International Journal of Security and Its Applications
Vol. 10, No. 11 (2016), pp.249-266
http://dx.doi.org/10.14257/ijsia.2016.10.11.21

Periodic Virtual Hierarchy: A Trust Model for Smart Grid Devices

Arezou Moussavi-Khalkhali*[1], Ram Krishnan[2], Mo Jamshidi[3]

[1, 2, 3] Department of Electrical and Computer Engineering, University of Texas-San Antonio, USA
[1]arezou.moussavikhalkhali@utsa.edu,[2]ram.krishnan@utsa.edu,
[3]mo.jamshidi@utsa.edu

Abstract

Authentication among various devices that form a smart grid is a fundamental issue. Due to the large-scale, distributed, and heterogeneous nature of a smart grid, authenticating devices based on their credentials such as secret keys is often not practical. Alternatively, certificate-based trust relationships can facilitate interactions in such scenarios. Two pivotal trust relationship models, hierarchical and peer-to-peer, have been well-researched in the literature. However, the devices in a smart grid do not benefit exclusively from a single trust relationship model owing to the heterogeneous nature of its control structure. In this paper, we propose a periodic hierarchical trust relationship model suitable for real-time applications in a smart grid and robust to the single point of failure problem, which is common in hierarchies. The proposed model deploys a two-layer security authentication mechanism among different devices within those domains in which there are control hierarchies using short-term and long-term certificates, the latter of which is verified periodically.

***Keywords**: Smart Grid Security; Trust Models; Certificate-based Authentication; Trust Chains; Threshold-Scheme Cryptography; Resilient Hierarchical Trust Relationships*

1. Introduction

A Smart grid is a large-scale, distributed, and highly heterogeneous system with a large number of devices from various vendors. To name a few, the smart grid devices range from advanced metering-side devices such as smart meters to distribution-side remote terminal units (RTU) and intelligent electronic devices (IED) such as, fault circuit indicators, transformer monitors, *etc*. to transmission-side devices such as phasor measurement units (PMU) and transmission RTUs/IEDs. The devices often need to communicate with each other over different kinds of networks in order to monitor, control, and regulate the flow of electricity. Certain devices may control a set of devices while others cannot. For critical infrastructures such as a smart grid, providing an attack-resistant network is a fundamental necessity and in particular, authenticated communication is an essential component of any secure network implementation. A fundamental problem for a smart grid operator is to deploy these devices in the field, and manage their control relationships. Using a symmetric key based mechanism in such a context is impractical due to the sheer scale involved in distributing and managing those keys and securing them.

Contrary to symmetric keys, public key certificates can offer a scalable, flexible, and stronger mechanism for authentication of devices in large networks. To facilitate certificate-based authentication, different trust models are developed to

[*] Corresponding Author

manage certificate issuance and revocation. Each of the existing trust models has its own strong and weak points, which make them suited for a part of smart grid. According to the NIST document [11], a smart grid consists of seven domains: generation, transmission, distribution, operations, marketing, service provider, and customer domains, each of which comprises many devices. In many of these domains a hierarchy is established between devices, where a higher level device is often capable of controlling lower level ones in the hierarchy. Of course, not all of the devices in these domains fall within a hierarchy. For the former framework, adoption of a hierarchical trust model resistant to a single point of failure, pertinent to each domain or each part of a domain with control hierarchies is preferred. For the latter scenario, where the relationships of devices do not fall into a hierarchical structure, alternative models like peer-to-peer or a hybrid of peer-to-peer and hierarchical models are appropriate. There are various types of trust models where nodes issue certificates to each other. Hereafter, the term "node" refers to a device or an organization that possess at least a public and a private key pair that can be identified uniquely. An "edge" shows the direction of certificate issuance; *i.e.*, the issuer certificate authority (CA) declares that the ID and the public key of a key pair owner are related to each other by issuing a certificate to the owner. Trust models in this context are well-studied in the literature [1, 10, 12, 16, 17, 22]. However, most fall within 2 categories described briefly in the following subsections.

1.1. Hierarchical Trust Models

A hierarchical trust model is a simple model that is subdivided into several groups. The hierarchy is based on an *m-ary* tree in which each parent node could have at most *m* children nodes. The tree can be either a full tree or a non-full tree based on the application design. Figure 1 shows a strict hierarchy. The term "strict" stems from the fact that all of the edges are in one direction, starting from a superior node and ending in a subordinate node. The root of the tree is called the "Root CA" or the "Trust Anchor" or "TA". The TA has a self-signed certificate, which eliminates the need for a trusted party to sign and vouch for the TA's credentials. The TA usually issues certificates for subordinate (or intermediate) CAs who then can issue certificates for either other subordinate CAs or directly to end entities. In general, hierarchical architectures benefit from straightforward path construction and verification, but the major issue is that they suffer from single points of failure. That is, if the Root CA's private key is compromised, the whole trust structure fails. In fact, compromise of any intermediate level node's private key ends up in the trust failure of that node's subtree.

Variants of hierarchical trust structure is composed of *bidirectional* and *loose* hierarchies. In a bidirectional hierarchy, there is no notion of a Root CA since all the edges are bidirectional. Whereas, in a loose hierarchy [1], the relying party[1] does not need to walk the trust chain up to the Root CA, but once it reaches a common ancestor of itself and the target node, path verification stops. Another variation of a hierarchical structure is a virtual hierarchy proposed by Marchesini and Smith [18]. A virtual hierarchy is a logical hierarchy formed in a peer-to-peer network. Sharing a private key among several CAs, virtual hierarchies address the single point of failure issue in regular hierarchical trust models. This model is discussed in section 2 in detail.

[1] Relying party is the party that needs to verify the validation of a certificate vs the target which owns the certificate to be verified.

