Zero-configuration Identity-based Signcryption Scheme for Smart Grid

Hayden K.-H. So, Sammy H.M. Kwok, Edmund Y. Lam and King-Shan Lui
Department of Electrical and Electronic Engineering
The University of Hong Kong
Pokfulam Road, Hong Kong
Email: hso@eee.hku.hk, samkwo@hkusua.hku.hk, elam@eee.hku.hk, kslui@eee.hku.hk

***Abstract*—The success of future intelligent power deliver and transmission systems across the globe relies critically on the availability of a fast, scalable, and most importantly secure communication infrastructure between the energy producers and consumers. One major obstacle to ensure secure communication among various parties in a smart grid network hinges on the technical and implementation difficulties associated with key distribution in such large-scale network with often-time disinterested consumers. This paper proposes the use of an identity-based signcryption (IBS) system to provide a zero-configuration encryption and authentication solution for end-to-end secure communications. The suitability of employing such identity-based cryptosystems in the context of smart grids is studied from the perspective of security requirements, implementation overhead and ease of management. Using the design and implementation experience of our proposed system as an example, we illustrate that IBS is a viable solution to providing a secure and easy-to-deploy solution with close to zero user setup required.**

I. Introduction

Future power deliver and transmission systems across the globe are poised to evolve into highly sophisticated smart grids that rely heavily on information technologies for control and feedback. One key component of these smart grids will be to utilize advanced metering infrastructures (AMIs) that comprise of smart-meters and monitoring sensors installed at customer homes. Smart-appliances, such as an intelligent washer-machine, will then be able to engage in demand response control through real-time communication with the attached smart-meter. Furthermore, distributed monitoring sensors will be responsible of reporting back energy usage and other user demand to various energy producers through layers of data collectors, home gateways and substations. Fig. 1 depicts a conceptual organization of such AMI.

In order to facilitate two-way communications between the consumers and the producers, it is perceivable that a wide spectrum of physical networks will be employed in different segments of the smart grid, ranging from custom-built dedicated wireless radios and power line communication network to commodity public switched telephone network (PSTN), or even the Internet. The highly distributed and the inherently insecure nature of the latter have made them particularly vulnerable to wide-scale, remote, and distributed security attacks. Therefore, security concerns such as consumer privacy, device authentication, as well as data integrity must first be addressed before any higher-level energy policies, such as intelligent pricing strategies, can be successfully implemented and deployed.

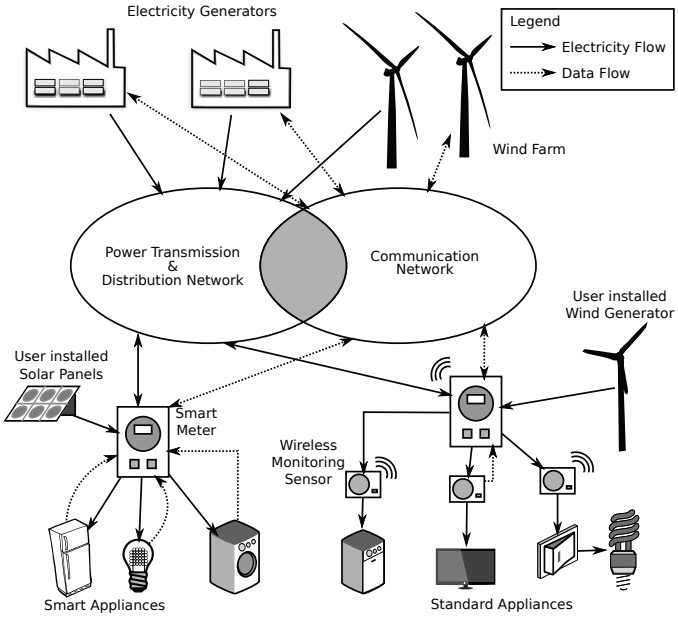

Fig. 1. Conceptual organization of an advanced metering infrastructure (AMI) that relies on a scalable and secure communication network.

