siemens.com

## I. Introduction and Motivation

Wireless sensor networks [1] have inspired tremendous research interest in recent years with a wide variety of potential applications on the horizon. Typical application domains range from environmental conditions and wildlife habitat monitoring; to security surveillance, smart homes and improved health care; to industrial diagnosis and control. Due to the advancement in wireless communications and Micro Electro Mechanical Systems (MEMS), these low-cost, low-power, multi-functional, tiny embedded devices can sense the environment, perform data processing, and communicate with each other over short distances.

With the increasing threat of terrorism around the world, more attention is being paid to the security of the electric transmission infrastructure. Experiences in countries like Colombia, which has faced as much as 200 terrorist attacks on its transmission infrastructure per year, demonstrate the vulnerability of the power system to these kinds of events. Although it is very difficult to avoid or predict when and where these terrorist acts can occur, quick assessment of the situation can help operators to take optimal actions to avoid cascading events and the consequent partial or total blackouts. The mechanical failures resulting from malicious attacks on a transmission line are basically the same as those that would result when extreme natural events affect a portion of the transmission line. Thus, any analysis conducted in this regard can also help in taking preventive and corrective action when acts of sabotage are directed on the transmission infrastructure.

The current method to assess the damage caused by any unexpected physical event on the transmission grid is the visual inspection of the transmission infrastructure. With problems that occur in concentrated environments, like substations or generating plants, it is not difficult to find and assess the damage with a fairly small crew or with adequately localized video surveillance. But in transmission lines which are geographically dispersed over hundreds of miles, this task is more challenging. Any disturbance or dislocation in the electric network (see Figure 1) is sensed primarily by observing the electrical behavior of the power system via the observations and analysis done using the data obtained using measurements of electrical quantities. Thus, once an event occurs, the operator in the control center only receives indication that an electrical fault occurred, but not whether it is temporary or permanent. Therefore, the operator has to try to reinsert the faulted line in order to check the temporary/permanent condition of the event. Once all the attempts fail, then the line is marked as permanently out of service. The recent blackout event of August 14th, 2003 in the U.S. has shown that failure to assess and understand the condition of the power system and delay in taking appropriate corrective actions after just a single outage can lead to widespread blackouts of large areas of the power system.

Along with this development, the increasing popularity and great potential of sensor networks as an emerging technology holds the promise of installation of these embedded networks in the electrical power grid. Detection of mechanical failures in transmission lines such as: conductor failure, tower collapses, hot spots, extreme mechanical conditions, etc can be done using sensor network. In addition, devices such as phaser measurement units and distribution network monitors that are tied to a backbone command network, such as a SCADA system, which can be wireless or wired, exchange critical information for the successful operation of a power system. These measurements can be transmitted using a sensor network to a centralized control system. The main goal is to obtain

a complete physical and electrical picture of the power system in real time, securely, and determine appropriate control measures that could be automatically taken and/or suggested to the system operators once an extreme mechanical condition appears in a transmission line.

Fig. 1. Addition and doubling of points $P$ and $Q$ on the elliptic curve.

However, due to the severe constraints on memory, energy, and communication resources on the sensor nodes, traditional security schemes used in wired networks, such as the Internet, or infrastructure-based wireless network, such as the telecommunication networks, cannot be applied here directly. Moreover, the security requirements could largely vary even across different parts of the same application. Thus, when looking for security requirements of a system, we need to distinguish between three main perspectives. The first one deals with the question of how the attacker reaches the target information system, in our case the power grid: in person, in which case the countermeasures are collectively called physical security, or via the electronic network, in which case the countermeasures are network and system security. The second perspective looks at what fundamental types of threats the system is is to be secured against. Therefore, in order to describe a security solution, we first need to understand the scenario and the threat models. In a sensor network, typically there are three types of communications that take place:

- Base station to the source nodes; e.g. routing beacons, queries, reprogramming the entire network.
- Source nodes to the base station; e.g. sensor readings.
- Between two source nodes; e.g. routing messages, sensor readings for in-network aggregation.