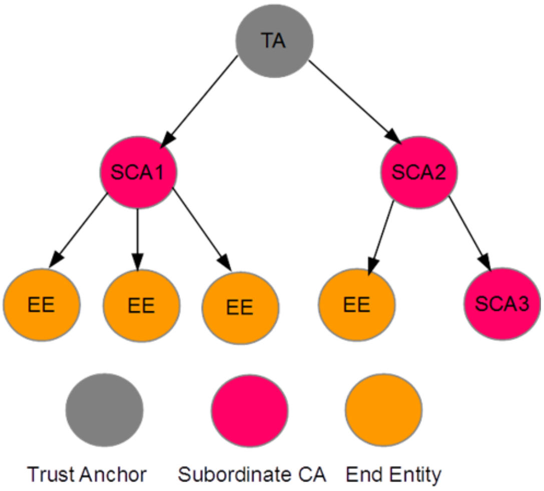

Figure 1. Strict Hierarchy

1.2. Peer-to-Peer or Mesh Trust Models

Peer-to-peer or mesh trust models are other common architecture of trust relationships. In these structures, there is no notion of TA in the sense of the hierarchical model. Each CA can cross-certify any other CA in a graph; as a consequence, formation of a cycle in the graph is inevitable, which makes path construction and verification complex. However, unlike hierarchies mesh structures do not suffer from a single point of failure. Similar to the hierarchies, the mesh structure can have variations: partial mesh and full mesh. In a full mesh model, all nodes may have cross-certified the others but all relationships must be bilateral. In a partial mesh model, unilateral certifications may also exist between two CAs [RFC 5217]. Because of the combination of unilateral and bilateral directions, a trust path is not guaranteed to exist between every pair of nodes in a partial mesh architecture. Figure 2 shows an example mesh architecture.

The rest of the paper is structured as follows. In section 2, related works are surveyed. Section 3 discusses different domains of a smart grid and devices categorized in each domain. Section 4 specifies the periodic virtual hierarchy (PVH) model, which is the proposed trust model for device interaction in a smart grid. A proof of time efficiency is shown in this section. Section 5 discusses the trust relationship suggested in section 4, presents suggestions for future research, and concludes the paper.

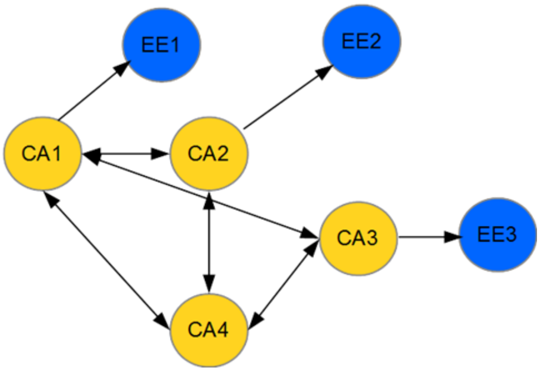

Figure 2. Mesh Architecture

2. Related Work

Simple path discovery and validation are the compelling benefits of hierarchical trust models. However, as mentioned earlier, hierarchical trust models suffer from a single point of failure. There is thus a need for models that will ensure some degree of resiliency against a single node compromise. To this end, a number of resilient hierarchical models have been proposed, which are surveyed bellow.

1. Tzvetkov [25] proposed a scheme in which the trust is spread between two parties: one Issuer CA and several Accredited CAs. These two sets of CAs periodically authenticate each other based on a non- PKI mechanism, such as a biometric authentication method. An Issuer CA receives the users request upon issuing a certificate. At a next authentication period, the Issuer CA sends the user's request to the Accredited CA and if the authentication process between these two CAs is successful, then the Accredited CA signs the public key of the user, sets an attribute, and sends the signed public key in the form of a certificate attribute to the Issuer CA. The Issuer CA sets the added attribute in the extension field of the certificate and distributes it to the user. In addition to signature verification, the certificate validation process is also considered. If one CA (accredited or issuer) is revoked in more than $n/2$ CRLs[1], then that CA is not trusted. (n is the number of total CAs.) This scheme relies on a non-PKI authentication mechanism, e.g. biometric authentication. However, biometric authentication, which is not suitable for machine-to-machine authentication, is not suitable in our context. In addition, two levels of validation, signature validity and the CA's authentication, add to the complexity of the structure.
2. Le *et al*. [15] proposed a key insulated signature scheme to mitigate the damage caused by a CA's private key exposure in hierarchical trust models. In the key insulated method, all CAs have one public key, but each CA has its own private key, which leads to a shorter verification path. Their method offers a scalable hierarchy that is desired also in our application, but relying on utilizing a special secure device called Dev, makes this scheme not suitable for our purpose.
3. Distributing the authority can be considered as a general solution to any single point of failure. Virtual Hierarchies (VH), proposed by Marchesini and Smith [18] is a model in which every two or more CAs make a virtual CA (VCA) and share a private key which is then used as a joint signature of the CA set. Each of the CA sets or VCAs can issue certificates for a certain number of CAs (this number depends on the designer and the efficiency desired from the architecture) to make a collective, which is a single logical node in the virtual hierarchy. The first collective is the Root CA of the virtual hierarchy. If a CA wants to connect to one of the other CA, which itself is connected to a VCA, both CAs must form a new VCA and share a private key; *i.e.*, to connect to any CA which is not a member of a VCA, a new VCA must be established. For instance, after having the pre-defined maximum number of CAs in a collective, any newly added CA must form a new VCA with one of the non-virtual CA members of the collective and share a private key. This procedure is followed to create a simple tree as illustrated in Figure 3, which provides the basis for periodic virtual hierarchy. In this figure the number of directly connected nodes to each VCA is limited to 2.
4. Kim *et al*. [14] used Forward-Secure Signature (FSS) for each issuer CA in the hierarchy to make a disaster-coverable structure. It utilizes the existing PKI infrastructure instead of proposing a new PKI system using new CA

[1] Certificate Revocation List: a list containing revoked certificates

servers and clients. The primary threat that is being addressed here is to mitigate the level of compromise in the event of an intrusion. Braun *et al*. [5], proposed a scheme to avoid breakdown of a hierarchical trust model by using an FSS scheme as well. They also claimed that the time synchronization, which is essential to FSS, is not required in their proposed scheme, which could make it appropriate for smart grid. A few other models exist with or without some loose time synchronization FSS schemes [7, 27, 28]. Models based on FSS schemes are potential candidates for smart grid as well as a multi-party sharing schemes that can be provisioned to ascertain the robustness of the hierarchical structure against node compromise. The application of these two schemes together is left for future research.