Security issues concerning smart grids and AMIs have been studied extensively by a number of security experts [1], [2], [3]. Collectively, it is not difficult to see that given the sheer size and complexity of AMIs, any truly secure system must incorporate appropriate security measures in all levels of the system. It includes the physical security of the smart-meters, which are usually located in insecure facilities, as well as the cyber security of the communication network among the meters and the energy sensors and smart-appliances, among other things.

The focus of this paper is on securing network communications in AMIs. In particular, we propose the use of an identity-based signcryption system to address the security issues of *confidentiality* and *authenticity* in an AMI communication network. Compared to other public-key systems, our proposed system provides scalable and secure communications among smart-meters, smart-appliances and monitoring sensors with-

out requiring complex setup from the users. As the proposed signcryption scheme requires no per-device software setup from the user, we term this scheme a *zero-configuration signcryption scheme*. Because of its very simple key management mechanism, our scheme is particularly suitable for use in smart grids in which secure communications must be provided for use among a large number of devices in the grid.

The rest of the paper is organized as follows. In Section II, background information about identity-based crypto systems and its role in smart grids will first be presented. Details about proposed signcryption scheme will be shown in Section III, followed by results of our initial implementation in Section IV. We discuss security and application considerations in Section V and will conclude the paper in Section VI.

II. Identity-based Cryptography and Smart Grid

A. Identity-Based Cryptography

The concept of identity-based Cryptography (IBC) was first introduced by Shamir in 1984 [4]. In 2001, Boneh and Franklin [5] invented the first feasible solutions for IBC using the Weil pairing on elliptic curves. Since then, many ID-based key agreement protocols and signature schemes using bilinear pairing have been suggested [6].

IBC is a public key cryptosystem in which each user has two keys. When one key is used to encrypt a message, decryption is performed using the other key. In a public key system, one key is published, namely, a *public key*, while the other one is kept as a secret, namely, a *private key*. When Alice wants to send a private message to Bob, she encrypts the message using Bob's public key. Bob then applies his private key to decrypt the encrypted message from Alice.

The main differentiating characteristic of an IBC system is that any agreed upon on and publicly available unique information about a user, such as her email address, can be used to generate the user's public key. As a result, Alice may send encrypted messages to Bob without any prior communication with Bob or any trusted third-party such as a certificate authority (CA). It is the responsibility of the receiver, Bob, to establish authenticated communication with a key-generating server (KGS) to obtain his private key, only if he wishes to decrypt the encrypted messages. Furthermore, Bob may keep his private key for as long as the key is valid without any further communication with the KGS. This feature greatly simplifies the cryptosystem setup and reduces key-exchange data traffic. Furthermore, as we will argue later, this asymmetry in key management is very useful with smart grid systems where a large variety of components with different computing and power requirements are presence.

In contrast, in X.509, and in many other traditional public key systems, a trustworthy certificate authority (CA) is responsible for providing legitimate key information for all communication. It is required because although public keys are not sensitive information, it is necessary to ensure that Alice acquires the real public key of Bob, instead of fake ones provided by Trudy. As a result, before sending any message to Bob, Alice needs to obtain a certificate from CA that contains Bob's key.

In other words, in X.509, it is the sender's responsibility to talk to the KGS, while in IBC, it is the recipient's duty to obtain the necessary key from the KGS. In the following, we would like to demonstrate the advantages of using IBC in smart grid applications. Details of our proposed IBC scheme will be provided in Section III.

B. Utilizing IBC in Smart-Grids

We first study the communication pattern of a smart grid. It is expected that large amount of sensors and measurement devices are used to continuously monitor the energy generation, transmission, and usage. Some data are collected every second or more frequently than other. Sensors would send their data to a nearby *sink node*. For instance, the smart meter installed in a household would be responsible for collecting the data of all the sensors residing in that home. It can be observed that traffic mainly goes from the sensors to the sink node in such scenario. Moreover, sensors and measurement devices are likely to be battery-powered and subject to energy constraint. On the other hand, the sink node, such as the smart meter, does not have energy concern and may connect to the KGS through the Internet directly. Therefore, in smart grid, we should offload the senders of messages as much as possible. As IBC relieves the senders from talking to the KGS, it is a very promising security solution for smart grids.