In our scenario, where sensor networks are deployed to monitor events happening in the power system, we focus on the communication between the source nodes and the base station. Hence, in this article we describe a security solution that concerns only the first two types of communications. Specifically, our security requirements are the following:

- *Base Station to Source Nodes*: Each recipient source node should be able to *authenticate* and verify the *integrity* of the messages sent by the base station.
  Authentication is to ensure that the messages were actually sent by the base station and not by some malicious entities, whereas integrity is to ensure that the messages have not been tampered and modified by an adversary on the transit. Traditionally, in two-party communication case, data authentication can be achieved through a purely symmetric mechanism, where the sender and the receiver share a secret key to compute a keyed

Message Authentication Code (MAC) of the message. When a message with a MAC arrives at a receiver node, it can verify the MAC to make sure that the message was originally sent by the claimed sender and not by an adversary. However, this style of authentication cannot be applied to a setting where the base station broadcasts messages to the source nodes, without placing much stronger trust assumptions on the source nodes. This is because if the base station sends authentic data to mutually untrusted source nodes using a symmetric MAC, then any one of the source nodes knows the MAC key, and therefore, could impersonate the base station and forge messages to other source nodes. Hence, we need an asymmetric mechanism to achieve authenticated broadcast.

- *Source Nodes to Base Station*: The messages sent by a source nodes to the base station should be *confidential*, so that no other sources nodes or an eavesdropper can read those messages.
  The standard way to provide data confidentiality is to encrypt it using a secret key which is known only to the intended receiver. This requires establishment of a secure communication channel from the source nodes to the base-station. Moreover, the base station should also be able to check the *integrity* of these messages so as to guarantee that they have not been tampered and modified on the transit. One way to provide data integrity is by using a message digest (also called MAC).

We would also want *scalability* of the network, which means, that when new nodes join the network, the security solution should be able to take care of the new situation seamlessly.

The rest of the article is organized as follows. In Section II, we give a background on the types of cryptosystems and compare their strengths and weaknesses as relevant to sensor networks. In conjunction with that, we also briefly describe Elliptic Curve Cryptography (ECC) [8] and a broadcast authentication protocol called $\mu$-tesla proposed by Perrig et al. []. In Section III, we describe our security solution of how to apply $\mu$-tesla and ECC for providing the security requirements that we described above.

## II. Background

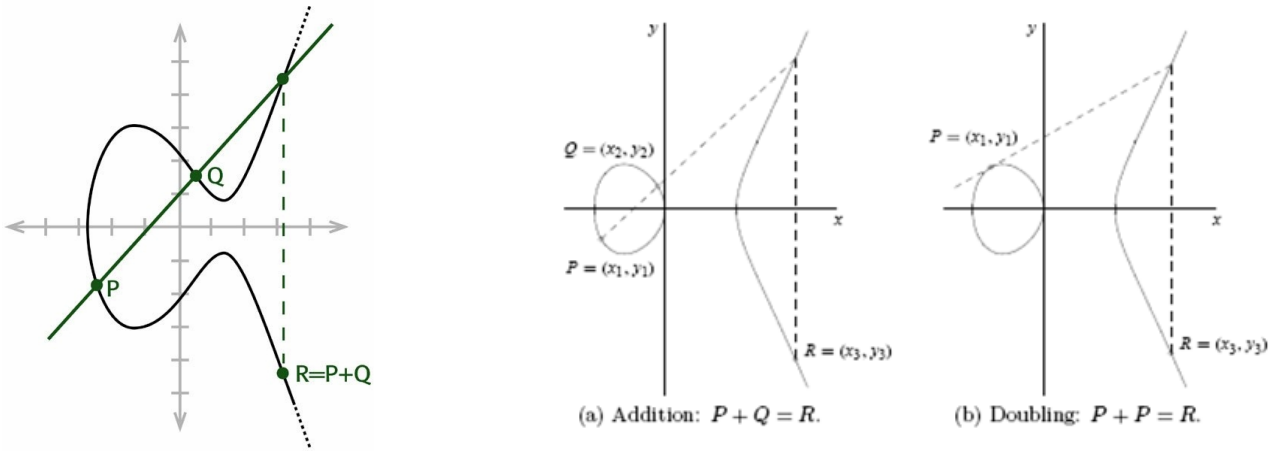
### A. Types of Cryptosystems

The traditional ways to provide network security can be broadly classified into two types: (1) symmetric ciphers, such as RC4, RC5, MD5, DES, AES etc, and (2) public key cryptography, such as RSA, Diffie-Hellman, and the recently proposed Elliptic Curve Cryptography.