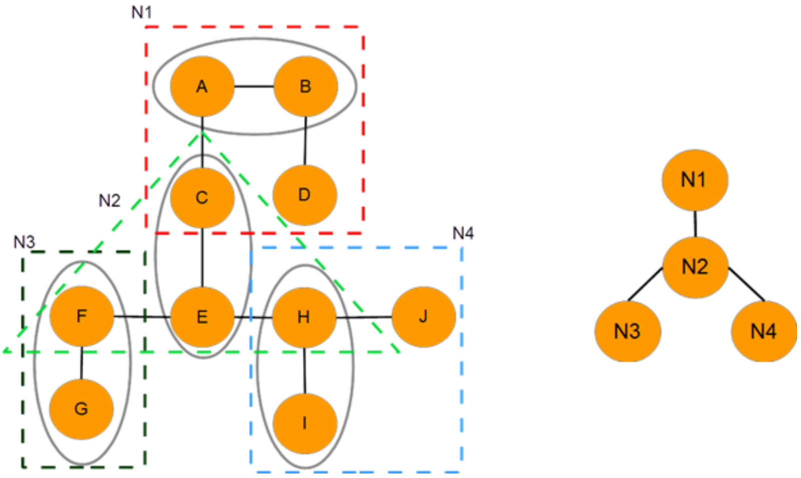

Figure 3. Logical Implementation of a Sample Virtual Hierarchy Model (The Left Figure) and the Respective Virtual Hierarchy That Is Formed (The Right Figure)

3. Smart Grid Domains and Devices

As noted earlier, a smart grid consists of seven domains: generation, transmission, distribution, operations, marketing, service provider, and customer domains, each of which comprises many devices [11]. Forty-nine main actors consisting of devices, individuals, and software are specified for the seven domains. Each of these actors may be helped by peripheral devices to support their functionality, but those are not considered among the actors. As the name implies, the operations domain mostly includes management devices or head-end devices, which are capable of controlling other tools. As mentioned earlier, not all devices fall in a control hierarchy; however, we focus on those that form a control hierarchy in this paper. The specification and functionality of devices, which make them fall in a control hierarchy, also make them suitable to be architected in a hierarchical trust relationship. In order to implement the proposed model devices in a smart grid are grouped into different types based on their functionality and ability to control other devices.

- Type$_n$ - End devices include field devices like level meters, temperature meters, field sensors, measurement devices, relays, environmental monitoring devices, and IEDs [1]. Based on the interactions between devices and functionality of end devices, the definition can change. For example, some

[1] Intelligent Electronic Devices

PLCs[1] and RTUs (Remote Telemetry Unit) can be categorized as field devices, which act as clients or slaves for some other master PLCs [23]. Devices in this group are called "type$_n$" or "t$_n$" devices. End devices in this context are devices which are the users of certificates and they do not have the capability of issuing certificates for lower level devices. However, they are issued certificates to be eligible to communicate with other devices.

- Type$_2$ to Type$_{n-1}$ - Distributed and Local Control Systems consist of collectors, end device controllers (such as some field devices), PLCs, RTUs (Remote Terminal Unit), GIS [2], logging tools (data historian), and intermediate SCADA[3] servers. This part can consist of several levels of devices. Tools in this group are referred to as "type$_2$" to "type$_{n-1}$" or t$_2$ to t$_{n-1}$ devices.
- Type$_1$ - Management level devices comprise head-end devices including GIS, monitoring devices, WAMS[4], SCADA server (MTU[5]), HMIs[6], work stations, alarms, monitoring devices, and databases. These devices are referred to as "type$_1$" or "t$_1$" devices.
- Type$_0$ - There are the communication devices that may fall in between different levels including firewalls, routers, modems, and control network (connects supervisory level devices to lower level control devices). These devices are referred to as "type$_0$" or "t$_0$" devices.

Since a smart grid has a strong control hierarchy component, the virtual hierarchy [18] (section 2) model is a good candidate as it mitigates any single point of failure. However, the model is too general and does not provide implementation details such as suitable crypto schemes in the context of a smart grid. Therefore, it is necessary to tailor virtual hierarchies toward addressing the authentication issues in a smart grid. The next sections discuss our scheme, periodic virtual hierarchy, which benefits from the advantages of virtual hierarchy while enhancing it to be applicable to the domain of trust relationship in the smart grid devices.

4. Applying Periodic Virtual Hierarchy to the Smart Grid Devices

As pointed in previous section, the virtual hierarchy (discussed in section 2) is a theoretical model and hence is not directly usable in application domains such as a smart grid. A number of issues need to be resolved and studied before the model can be utilized. Our contribution is to investigate these issues and propose a concrete periodic virtual hierarchy model for a smart grid.

In order to implement the PVH in a smart grid, first it is necessary to adopt a secret sharing scheme for the purpose of assembling VCAs. The secret sharing scheme allows the members of a VCA to share secrets with some guarantees such as perfect secret sharing (discussed later). Second, some constraints are required to be applied to the devices in order to apply the model to a smart grid infrastructure. The following sections discuss these issues in detail.

4.1. Secret Sharing Schemes Suited to Forming VCAs

Below are some specifications related to secret sharing and multi-party computation that can be used in virtual hierarchies so as to form a virtual CA. The following provides some considerations toward choosing an appropriate scheme to

[1] Programmable Logic Controller
[2] Geographical Information System
[3] Supervisory Control and Data Acquisition
[4] Wide Area Management System
[5] Master Terminal Unit
[6] Human Machine Interface

be put in operation in order to distribute the signing authority among several CAs in a virtual hierarchy.

- There is a need for a method in which all the shares are not required to collaborate in order to reconstruct the secret, in that a device may not function properly, e.g. because of a power outage, or a device being compromised. In addition, adding a share without changing the previous shares is to be considered. Thus, developing a threshold scheme in which at least *t* shares out of *n* total shares, denoted <t,n> where $1< t \leq n$, are able to reconstruct the secret is required.