IBC also allows re-keying, also called key revocation, to be initiated by the sender, which is different from conventional public-key infrastructure (PKI). When Alice wants Bob to use a different pair of public key and private key, she simply encrypts the packet using a new public key of her choice, such as, a key generated by using Bob's ID appended with a timestamp. When Bob receives packets that are encrypted using a new public key, he must obtain the corresponding new private key from the KGS for decryption. This feature allows the measurement devices to issue re-keying based on their individual needs. For example, different devices may take measurements at different frequencies with different levels of on-board buffering. Moreover, to enhance security, the same key should not be used for too many packets. When IBC is used, individual device can determine when to change the key according to its own data and security requirement. Traditional PKI does not offer this flexibility.

III. Proposed Signcryption System

As a proof of concept on the use of IBC in smart grid systems, we have developed a signcryption scheme that is based on the Boneh-Franklin identity-based encryption (IBE) scheme [5] for data encryption and the identity-based signature (IBS) scheme proposed in [7] for authenticating data packets transmitted among devices. As our scheme provides both encryption and signature-based authentication, it is termed as a *signcryption* scheme. In our scheme, we assume each device in the system, such as a smart-meter, has its own unique machine identity number (ID). For instance, the manufacturing serial

number of a smart-meter can be used as its unique machine ID. This ID is in turn used as the identity of the device for all subsequent cryptographic functions.

In our scheme, for encryption purposes, Tate pairing [8] on an elliptic curve E is used to generate the shared secret between the message sender and receiver. Such shared secret is used as the per-packet key for encrypting the data packet. For authentication purposes, the same Tate pairing is used to sign and verify the data packet. Tate pairing was chosen in our scheme because of its relatively low computational cost when compared to other pairing maps. For example, an effective algorithm for calculating Tate pairing was proposed by Miller and was further improved by the works of [9] and [10]. In [11] and [12], a fast formula for the Tate paring computation of supersingular elliptic curve over binary field was proposed. In IBE, it makes use of the bilinearity property of Tate pairing in the calculation of a common secret for packet encryption.

A. Overview

There are two distinct phases in our signcryption scheme. First, a device must register with a central key-generating server (KGS) to obtain its private key if it wants to decrypt message received or sign message to be transmitted during its operation. The KGS holds the master key of the system that is required for generating the private key of a device. Once equipped with its private key, a device may then communicate with any other devices in the smart grid without contacting the KGS again. In this sense, the workload of our KGS is much lower than a certificate authority (CA) of a conventional public-key infrastructure (PKI).

Subsequently, when a device A wants to transmit data to device B, A would encrypt each individual packet with a unique key generated based on B's public key and sign each packet using its own private key. Upon receiving an encrypted packet, B decrypts the encrypted packet using its own private key and verifies the content of the decrypted packet using A's public key. As a proof-of-concept, AES was chosen in our scheme for the encryption of the content of data packets.

B. System Setup

One characteristic of the use of Tate pairing is that the choice of the underlying elliptic curve E affects not only the efficiency of all subsequent computations, but also the security level of our scheme. The communication overhead for encryption and signature is directly proportional to the degree of E. At the same time, the security level of the signcryption scheme increases with the degree of E. Therefore, a trade-off must be made to balance the effects.

In our current implementation, we chose the underlying elliptic curve as a supersingular curve E over $F(2^m)$ with different m to achieve different security levels. Under different choices of m, the security level of the proposed scheme is comparable to an RSA encryption scheme with equivalent key length as shown in Table I.

TABLE I
SECURITY LEVEL OF THE PROPOSED SCHEME COMPARED TO RSA AS A FUNCTION OF m.

m	113	163	233	283
RSA key length (bit)	512	1024	2240	3456

C. Master Key and Device Registration

For each instance of our IBE system, the KGS must first generate a set of system-wide parameters according to the following steps:

STEP i : Select a point P on E.
STEP ii : Generate a field $x \in Z_p^*$ randomly.
STEP iii : Calculate xP.