Adopting either of these choices for sensor networks poses many challenges. Symmetric ciphers require every transmitter-receiver pair to share a common secret key that is used for both encryption and decryption. Efficient distribution and management of these keys for a large-scale network is the biggest bottle-neck. As the number of nodes could be large, possibly in the order of thousands, and are more likely to be randomly deployed rather than hand-placed, a-priori knowledge on network topology and neighborhood information is unavailable. Moreover, when new nodes join the network, new keys have to be efficiently passed over to this new node so that it can communicate with others.

On the other hand, a public key cryptosystem requires a pair of keys between every transmitter-receiver pair: a public key that is used for encryption, and a private key that is used for decryption. In particular, a transmitter encrypts the messages using the receiver's public key and the receiver decrypts them using its own private key. Since the private keys are never disclosed, nobody else except the intended receivers can decrypt those messages. Two of the most popular public-key cryptosystems, RSA and Diffie-Hellman, are based on the difficulty of factoring large primes and solving the Discrete Logarithm Problem (DLP), respectively. For sufficient security, both RSA and Diffie-Hellman require a key size of at least $1024$ bits. Unfortunately, modular arithmetic with such large key sizes has very high computational overhead when implemented on resource constrained motes. Because of this, implementation of public-key cryptosystems on sensor nodes has traditionally been unfavored.

However, some of the recent implementations of public-key cryptosystems based on ECC, which is also based on DLP, on MICAz and TelosB motes have shown favorable promises [2] [3] [4] [6]. Besides the implementation efficiency of ECC using smaller key sizes for achieving comparable security as RSA or Diffie-Hellman, the best known attacks on ECC-DLP run in time proportional to the square root of the group size of the chosen elliptic curve, such as the Pollard Rho and the BabyStepGiantStep algorithms []. By comparison, much more efficient attacks are known for both RSA and $modp$ discrete log-based cryptosystems. For RSA, the attack goes via. factoring using the Number Field Sieve, and for $modp$ systems it is the index calculus method. An elliptic curve over




(a) (b)

Fig. 2. (a) Adding and doubling points on an elliptic curve.

a 163-bit field currently gives the same level of security as a 1024-bit RSA modulus or Diffie-Hellman prime. The difference becomes even more dramatic as the desired security level increases. For instance, 571-bit ECC is currently equivalent in security to $15,360$-bit RSA or Diffie-Hellman. These schemes provide a cleaner and more flexible interface requiring no key pre-distribution and pair-wise key sharing that are needed in symmetric-key based schemes. Because of these facts, we propose to use ECC in sensor networks.

However, one aspect of being able to use ECC (or any public-key cryptosystem) is that it requires a centralized authority, called the key distribution center (KDC), for storing the public keys of all the nodes in the network. Fortunately, the scenario that we consider requires confidentiality and integrity of the messages sent only from the source nodes to the base station. Therefore, a source node needs to know the public key only of the base station and not that of other source nodes. Hence, as a first step of network initialization we need to find out is a way to securely distribute the public key of the base station to all the source nodes. In this invention disclosure, we use the secure broadcast authentication scheme, called $\mu$-tesla, proposed by Perrig et al. in [], to distribute the public key of the base station to the source nodes, thus obviating the need of a centralized KDC. In the next section, we provide a background on ECC and $\mu$-tesla.

### *B. Elliptic Curve Cryptography*

ECC is based on DLP applied to the Abelian group formed by the points on an elliptic curve over a finite field. The essential security foundation of ECC relies on the absence of a subexponential algorithm for solving the DLP over cryptographic curves. An elliptic curve $E$ over a finite field $\mathbb{F}_p$ (a Galois Field of order $p$) is a smooth curve in the long Weierstrass form:

$$y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6, \tag{1}$$

where $a_i \in \mathbb{F}_p$. The set of points $(x, y) \in \mathbb{F}_p \times \mathbb{F}_p$, along with a point at infinity $\mathcal{O}$, form an Abelian group, denoted by $E(\mathbb{F}_p)$. Here, smooth means that there is no point on $E(\mathbb{F}_p)$ where both the partial derivatives vanish. For large values of $x$, the equation is approximated by $y^2 = x^3$, which can be parameterized as $x = t^2,\ y = t^3$, and one says that *x has degree 2* and *y has degree 3*. The subscripts of the coefficients $a_i$ in equation (1) indicate the degrees that must be given to the coefficients in order to make the equation homogeneous, i.e., each term has the same total degree, which is 6 in this case. Performing the following change of variables:

$$\begin{aligned} x &\longrightarrow x - \frac{a_2}{3} \\ y &\longrightarrow y - \frac{a_1x + a_3}{2} \end{aligned}$$

equation (1) reduces to

$$y^2 = x^3 + ax + b, \tag{2}$$

where $a = \left(\frac{1}{9}a_2^2 + a_4\right) \in \mathbb{F}_p$ and $b = \left(\frac{2}{27}a_2^3 - \frac{1}{3}a_2a_4a_6\right) \in \mathbb{F}_p$, and the smoothness criteria reduces to the determinant $\Delta = -(4a^3 + 27b^2)$ not being equal to 0. The algebraic structure of an Abelian group defined over the set of points in $E(\mathbb{F}_p)$ define cryptosystems based on elliptic curves. The group obeys the following rules:

- Group operation: The group operation is addition $(+)$ and is defined as follows. If $P$ and $Q$ are two distinct points on an elliptic curve then the line $l = \overline{PQ}$ intersects the curve at exactly one point, say $-R$. Then, addition is defined as: $P + Q = R$, where $R$ is the reflection of $-R$ on the $x$-axis.
- Zero element: The point at infinity $\mathcal{O}$ is defined as the zero element. If $P$ is the point $\mathcal{O}$, then $-P$ is also defined as $\mathcal{O}$. For any other point $Q$, $Q + \mathcal{O} = Q$. In $\mathbb{F}_p$ we can visualize $\mathcal{O}$ as sitting infinitely far up the $y$-axis.
- Inverse element: The negative of a point $P = (x, y)$ is defined as $-P = (x, -y)$. If $Q = -P$, then we define $P + Q = \mathcal{O}$.
- Doubling: Let $l$ be the line tangent to the curve at $p$ and let $-R$ be the only (the third) point of intersection of $l$ with the curve; the we define $2P = R$.

Since this is an Abelian group, the group operation is commutative, i.e., $P + Q = Q + P$, and thus the points on an elliptic curve form an Abelian group under addition. We illustrate the operations on points $P, Q, R$ in Figure 2.

### C. $\mu$-Tesla: Broadcast Authentication

Authenticated broadcast requires an asymmetric mechanism, otherwise any compromised receiver can forge messages from the sender. In $\mu$-tesla this asymmetry is achieved through a delayed disclosure of symmetric keys. $\mu$-tesla requires that the base station and nodes are loosely time synchronized, and each node knows an upper bound on the maximum synchronization error. To send an authenticated packet, the base station simply computes a MAC on the packet with a key that is secret at that point in time. When a node gets a packet, it can verify that the corresponding MAC key was not yet disclosed by the base station (based on its loosely synchronized clock, its maximum synchronization error, and the time schedule at which keys are disclosed). Since a receiving node is assured that the MAC key is known only by the base station, it is assured that no adversary could have altered the packet in transit. The node stores the packet in a buffer. At the time of key disclosure, the base station broadcasts the verification key to all receivers. When a node receives the disclosed key, it can easily verify the correctness of the key. If the key is correct, the node can now use it to authenticate the packet stored in its buffer.

## III. Providing Authentication, Data Confidentiality, and Integrity

As we mentioned earlier, the first step of being able to use ECC in sensor networks is to securely distribute the public key of the base station to all the source nodes without using a trusted, and possibly third party owned KDC. Once the source nodes know the public key of the base station, they can encrypt the messages using the public key and send them to the base station, which then can decrypt using its private key. In addition to this, messages sent from the base station have to be authenticated by the source nodes, and hence, an asymmetric scheme is necessary to provide authenticated broadcast. We describe next how both of these goals are achieved using $\mu$-tesla.

### A. Authenticated Broadcast

The base station first generates a sequence of $n$ one-way secret keys (or a key chain of length $n$). It chooses the last key $K_n$ randomly, and generates the remaining $(n-1)$ keys by successively applying a one-way cryptographic hash function $H$ (e.g. MD5) as:

$$K_j = H(K_{j+1}), \qquad 1 \leq j \leq (n-1) \tag{3}$$

Because $H$ is a one-way generating hash function, anybody can compute the keys backward, i.e., compute $K_0, K_1, ..., K_j$ given $K_{j+1}$, but not in the forward direction, i.e., compute $K_{j+1}$ given only $K_0, K_1, ..., K_j$. Once this key chain of length $n$ is generated, the base station divides time into $n$ intervals of equal length and associates