  The required scheme may also benefit from the Perfect Secret Sharing (PSS) property, where *t-1* shares cannot reveal any information about the secret [4]. Shamir's secret sharing and shared RSA scheme address these requirements; however, the main problem is that these methods rely on a dealer which makes them inappropriate for our trust model. The reason is simply because the dealer is able to forge a signature on behalf of the other shares bringing the single point of failure issue. Furthermore, in Shamir's secret sharing the secret becomes known to at least one player. Nevertheless, the elimination of a trusted party or a dealer has been broadly studied in the literature [8, 21, 26] each of them with pros and cons the consideration of which is beyond the scope of this paper. Interested readers may refer to Beimel [2] to survey secret sharing schemes.
- RSA distributed signature, also known as Shared RSA signature, is another possible scheme for not revealing the secret to any of the players. A robust distributed RSA scheme is considered by Herranz et.al [13]; also, Damgard and Koprowski [9] propose a scheme for distributed RSA without the need for a trusted third party (TTP) or a dealer. Having a TTP is expensive or sometimes not applicable whereas having a dealer among the players reveals the secret to the dealer. Still, to our knowledge this method requires a combiner, a party who needs to collect signed shares.
- Multi-Party Computation (MPC[1]) offers a method where the shares, *i.e.*, inputs to reconstruct the secret, must remain secret. In MPC, shares are secret, but the output can be shown to the parties. In previous schemes, there must be a dealer who knows the secret and is trusted by other players. Here, this dealer could be a TTP or one of the players who wants to participate in the secret sharing process. In MPC, each player acts as a dealer and has its own private key, which obviously must remain secret to others. In our application, each party can calculate the others' shares from their own private key and send the other parties their shares, then delete the private key. Now each of the players has one share of their own and one share of others' private keys. At the end, the players perform a set of simple operations to find the secret. The problem is that all of them will have the result. Therefore, performing the algorithm randomly only at a single device to find the last value could be a good option. Bresson [6] proposes a scheme in which the output of MPC remains secret too.
- Ring signature-based mechanisms (e.g. multi-party concurrent signatures [24]) are able to serve as another possible scheme for VCAs. It allows all the participants to remain anonymous from the outsider view, but insiders or participants are able to distinguish the producer of the signature. However, current schemes are too complicated for utilizing in real-time applications in a smart grid.

[1] MPC with preserving the secrecy of the inputs is called Secure MPC or SMC [6].

- Multi-signatures proposed by Bellare and Neven [3] could be another choice to be employed in the virtual hierarchy in which any subset of players having a public and private key pair can sign a document (e.g. create a certificate) jointly. Their public key is required to verify the signature.

4.2. The Smart Grid Customization of Periodic Virtual Hierarchies

In this section, we discuss configuring a periodic virtual hierarchy model in a smart grid, also discuss implementation issues and its time-efficiency as compared to the standard virtual hierarchy. Since devices that are usually used in a smart grid belong to control hierarchies, some constraints are to be applied to the virtual hierarchy to maintain the control position of these devices. To facilitate these requirements' expression, we break down these devices into three categories for the purpose of our discussion. Categories one and three comprise those devices that are used in the root and in the leaves of the tree. All the other levels in the hierarchy compose the second category. Devices that belong to type$_0$ (section 3), which are mainly communication facilitators, are distributed between the other three categories and it is not necessary to assign an individual category to them. In the following section, the process of applying PVH is discussed in more detail.

4.2.1. Categorizing the Devices in a Smart Grid

The categories based on which the devices are grouped are as follows. Category 1 (C1) consists of type "t_1" devices. Devices in this category form the Root CA in the hierarchy. Tools and systems which are able to control every other device, but cannot be controlled by others, fit into this category. Category 2 (C2) consists of type t_2 to t_{n-1} devices. This category consists of a set of levels that have a control relationship; *i.e.* upper level devices act as collectors or controllers for the lower level devices. Category 3 (C3) devices constitute the leaves of the hierarchy. These devices which are controlled by other devices, but cannot control others, consist of type t_n devices. The following specifications about categories and types of devices must be considered toward setting up the periodic hierarchy in a smart grid.

- A category may consist of one or more types of devices. Category 1 and others each consists of one type of device, but category 2 comprises more than one type of device. (A type is a collection of devices.)
- A device can be of more than one type. For example, a PLC can belong to t_2 that is controlled by another identical PLC in t_1.
- Types constitute totally ordered sets, which means there is a master-slave control relationship between any two devices each from different types.
- Devices belonging to t_1 form the highest layer of the hierarchy which is the Root CA, and those which reside in t_n constitute the leaves of the hierarchy. It is assumed that devices from t_k, where $1\leq k \leq n$-1, are above t_{k+1} in the control hierarchy.

4.2.2. Requirements to Setup a Periodic Virtual Hierarchy in a Smart Grid

As noted earlier, where there is a control hierarchy, a hierarchical trust model can be used. For instance, when there is a SCADA which is usually used in transmission and distribution domains, there is a control hierarchy in these two domains. The management layer of transmission and distribution domains lies in the operations domain. Periodic virtual hierarchy uses the idea of virtual hierarchies in order to address the single point of failure problem in hierarchies and applies it to a smart grid. Below are the requirements to make a hierarchical trust model in a smart grid.

1. Grouping the devices in a VCA must follow the rule that the devices belonging to a single type form a VCA. For example, all data collectors under the control of a transmission SCADA build a VCA, while all PLCs form a different VCA. Also, different devices from the same type can be grouped together. Note that devices in a VCA correspond to an individual type of device; *i.e.*, a VCA cannot comprise devices from both t_1 and t_2.
2. If different devices from the same types, say a PLC and a data collector, form an individual VCA, branches that emanate from that VCA can communicate with each other, whether they are under the control of the PLC or the data collector. As a result, it is necessary to limit the communication among sub-branches with the aid of policy determination issued in their certificate.
3. All the devices in C1 form a single VCA. In other words, in the root level, all devices collaborate to form a single VCA as a Root CA. But there is a limit *n* to the number of nodes that constitute VCAs in C2. However, if required, another VCA can be formed and then all VCAs cross-certify with the Root VCA. In this case, there is just one central VCA that acts as a Root VCA and no node can be connected to other cross-certified VCAs at the management level.
4. C3 and all intermediate CAs which are the last level of a CA hierarchy, *i.e.*, those CAs issuing certificates for end-users and not for other CAs, in each level of the hierarchy contains no VCAs since devices that belong to t_n are just users of certificates and are not able to issue certificates for other devices.
5. Depending on the robustness of the secret sharing algorithm to be used, there might be a limitation in choosing *n*. (Many schemes discussed in section 4.1. are vulnerable to compromising more than *n*/2 out of the *n* total shares.)
6. To each of the VCAs, *m* nodes can be connected directly to form a collective. These directly-connected nodes are called “joining” nodes.
7. All the nodes forming a VCA must be of the same type.
8. Nodes exceeding limit *m* must form a new VCA with one of the joining nodes.
9. On account of not having a VCA in the leaves of the hierarchy, there is no limit to the joining nodes of the devices of C3.

4.3. Implementation of Periodic Virtual Hierarchy in a Smart Grid Domains

The procedure to implement a PVH in domains containing a control hierarchy is as follows. Figure 4 illustrates an example of a proposed hierarchical model in a smart grid.