While x is kept by the KGS as the master key, P and xP are announced to all users as the public parameters for the system.

During the manufacture of a device (e.g. Alice), a pair of device-registration keys, A_DR and Sa_DR, and the system parameters, P and xP are embedded into the device. A_DR is generated by some unique manufacturing ID of Alice (e.g. serial no. of smart meter or MAC address of server), whereas Sa_DR is calculated by KGS using the following equation:

$$Sa_DR \leftarrow x \cdot A_DR \tag{1}$$

Before Alice can communicate with other devices in the smart grid network, she must make registration to KGS through the following procedures:

STEP i : Alice calculates her public key, A, using her device ID.
STEP ii : Alice constructs a packet containing A_DR, A and other registration information, sign the whole packet by Sa_DR to form a digital signature, SIG
STEP iii : Alice sends the packet with SIG to KGS
STEP iv : KGS receives the packet with SIG, verifies SIG using A_DR
STEP v : KGS calculates the private key of Alice, Sa using the following equation:

$$Sa \leftarrow x \cdot A \tag{2}$$

STEP vi : KGS encrypts Sa by A_DR to form Sa'
STEP vii : KGS signs Sa by its private key to form $Sig(Sa)$
STEP viii : KGS sends Sa' and $Sig(Sa)$ to Alice
STEP ix : Alice receives Sa', verifies $Sig(Sa)$ by KGS's public key and then decrypts Sa' by Sa_DR to obtain Sa

After Alice obtains Sa, she can use it for subsequent communications with other devices in the smart grid network. A device-registration key pair is only used for device registration for a single device and KGS will ignore duplicated use of any key pair.

D. Data Packet Transmission

When Alice sends data packets to Bob, the packets are encrypted and signed using the following steps:

STEP i : Calculate the public key of Bob based on his machine ID.

STEP ii : Pick k randomly from Z_p^*.

STEP iii : Calculate kP.

STEP iv : Calculate the shared secret for this packet, s, using the underlying predefined Tate pairing $\tau(\cdot)$ where

$$s \leftarrow \tau(B, xP)^k. \tag{3}$$

STEP v : Convert s to a key of length of 128, 192 or 256 bits depending on the value of m and use it as the per-packet key, K_s.

STEP vi : Encrypt content of the data packet, M, to form M' by an AES block cipher using K_s as the encryption key.

STEP vii : Calculate r where

$$r \leftarrow \tau(P, P)^k. \tag{4}$$

STEP viii : Calculate v, the signature of M, using the equation

$$v \leftarrow h(M, r), \tag{5}$$

where $h(\cdot)$ is an MD5 hash function.

STEP ix : Calculate U with

$$U \leftarrow vS_A + kP. \tag{6}$$

STEP x : Send $X(kP)$, M', $X(U)$, and v to Bob where $X(kP)$ and $X(U)$ are the x-coordinate of kP and U respectively.

E. Data Packet Reception

When Bob receives encrypted data packets with signature from Alice, he takes the following steps to decrypt and verify the packets:

STEP i : Calculate kP based on the received $X(kP)$.

STEP ii : Calculate s using his private key, i.e.

$$s \leftarrow \tau(xB, kP). \tag{7}$$

STEP iii : Convert s to K_s.

STEP iv : Decrypt the received data packet, M' using K_s as the decryption key.

STEP v : Calculate U based on the received $X(U)$.

STEP vi : Calculate r using Alice's public key, where

$$r = \tau(U, P)\tau(A, -x \cdot P)^v. \tag{8}$$

STEP vii : Calculate v' using Equation (5).

STEP viii : Accept the signature if and only if v' is equal to the received v.

TABLE II
PROCESSING TIME FOR A DATA PACKET.

m	113	163	233	283
Encryption (ms)	12	30	64	90
Signing (ms)	2	3	5	6
Decryption (ms)	10	27	49	83
Verification (ms)	22	56	101	170

TABLE III
COMMUNICATION OVERHEAD OF THE PROPOSED SCHEME.

m	113	163	233	283
$X(kP)$ (bit)	113	163	233	283
$X(U)$ (bit)	113	163	233	283
v (bit)	113	128	128	128
Total overhead (byte)	42	57	75	87

IV. IMPLEMENTATION RESULTS

To demonstrate the feasibility of the above methodology, we have implemented the proposed signcryption in software.