each key with an interval. In time interval $j$, it uses the key of the current interval, $K_j$, to compute the MAC of packets in that interval. Note that, the content of a packet could either be the base station public key, $K_{pub}$, or any other message that it wants to broadcast in the entire network. The base station will then reveal the key $K_j$ after a delay of $\delta$ time intervals after the end of the time interval $j$. This key disclosure time delay $\delta$ is on the order of a few time intervals, as long as it is greater than any reasonable round trip time between the base station and the source nodes. One important property of the one-way key chain is that once a receiver has an authenticated key of the chain, subsequent keys of the chain are self-authenticating. Therefore, to bootstrap $\mu$-tesla, each source node needs to have one authentic key of the one-way key chain as a *commitment* to the entire chain. Also, the source nodes need to be loosely time synchronized with the base station and should know the key disclosure schedule of the one-way key chain.

When a source node receives the packets with the MAC, it needs to ensure that the packet could not have been spoofed by an adversary. The threat is that the adversary already knows the disclosed key of a time interval, and so it could forge the packet since it knows the key used to compute the MAC. Hence, the receiver needs to be sure that the base station did not disclose the key yet which corresponds to an incoming packet, which implies that no adversary could have forged the contents. This is called the *security condition*, which receivers check for all incoming packets. Therefore, the base station and source nodes need to be loosely time synchronized and the source nodes need to know the key disclosure schedule. If the incoming packet satisfies the security condition, the source node stores the packet (it can verify it only once the corresponding key is disclosed). If the security condition is violated (the packet had an unusually long delay), the receiver needs to drop the packet, since an adversary might have altered it.

As soon as the source node receives a key $K_j$ of a previous time interval, it authenticates the key by checking that it matches the last authentic key it knows $K_i$, using a small number of applications of the one-way function $H : K_i = H^{j..i}(K_j)$. If the check is successful, the new key $K_j$ is authentic and the source node can authenticate all packets that were sent within the time intervals $i$ to $j$. It also replaces the stored key $K_i$ with $K_j$.

### *B. Data Confidentiality and Integrity*

Once the base station distributes its public key in an authenticated way to all the source nodes in the network, they can encrypt the message using that key whenever they want to send data to the base station. The base station can use its private key to decrypt those messages. Data integrity can also be provided by computing a keyed MAC (HMAC).

## REFERENCES


[1] Ian F. Akyildiz, WellJan Su, Yogesh Sankarasubramaniam, Erdal Cayirci, A survey on sensor networks, *IEEE Communications Magazine*, vol. 40, no. 8, pp. 102–114, Aug, 2002.

[2] Haodong Wang and Qun Li. Distributed User Access Control in Sensor Networks, *IEEE International Conference on Distributed Computing in Sensor Systems (DCOSS)*, pp. 305–320, San Francisco, CA, June 19-20, 2006.

[3] Haodong Wang, Bo Sheng, and Qun Li. Elliptic Curve Cryptography Based Access Control in Sensor Networks, *International Journal of Security and Networks*, 1(3/4), pp. 127-137, 2006.

[4] Haodong Wang and Qun Li. Efficient Implementation of Public Key Cryptosystems on MICAz and TelosB Motes, *Technical report, College of William & Mary*, WM-CS-2006-7, October, 2006.

[5] Chiu C. Tan and Qun Li. A Robust and Secure RFID-based Pedigree System, *International Conference on Information and Communication Security (ICICS)*, pp. 21–29, Raleigh, NC, Dec. 4-7, 2006.

[6] Haodong Wang, Bo Sheng, and Qun Li. TelosB Implementation of Elliptic Curve Cryptography over Primary Field, *Technical Report, College of william & Mary*, WM-CS-2005-12, October, 2005.

[7] David J. Malan, Matt Welsh, and Michael D. Smith. A Public-key Infrastructure for Key Distribution in TinyOS Based on Elliptic Curve Cryptography, *First Annual IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks (SECON)*, pp. 71–80, October, 2004.

[8] Kristen Lauter. The Advantages of Elliptic Curve Cryptography for Wireless Security, *IEEE Wireless Communications*, vol. 11, no. 1, pp. 62–67, February, 2004.

[9] Adrian Perrig, Robert Szewczyk, Victor Wen, David Culler, J. D. Tygar, SPINS: Security Protocols for Sensor Networks, *Mobile Computing and Networking*, pp. 189–199, 2001.