1. Prior to setting up the tree, each node contains a private and public key pair along with a unique serial number (S/N) and the type of node (t).
2. It is assumed that there is a database, hereafter called the server, which maintains the serial number of legitimate devices associated with each node’s public key and type of device to be added to the trust model. This prevents any rogue node or other out of network nodes to be connected to the network. There may be more than one server to avoid another single point of failure where servers share a key to communicate and synchronize with each other.
3. To establish a VCA all nodes need to check their peers’ S/N and public key with the server prior to running a secret sharing algorithm. This is done by broadcasting the S/N and type (*t*) encrypted with the node’s private key to other peers in a VCA.
4. In the management level, devices from C1 and t_1 share a private and a public key with for example, the distributed threshold RSA scheme without the need for a dealer. These set of devices form a VCA, which is the Root CA.

5. Now, devices from C2 and t_2 are added to the hierarchy. According to the VH structure, they directly apply to the Root CA to get their certificates. They send the certificate request by first sending their encrypted S/N and *t*. The limitation for joining nodes is set to 6 in Figure 4. Therefore, the seventh node must form a new VCA with one of the joining nodes.
6. Given that *n* is the limit of the number of devices forming a VCA, every n device from t1 establishes a VCA according to the requirements mentioned above. In our example in Figure 4, *n* is also set to 6.
7. Depending on how many layers are dedicated for intermediate levels, there will be at most *l* layers in the hierarchy: one layer contains the Root CA, one layer includes the end devices, and the rest make intermediate layers. *l* is the height of the tree.

   Using a <t,n> threshold scheme, for each message exchange between two single nodes in two different VCAs, all *t* devices out of *n* devices forming a VCA have to collaborate to verify and sign messages. This incurs a heavy load and processing time which are critical factors in large real-time networks such as a smart grid. Another problem arises when nodes need to know with which exact node they are communicating. Therefore, there is a need to specify the target node's ID as well as a source node's ID. A two-level certificate issuance procedure is adopted to avoid such issues:

a) Each node applies to its upper VCA to get a certificate, which is called a short-term certificate. If the child node targets a special parent node, then the child node sends its request to that special parent. Otherwise a limit is set for each CA inside the VCA to confine issued certificates by each node to avoid congesting one node. Utilizing short-term certificates resolves the second issue mentioned above as each node has its own certificate.

b) After a VCA forms the last node (in this example the 6th node) it applies to the parent VCA to get a certificate for the whole VCA, called a long-term certificate, along with its own short-term certificate. This will address the aforementioned first problem.

The first time that two nodes from two different VCAs want to communicate with each other, they exchange their long- and short- term certificates. After the relying party verifies the long-term certificate, the rest of the communication including authentication is done by sending the node's short-term certificate, thus, decreasing the process load and verification time. To periodically validate the short-term certificates, and to thwart the consequences of compromising the private keys of individual nodes in a VCA, the nodes periodically send an update message. The update message signed by using a threshold crypto system is sent along with the nodes' long-term certificate jointly with their short-term certificate. This way, the process of one time short-term certificate issuance is imposed on the network, but the overall scheme would incur less processing time and overhead to the network. Proof of why periodic virtual hierarchy takes less time compared with the original virtual hierarchy is given in the next section. Algorithm 1 shows the procedure of implementation of PVH explained above. Managing the CRLs, as well as managing the key revocation lists are beyond the scope of this paper and are left for future works.

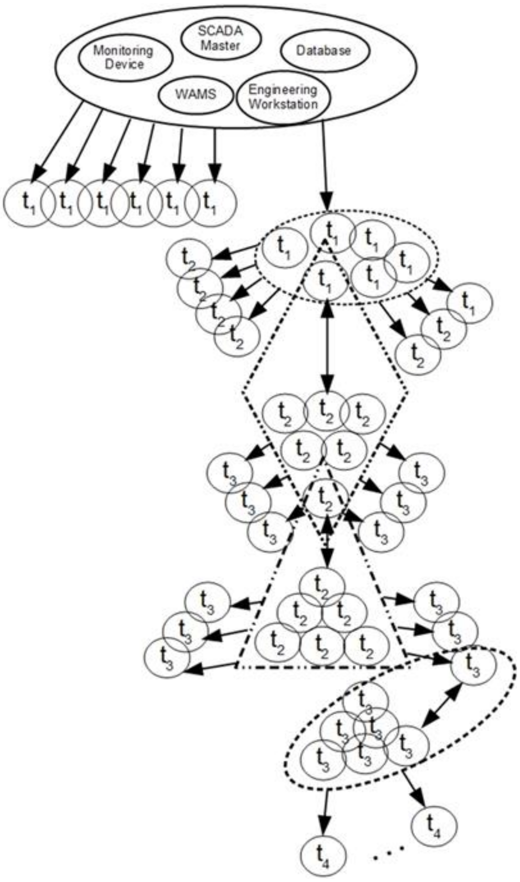

Figure 4. An Example of a Periodic Virtual Hierarchy in a Smart Grid

```
Algorithm 1: Implementation of Periodic Virtual Hierarchy
//Initializing the Root VCA
VCA_ESTABLISHMENT ()
//Initializing the next layers of the hierarchy
current layer = 2
While current layers < $l_{max}$-2 do                    // $l$ is the height of the tree
    While $n$ of the same type (t(k)) < $n_{max}$ do //$n_{max}$ is max number of directly connected nodes
        node $n$ sends (E(S/$N_i$,$t_i$)||request short-term cert) to the server  //E is an encryption function
        the server forwards the request to the parent node in the VCA
        node $n$ gets its short-term certicate after being verifed by its parent node in the above VCA
    End While
VCA establishment() for type(t(k))
//if $n$ < $n_{max}$ for t(k) and a request is sent by a node of type (t(k+1)), a new procedure starts for
    // nodes of type (t(k+1))
the last node of type (t(k)) sends a request to the parent CA to get the VCAs long-term cert
End While
Procedure VCA_ESTABLISHMENT ()
For $i$ = 1 to $n$ //$n$ is the number of nodes allowed in the VCA
        node $i$ broadcasts E(S/$N_i$,$t_i$)
        node $i$ receives E(S/$N_i$,$t_i$) from nodes ≠ $i$
        node $i$ checks the legitimacy of S/N and $t$ with the server
End For
nodes $i$ to $n$ share the private/public key using distributed threshold RSA scheme
Return
```

As one would expect, the leaf nodes do not require the long-term certificates since they do not form VCAs. However, short-term certificates are issued by their parent VCA in order to mitigate the connection of rogue nodes to the network.