A. Implementation Details

The identity-based signcryption scheme was implemented in a 1.6GHz Pentium IV computer under Microsoft Windows environment using Microsoft Visual C++ 6.0 development tool. An open source library called MIRACL was used for implementing all the cryptographic algorithms.

We implemented the proposed scheme using elliptic curves with different values of m and measured the time for processing a data packet. The size of a data packet is 128 bytes. The result is listed in Table II.

B. Communication Bandwidth Overhead

Referring to Section III-D, the communication overhead in the data packet is $X(kP)$, $X(U)$ and v which is proportional to m as shown in Table III.

C. Speed Improvement

The most time-consuming step in the proposed signcryption scheme is the calculation of Tate pairing. To improve the throughput of the signcryption scheme, we propose a key caching scheme to reduce the number of Tate pairing calculations for every data packet.

The proposed key caching scheme works as follows. After Alice encrypts and signs the first data packet sent to Bob, she caches kP, s and r. When she sends the next data packet to Bob, she calculates kP, s and r using the following equations instead of following steps ii-iv and vii in Section III-D:

$$kP \leftarrow kP + kP \tag{9}$$

$$s \leftarrow s^2 \tag{10}$$

$$r \leftarrow r^2. \tag{11}$$

TABLE IV
PROTECTION PROVIDED BY THE PROPOSED SCHEME.

Attack	Threat	Protection provided
Attacker hijacks data sent from smart meters.	Loss of customer privacy.	Meter data are encrypted and no information would be leaked to unauthorized person.
Attacker modifies control commands sent to smart meters.	Smart meters cannot work properly.	Control commands are signed and smart meters would ignore all forged commands.
Customer repudiates metered values sent to power company.	Power company cannot charge the customer.	Meter data are signed with source information and time stamp, making repudiation impossible.

Similarly, Bob caches kP and s after decrypting the first data packet received from Alice. The cached values are subsequently used to calculate the new kP and s for decrypting the next data packet from Alice.

After sending N (a parameter predetermined by Alice) data packets to Bob, Alice resets her key cache and follows step ii- iv and vii in Section III-D again to calculate kP, s and r.

On the other hand, as the term, $\tau(A, -x \cdot P)$ in Equation (8) depends only on the public key of Alice, Bob can also cache it after verifying the first data packet received from Alice and use it for subsequently verifying data packets sent from Alice.

With the proposed key caching scheme, the total number of Tate pairing calculations for transmitting and receiving a data packet can be reduced to zero and one respectively and the corresponding processing time for encrypting, signing, decrypting and verifying a data packet with $m = 113$ can be reduced to 16μS, 0.6mS, 16μS and 12mS respectively.

Note that this proposed key caching scheme has intrinsic capability for solving the synchronization problem. That is, if Bob fails to receive any of the data packets from Alice, the cached s cannot be used to calculate the new s for the decryption of the next data packet from Alice. However, Bob can still use the received $X(kP)$, $X(U)$ and v and follows all steps in Section III-E to decrypt and verify the next data packet and resynchronize the key exchange between Alice and him.

Another simple way for speed improvement is to use per-session key instead of per-packet key in the encryption of packets.

V. DISCUSSIONS

A. Security Consideration

The proposed signcryption scheme can provide end-to-end encryption, source authentication and message integrity for data sent between devices in a smart grid. Thus, it can protect against not only passive attacks (eavesdropping and sniffing data as it passes over the grid) but also active attacks (altering data and masquerading as another individual to send data over the grid). Example of attacks and the protections provided by the proposed signcryption scheme is shown in Table IV.

Other security considerations are:

1) Chosen-Plaintext Attack: As any user in the system can use the signcryptor to encrypt any chosen plaintext, the system is subject to chosen-plaintext attack in which an attacker can choose the plaintext that gets encrypted and obtain the corresponding ciphertext from the output of the encryptor. However, as an AES cipher is used for the encryption, it is unlikely to discover the key for the encryption by simply analyzing the plaintext-ciphertext pairs. In addition, each data packet is encrypted by a unique per-packet key. Therefore, the security of the system will not be seriously affected even if one of the keys is discovered by the attacker.

2) Key Escrow: As the KGS is in possession of the master secret, x, it encompasses the full knowledge of private keys of all devices, allowing it to decrypt any message sent to any device or impersonate any device to sign any message sent to others. There are two ways to reduce the risk of breaking the entire IBC system owing to the compromise of the KGS; first, by using distributed key generating servers, and second, by using short-lived master key.

In the first method, x is split into two or more parts. Each part, x_i, is then kept independently by a different key generating server, KGS$_i$. When a device, such as Alice registers with the system, she must approach each KGS independently. Each KGS will then return a partial private key as well as x_iP to her after verifying her identity. Once equipped with such information, Alice may then calculate her true private key, xA as well as xP. Since each KGS$_i$ possesses only x_i, no individual KGS can calculate the private key of any device unless all KGS conspire to do so, which also reduces the risk of compromising x if any one KGS is compromised.

The second method to lower the chance of compromising the master secret key, x is by employing a short-lived master key. In this case, KGS changes the value of x at a regular interval. With each new master key, private keys for all devices are also updated. The details of the key update scheme is presented in Section V-A4.

3) Key revocation: KGS maintains a key revocation table containing IDs of all devices whose keys have been revoked. Whenever the private key of an individual device (Bob) is lost or compromised, Bob can no longer use his private key which was generated by KGS based on his ID. Thus, Bob needs to report the case to KGS, change his device ID and register to KGS. Upon the request from Bob, KGS issues to Bob a new private key based on his new ID, adds the old ID of Bob into the key revocation table and broadcasts the table to all devices to notify them the update of the table.

During normal communication between devices, when a device (Alice) needs to communicate with another device (Bob), Alice first checks whether the ID of Bob appears in the key revocation table. If Bob's ID is found, Alice realizes that Bob's key has been revoked and so she will cease the communication with Bob.

4) Key Update: To further enhance the security of the system, the master key of the system, x should be updated regularly. To achieve this, the following key update scheme is

TABLE V
KEY UPDATE PROCESSING TIME.

m	113	163	233	283
Parameter generation time	0.55ms	0.85ms	1.5ms	2.3ms
Private key generation time	0.55ms	0.85ms	1.5ms	2.3ms

proposed. Each master key, x, can only be used for certain period of time. Upon expiry of x, KGS generates a new key, x', and then calculates a new $x'P$ and the private keys of all devices in the system. Finally, KGS encrypts the new private key of each device by the device's old public key and then sends the encrypted new private key to each device. Upon receiving the encrypted private key, each device decrypts the received key using its old private key and then obtains the new private key.

Table V tabulates the time requirement for calculating the new $x'P$ and the private keys of each device.

After generating the private keys of all smart meters, KGS needs to distribute the private key to each smart meter. As the total number of smart meters in a smart grid may be huge, it is not possible to distribute the keys to all smart meters simultaneously. It is proposed to have a grace period in which both the expired key and the new key can coexist. During the grace period, if Alice wants to communicate with Bob, she first asks Bob if he has already updated his key. If not, Alice will only use the old keys in the communication with Bob.

B. Application of the proposed scheme

Using different m, the proposed scheme can be used in different applications for smart grid. For example, in a transmission grid, many sensors deployed in overhead transmission lines may sample signals at 50Hz. If the proposed signcryption scheme with $m = 113$ and the key caching scheme as described in Section IV-C is used, the time for processing one data package to be sent from a sensor is only about 12ms and so the signcryption scheme is fast enough to handle the traffic in the transmission grid. On the other hand, in distribution grid, smart meters report energy usage at much lower frequency. However, information such as billing information to be transmitted in the grid is so important that tampering with such information may lead to a great money loss to the power company. In such case, we can choose a greater value of m with a view to increasing the security level of the signcryption algorithm.