4.4. Proof of Time Efficiency of the Periodic Virtual Hierarchy Compared to the Virtual Hierarchy

This section demonstrates the proof of why the periodic virtual hierarchy is more suitable to use in a smart grid in terms of the time required to verify one signature or sign a message than virtual hierarchy. Table 1 shows the terms and variables which are used for this proof. Times are considered in milliseconds. In periodic virtual hierarchies, nodes sign or verify messages using a <t,n > scheme in every T´ time unit where $T_{min} < T´ < T_{max}$.

Table 1. Notations and Terms Used to Prove Time Efficiency of Periodic Virtual Hierarchy Vs. Virtual Hierarchy

m=min number of messages exchanged between two devices; $m \geq 2$: Communications, either between two nodes from the same level or nodes from different levels, start by sending a request and receiving the response to verify the identity of the communicating nodes (the authentication phase). As a result, each node has an incoming request that requires verification and an outgoing message that needs a signature performed by the node.

n= max number of messages exchanged between two devices; no upper limit for messages is set; however, based on the typical functionality of the nodes, n can be set to a certain number to protect the network from a denial of service attack.

A message could be any piece of exchanging information; Intuitively, m and n can be considered as the number of processes performed in each node.

L = maximum number of levels in a tree (hierarchy)

l= height of the tree; $l \in (1,L]$;

T´ >1;

T´ shows the period in which all t out of n nodes need to collaborate in order to sign and verify a message sent with nodes long- and -short-term certificates.

Tv : time taken to verify the signature of a message by a threshold cryptosystem < t,n > in a traditional VH;

$$Tv \in [\underline{Tv};\overline{Tv}]; Tv > 0$$

I) T_{min} = Minimum time taken to verify or sign a message;

$T_{min} > 0$; $m*\min(\underline{Tv},\underline{Ts}) < T_{min} < n * \max(\overline{Tv},\overline{Ts})$

Ts: time taken to sign a message by a threshold cryptosystem < t,n > in a traditional VH;

$Ts \in [\underline{Ts},\overline{Ts}]$; $Ts > 0$

II) T_{max} = Maximum time taken to verify and sign a message by a threshold cryptosystem < t,n > in a traditional VH;

$T_{max} > 0$

$m *(\underline{Tv}+ \underline{Ts}) < T_{max} < n* L*(\overline{Tv} + \overline{Ts})$

Ti :time taken to verify or sign a message by an individual node; $Ti \in [\underline{Ti};\overline{Ti}];Ti > 0$

Since signing or verifying a message by an individual node takes less time than doing the same thing with the same threshold algorithm by t out of n nodes, both signing and verification times are very close to each other and considered as Ti, where $Ti < \min(\underline{Tv},\underline{Ts})$

The superscript prime symbol (´) is used to show the equivalent of the above definitions in the periodic virtual hierarchy

$T'_{min} < T' < T'_{max}$

The following proves that $T'_{min} < T_{min}$ and $T'_{max} < T_{max}$ where,
$T'_{min} = \frac{T_{min}}{T'} + T_i$ and $T'_{max} = \frac{T_{max}}{T'} + T_i$

4.4.1. Proof of $T'_{min} < T_{min}$:

First, we derive the upper bound and lower bound of T'_{min}. From (I) in the table:

$m * \min(\underline{Tv}, \underline{Ts}) < T_{min} < n * \max(\overline{Tv}, \overline{Ts})$

Dividing by $T' > 1$ yields:

$$\frac{m * \min(\underline{Tv}, \underline{Ts})}{T'} < \frac{T_{min}}{T'} < \frac{n * \max(\overline{Tv}, \overline{Ts})}{T'}$$

Adding $Ti > 0$ gains the interval for T'_{min}:

$$\underbrace{\frac{m * \min(\underline{Tv}, \underline{Ts})}{T'} + Ti}_{A} < \underbrace{\frac{T_{min}}{T'} + Ti}_{T'_{min}} < \underbrace{\frac{n * \max(\overline{Tv}, \overline{Ts})}{T'} + Ti}_{B}$$

where A and B are lower bound and upper bound of T'_{min}.

Statement I:

$$\frac{m * \min(\underline{Tv}, \underline{Ts})}{T'} + Ti < m * \min(\underline{Tv}, \underline{Ts})$$

Proof:

$$Ti < \min(\underline{Tv}, \underline{Ts}) \xrightarrow{yields} m * Ti < m * \min(\underline{Tv}, \underline{Ts}) \tag{1}$$

$$Ti < m * Ti \tag{2}$$

$$(1), (2) \xrightarrow{yields} Ti < m * \min(\underline{Tv}, \underline{Ts}) \tag{3}$$

On the other hand:

$$\frac{m * \min(\underline{Tv}, \underline{Ts})}{T'} < m * \min(\underline{Tv}, \underline{Ts}) \tag{4}$$

Adding inequalities (3) and (4) yields:

$$\frac{m * \min(\underline{Tv}, \underline{Ts})}{T'} + Ti < 2m * \min(\underline{Tv}, \underline{Ts}) \xrightarrow{yields}$$
$$\frac{m * \min(\underline{Tv}, \underline{Ts})}{T'} + Ti < m * \min(\underline{Tv}, \underline{Ts})$$

The last expression, which is the result of the proof, shows that the lower bound of T'_{min} is less than the lower bound of T_{min}. The following gives the equivalent proof for the upper bounds.

Statement II:

$$\frac{\mathrm{n} * \max(\overline{Tv},\overline{Ts})}{T'} + Ti < \mathrm{n} * \max(\overline{Tv},\overline{Ts})$$

Proof:

$$Ti < \max\left(\overline{Tv},\overline{Ts}\right) \xrightarrow{yields} n * Ti < n * \max\left(\overline{Tv},\overline{Ts}\right) \quad (5)$$

$$Ti < n * Ti \quad (6)$$

$$(5),(6) \xrightarrow{yields} Ti < \mathrm{n} * \max\left(\overline{Tv},\overline{Ts}\right) \quad (7)$$

On the other hand:

$$\frac{\mathrm{n} * \max(\overline{Tv},\overline{Ts})}{T'} < \mathrm{n} * \max(\overline{Tv},\overline{Ts}) \quad (8)$$