VI. CONCLUSION

In this paper, a zero-configuration identity-based signcryption scheme is presented to illustrate security capabilities enabled by identity-based cryptosystems that are suitable for smart grid systems. The machine identity number (ID) of a device connected in a smart grid is used to generate unique keys to encrypt and sign each individual data packet sent among devices in the grid. IBC allows low power devices such as sensor nodes to transmit encrypted data to data collectors without ever contacting the key server, thereby greatly simplifying system setup without compromising security. Furthermore, since all information required to generate encryption keys is available with the sender, no communication with centralized key servers is needed during normal data transmission phases. Finally, since a new per-packet key is generated for each individual data packet, the risk of key hijacking is virtually eliminated.

The proposed signcryption scheme has been implemented in software and the initial implementation results indicate that it can work effectively to provide strong encryption and authentication for data packets sent among devices in a smart grid. Furthermore, using a key-caching scheme, high data throughput can be achieved. Because of the scalability of the signcryption scheme, it can be used flexibly to provide different security levels at different implementation costs that can fit different applications in smart grids.

ACKNOWLEDGMENT

This work was supported in part by the Research Grant Council of Hong Kong, project HKU 716408E.

REFERENCES

[1] S. McLaughlin, D. Podkuiko, and P. McDaniel, "Energy theft in the advanced metering infrastructure," in *4th International Workshop on Critical Information Infrastructure Security*, September 2009.

[2] H. Khurana, M. Hadley, N. Lu, and D. A. Frincke, "Smart-grid security issues," *IEEE Security and Privacy*, vol. 8, pp. 81–85, 2010.

[3] F. Cleveland, "Cyber security issues for advanced metering infrasttructure (ami)," in *Power and Energy Society General Meeting - Conversion and Delivery of Electrical Energy in the 21st Century, 2008 IEEE*, 20-24 2008, pp. 1 –5.

[4] A. Shamir, "Identity-based cryptosystems and signature schemes," in *Proceedings of CRYPTO 84 on Advances in cryptology*. New York, NY, USA: Springer-Verlag New York, Inc., 1985, pp. 47–53.

[5] D. Boneh and M. K. Franklin, "Identity-based encryption from the weil pairing," in *CRYPTO '01: Proceedings of the 21st Annual International Cryptology Conference on Advances in Cryptology*. London, UK: Springer-Verlag, 2001, pp. 213–229.

[6] L. Martin, *Introduction to Identity-Based Encryption*. Artech House Publishers, 2008.

[7] F. Hess, "Efficient identity based signature schemes based on pairings," in *9th Annual International Workshop on Selected Areas in Cryptography*, 2002, pp. 310–324.

[8] G. Frey, M. Müller, and H.-G. Rück, "The tate pairing and the discrete logarithm applied to elliptic curve cryptosystems," *IEEE Transactions on Information Theory*, vol. 45, no. 5, pp. 1717–1719, July 1999.

[9] P. S. Barreto, H. Y. Kim, B. Lynn, and M. Scott, "Efficient algorithms for pairing-based cryptosystems," in *CRYPTO '02: Proceedings of the 22nd Annual International Cryptology Conference on Advances in Cryptology*, 2002, pp. 354–368.

[10] S. D. Galbraith, K. Harrison, and D. Soldera, "Implementing the tate pairing," in *ANTS-V: Proceedings of the 5th International Symposium on Algorithmic Number Theory*, 2002, pp. 324–337.

[11] P. S. Barreto, S. D. Galbraith, C. O. Héigeartaigh, and M. Scott, "Efficient pairing computation on supersingular abelian varieties," *Designs, Codes and Cryptography*, vol. 42, no. 3, pp. 239–271, 2007.

[12] S. Kwon, "Efficient tate pairing computation for elliptic curves over binary fields," in *Lecture Notes in Computer Science volume 3574: Information Security and Privacy*, 2005, pp. 134–145.