Adding inequalities (7) and (8) yields:

$$\frac{\mathrm{n} * \max\left(\overline{Tv},\overline{Ts}\right)}{T'} + Ti < 2\mathrm{n} * \max\left(\overline{Tv},\overline{Ts}\right) \xrightarrow{yields}$$

$$\frac{\mathrm{n} * \max(\overline{Tv},\overline{Ts})}{\mathrm{T}'} + Ti < \mathrm{n} * \max\left(\overline{Tv},\overline{Ts}\right)$$

4.4.2. Proof of $\mathrm{T}'_{\mathrm{max}} < \mathrm{T}_{\mathrm{max}}$

Again like part 4.4.1, there is a need to derive the upper bound and lower bound of $\mathrm{T}'_{\mathrm{max}}$. From (II) in the table above:

$\mathrm{m}* (\underline{Tv}+\underline{\mathrm{Ts}}) < \mathrm{T}_{\mathrm{max}} < \mathrm{n} *\mathrm{L}*(\overline{\mathrm{Tv}} + \overline{\mathrm{Ts}})$

Dividing by T´ >1 yields:

$$\frac{\mathrm{m} * (\underline{Tv} + \underline{Ts})}{\mathrm{T}'} < \frac{T_{max}}{\mathrm{T}'} < \frac{\mathrm{n} * \mathrm{L}(\overline{Tv} + \overline{Ts})}{\mathrm{T}'}$$

Adding Ti > 0 gains the interval for $\mathrm{T}'_{\mathrm{max}}$:

$$\underbrace{\frac{\mathrm{m}*(\underline{Tv}+\underline{Ts})}{\mathrm{T}'} + Ti}_{\alpha} < \underbrace{\frac{T_{max}}{\mathrm{T}'} + Ti}_{\mathrm{T}'_{\mathrm{max}}} < \underbrace{\frac{\mathrm{n} * \mathrm{L}(\overline{Tv}+\overline{Ts})}{\mathrm{T}'} + Ti}_{\beta}$$

where α and β are lower bound and upper bound of $\mathrm{T}'_{\mathrm{max}}$.

Statement I:

$$\frac{\mathrm{m} * \left(\underline{Tv} + \underline{Ts}\right)}{\mathrm{T}'} + Ti < \mathrm{m} * \left(\underline{Tv} + \underline{Ts}\right)$$

Proof:

$$Ti < \left(\underline{Tv} + \underline{\mathrm{Ts}}\right) \xrightarrow{yields} m * Ti < m * \left(\underline{Tv} + \underline{\mathrm{Ts}}\right) \qquad Ti < m * Ti \quad (10)$$

$$(9),(10) \xrightarrow{\mathrm{yields}} \mathrm{Ti} < \mathrm{m}*\left(\underline{\mathrm{Tv}} + \underline{\mathrm{Ts}}\right) \quad (11)$$

On the other hand:

$$\frac{\mathrm{m}*\left(\underline{Tv}+\underline{Ts}\right)}{\mathrm{T}'} < \mathrm{m} * \left(\underline{Tv} + \underline{Ts}\right) \quad (12)$$

Adding inequalities (11) and (12) yields:

$$\frac{\mathrm{m} * \left(\underline{Tv} + \underline{Ts}\right)}{\mathrm{T}'} + Ti < 2\mathrm{m} * \left(\underline{Tv} + \underline{Ts}\right) \xrightarrow{yields} \frac{\mathrm{m} * \left(\underline{Tv} + \underline{Ts}\right)}{\mathrm{T}'} + Ti < \mathrm{m} * \left(\underline{Tv} + \underline{Ts}\right)$$

The last expression, which is the result of the proof, shows that the lower bound of T'_{max} is less than the lower bound of T_{max}. The proof for the upper bound of T'_{max} follows the same procedure.

5. Discussion and Future Work

A periodic virtual hierarchy for establishing trust relationships among the smart grid devices is proposed based on the virtual hierarchy work of Marchesini and Smith. The model is efficient in terms of time and power consumption and also path verification and validation thanks to its hierarchical structure. Since merely the first step of communications is done by exchanging the long-term certificates, and repeats in T´ intervals, it is assumed that the power consumption related to the processing load will be less. The security of the proposed trust model in terms of credential exchange is based on choosing the period T´, and how often the threshold cryptography is used in order to validate the long-term certificates. Clearly, there is a tradeoff between security, time efficiency, and processing power determined by T´ and the threshold crypto used in the model, which should be taken into account during initial implementaiton by an operator. Some studies conduct the research on implementing the threshold schemes. For instance, Mauland [19] and Nguyen [20] simulate distributed RSA in Python and Java respectively and derive the time taken for the key generation in different rounds with different key sizes. However, the problem with the distributed RSA is that in almost all of the proposed schemes there is a need for a combiner. (Although, the need for a dealer is addressed in many studies discussed in section 4.1.) Implementation of the PVH along with a threshold scheme is considered for future work.

On one hand, robustness against key compromise is provisioned by using a threshold scheme among CA nodes in order to establish a VCA, which also makes the hierarchy resistant to a single point of failure. On the other hand, the efficiency of the periodic virtual hierarchy in a smart grid highly depends on the number of collaborating nodes that form a VCA. Collaborating more nodes to share a private key in a VCA makes the model more robust to key compromise at the cost of losing efficiency in time and process. The reason is using short-term certificates more often than the long-term certificates. Therefore, in the same framework implemented for a smart grid, virtual hierarchies have more resilience to key compromise compared to PVH, but not efficient in time and process load. Considering a high-level view, PVH has two layers of credential protection. The outer layer frequently utilizes long-term certificates at T´ intervals, and the inner layer constantly exploits short-term certificates for communications. T´ is also the factor that determines the resiliency of the PVH vs the VH. Between two consequent T´ (remember that $T' < t < T'+1$), the network is vulnerable to key compromise. The vulnerability is the reason a tradeoff between the resiliency, delay, and process load is required.

Regarding the joining of a malicious party, the proposed method thwarts joining a fake node to the network by utilizing S/N and the type of the node, which are maintained in the server.

During the certificate issuance process, if a child node or a group of child nodes apply to a VCA in their parent collective to get their certificate, the network will stop working properly due to a Denial of Service (DoS). The implemented framework is resistant to the DoS attack as the number of certificates issued by a VCA node is limited to a certain number.

Other than inclusion of timestamps and nonces, including nodes' ID in short-term certificates safeguards the network from impersonation attacks.

In this paper, we have begun an investigation into designing trust relationships among the smart grid devices. In particular, we looked into devices that can form a hierarchical control structure and designed periodic virtual hierarchy as an efficient model for this purpose. A number of future works are to follow. We briefly discuss some open issues. In practice, a smart grid contains a combination of devices with different capabilities; some are more computationally more powerful than others. Hence, public key based solutions do not always apply. Furthermore, in transition from the current grid to a smart grid, one needs to deal with legacy devices that may not be amenable to computationally intensive approaches involving public key cryptography. The model proposed in this paper addresses the domains or parts of a domain in a smart grid that consist of a control hierarchy. Other domains consisting of devices that do not necessarily fall in a control hierarchy can form a bidirectional hierarchy or a peer-to-peer trust relationship without causing loops. To avoid complexity raised by loops, devices can cross-certify each other unless they create a loop. What is a good model that can handle heterogeneous control structures? Is there a difference between threshold crypto and digital signatures that makes one more suitable for VCAs in a smart grid? Is there any better scheme rather than public key based trust relationships based on which devices in a smart grid authenticate each other? How to manage the expired certificates? To answer some of these questions, we plan to implement different signatures and threshold cryptosystems in a test bed simulating a combination of legacy devices and modern smart devices and evaluate the applicability of those schemes for a more realistic smart grid with heterogeneous devices.

References

[1] Adams C, Lloyd S, “Understanding PKI: concepts, standards, and deployment considerations”, Addison-Wesley Professional, (2003).

[2] Beimel A, “Secret-sharing schemes: a survey”. InInternational Conference on Coding and Cryptology. Springer Berlin Heidelberg, pp. 11-46, (2011) May 30.

[3] Bellare M, Neven G, “Multi-signatures in the plain public-key model and a general forking lemma”. Proceedings of the 13th ACM conference on Computer and communications security, pp. 390-399, (2006) Oct 30.

[4] Blakley GR, Kabatianski GA, “On general perfect secret sharing schemes”, Annual International Cryptology Conference, Springer Berlin Heidelberg, pp. 367-371, (1995) Aug 27.

[5] Braun J, Hülsing A, Wiesmaier A, Vigil MA, Buchmann J, “How to avoid the breakdown of public key infrastructures”, European Public Key Infrastructure Workshop, Springer Berlin Heidelberg, pp. 53-68, (2012) Sep 13.

[6] Bresson E, Catalano D, Fazio N, Nicolosi A, Yung M, “Output privacy in secure multiparty computation”, Proceedings of YACC, (2006) Feb 9.

[7] Chow SS, Go HW, Hui LC, Yiu SM, “Multiplicative Forward-Secure Threshold Signature Scheme”, IJ Network Security, 7(3):397-403, (2008) Nov 1.

[8] Chu CK, Tzeng WG, “Optimal resilient threshold GQ signatures”, Information Sciences, 177(8):1834-51, (2007) Apr 15.

[9] Damgård I, Koprowski M, “Practical threshold RSA signatures without a trusted dealer”, International Conference on the Theory and Applications of Cryptographic Techniques, Springer Berlin Heidelberg, pp. 152-165, (2001) May 6.

[10] Dzambasow Y, Joseph S, Nicholas R, Hesse P, Cooper M, “Internet x. 509 public key infrastructure: Certifcation path building", RFC 4158, (2005).

[11] Grid NS, “Guidelines for smart grid cyber security: Vol.1”, smart grid cyber security strategy, architecture, and high-level requirements, (2010).

[12] Henderson M, Coulter R, Dawson E, Okamoto E, “Modelling trust structures for public key infrastructures”, Australasian Conference on Information Security and Privacy, Springer Berlin Heidelberg, pp. 56-70, (2002) Jul 3.

[13] Herranz J, Padro C, Saez G, “Distributed RSA signature schemes for general access structures”, Information Security, Springer, pp 122-136, (2003).

[14] Kim BM, Choi KY, Lee DH, “Disaster coverable PKI model utilizing the existing PKI structure”. On the Move to Meaningful Internet Systems: OTM Workshops, Springer, pp. 537-545, (2006).

[15] Le Z, Ouyang Y, Ford J, Makedon F, “A hierarchical key-insulated signature scheme in the CA trust model”, Information Security, Springer, pp 280-291, (2004).

[16] Linn J, "Trust models and management in public-key infrastructures". RSA Laboratories 20, (2000).

[17] Lloyd S, "Understanding certification path construction", PKI forum white paper, pp. 1-l4, (2002) Sep.

[18] Marchesini J, Smith S, "Virtual hierarchies-an architecture for building and maintaining efficient and resilient trust chains", NORDSEC2002-7th Nordic Workshop on Secure IT Systems, (2002).

[19] Mauland A, "Realizing distributed RSA using secure multiparty computations", PhD thesis, Norwegian University of Science and Technology, (2009).

[20] Nguyen H.L, "RSA Threshold Cryptography", Dept. of Computer Science, University of Bristol, (2005) May 4, Available via http://www.comlab.ox.ac.uk/files/269/Thesis.pdf.

[21] Pedersen TP, "A threshold cryptosystem without a trusted party", Workshop on the Theory and Application of of Cryptographic Techniques, Springer Berlin Heidelberg, pp. 522-526, (1991) Apr 8.

[22] Perlman R, "An overview of PKI trust models". Network, IEEE, 13(6):38-43, (1999).

[23] Stouffer K, Falco J, Scarfone K, "Guide to industrial control systems (ICS) security". NIST special publication, 800(82), pp.16-16, (2011).

[24] Tonien D, Susilo W, Safavi-Naini R, "Multi-party concurrent signatures", International Conference on Information Security, Springer Berlin Heidelberg, pp. 131-145, (2006) Aug 30.

[25] Tzvetkov V, "Disaster coverable PKI model based on Majority Trust principle", Proceedings of IEEE International Conference on Information Technology: Coding and Computing, ITCC 2004, Vol. 2, pp. 118-119, (2004) Apr 5.

[26] Wang X, "A Novel Adaptive Proactive Secret Sharing without a Trusted Party", IACR Cryptology ePrint Archive. 2011:241, (2011).

[27] Yavuz AA, Ning P, "Self-sustaining, efficient and forward-secure cryptographic constructions for unattended wireless sensor networks", Ad Hoc Networks,10(7):1204-20, (2012) Sep 30.

[28] Zhou J, Bao F, Deng R, "Validating digital signatures without TTP's time-stamping and certificate revocation", International Conference on Information Security, Springer Berlin Heidelberg, pp. 96-110, (2003) Oct 1.