A LIGHTWEIGHT MECHANISM FOR MUTUAL AUTHENTICATION IN SMART GRID

Mohammad Hossein Yaghmaee

Mashhad Electric Energy Distribution Company (MEEDC), Iran
yaghmaee@ieee.org

Fatemeh Naji Mohades

Mashhad Electric Energy Distribution Company (MEEDC), Iran
f.mohades@imamreza.ac.ir

ABSTRACT

Advanced Meter Infrastructure (AMI) is a critical component of smart grid, which, if compromised has serious impacts on the safety of utility and consumers. Message authentication is a serious problem and each message should authenticate and receiver checks that message come from a real sender and has no forgery during the transmission. For achieving this major we design two protocols, in first protocol we initially process with mutual authentication between sender and receiver then in second protocol messages between them are authenticate. Without authentication; an attacker can modify the message, forge a new message, or replay an old message to do the malicious operation. The current solutions for authentication like, traditional public key based digital signatures like RSA have heavy computation and are not suitable for resource constraint devices like smart meter.
In this paper, we exploit a mathematical problem called balls and bins algorithm in randomized algorithms topic and Elliptic Curve discrete logarithm problem (ECDLP) for generation and transmission of our parameter.
Key words: smart grid, Mutual authentication, message authentication, BiBa

I. INTRODUCTION

The Smart Grid (SG) motivation, aims at leading the current traditional power grid to set of new technologies and services that will make the electricity networks more reliable, efficient and secure with a two-way communication of electricity and information, creating a widely distributed energy delivery network.
The core of a smart grid is a key part named Advanced Metering Infrastructure (AMI). AMI consists of a set of hardware, software and communication network. The important elements are a Meter Data Management Server (MDMS), a Data Collector Unit (DCU), Smart Meter (SM), and hierarchical network like Home Area Network (HAN), Building Area Network (BAN), and Neighbor Area Network (NAN) [1].
The current solution for authentication in smart grids consists traditional public key based digital signatures like RSA [2], one-time signature (OTS) proposed by Kgwadi *et al*. [3], hash based message authentication code (HMAC) methods that use of Diffie-Hellman technique for key agreement process by Fouda *et al*. [4], and elliptic curve cryptography (ECC) -based method that use of "elliptic curve discrete logarithm problem" or ECDLP[9-11]. These solutions almost attend to smart grid restriction like low power smart meter but still public key computation on some of them are not suitable for AMI environment. Another downside with this method is the lack of consideration of mutual authentication between devices [5]. For this reason in the current study, we utilize a different method for generating a signature named birthday problem.
Birthday paradox [6]: The birthday problem asks, how many people must there be in a room for there to be at least a 50% chance that two of them were born on the same day of the year (assuming birthdays are distributed evenly)? Or how many balls must throw at bins to expect at least a birthday collision? Let X_{ij}be indicator random variable to count the number of collisions in birthday for a person j be at the day i or ball j going into the bin i, and m as people or balls, n as days or bins, then we have:

$$E(X) = \sum_{i \neq j} \Pr[X_{ij} = 1] = \frac{1}{n}\binom{m}{2} \qquad (1)$$

$$E(X) = \frac{1}{365}\binom{m}{2} = 1 \text{ we get m } \geq 23$$

Specially, in the proposed scheme, the smart meters, which are placed on users 'premises and the MDMS, which is placed in the utility of the SG can first achieve mutual authentication and then, the subsequent messages can be authenticated in a lightweight way. Detailed security analysis shows that the proposed schema can satisfy the valuable security requirements of SG communications.
Similar to [5] we consider that the DCU is a non-intelligent messenger by simply bypassing data to MDMS and, assume communication between smart meters and MDMS.
In this paper, we present two protocols. The first one is a mutual authentication schema between two entities in AMI like a SM and MDMS. Second one, is a light weight massage authentication method for securing communication between them.

II. RELATED RESEARCH WORK

A. The BiBa

The Bins and Balls (BiBa) was proposed by Perring [7], which is a signature scheme with a low verification overhead and small signature size. BiBa is one of the fastest signature schemas, but the disadvantage of BiBa is public key size. In [3] Kgwadi looks into an authentication algorithm for the SG and shows that

BiBa has a 75% smaller cost than Elliptic Curve Digital Signature Algorithm (ECDSA). BiBa uses one-way function without trapdoors based birthday paradox, and utilizes of finding hash value collisions of SElf-Authenticating vaLues (SEALs). The property for SEALs is that the verifier can efficiently authenticate the SEAL based on the public key, and that it is computationally infeasible for an adversary to find a valid SEAL given a public key. The security of this method is based on that an adversary has less SEALs than verifier so he/she has a low probability to forge a signature. We explain the BiBa schemas in a simplest way:

Key generation and signing

Signer generates t random string named SEALs, forms h=H (m), picks G_h from hash family function G, applies G_h on all SEALs to find one collision where $G_h(S_i) = G_h(S_j)$, then S_i and S_j forms the signature. After that, sends message m and signatures $\{S_i, S_j\}$ to verifier.

Verification

Verifier first should verify the SEALs, hence checks $S_i \neq S_j$ and authenticate them in an efficient way like a Merkle tree, after that computes h=H (m) and checks $G_h(S_i) = G_h(S_j)$.

III. OUR SCHEMA

We consider a mutual authentication scheme between communicating elements such as the smart meter and the MDMS utilizing a pre-shared password and the message authentication for secure communication. These are required to prevent the various security threats possibly happened in the smart grid environment like massage forgery. Similar to the *X.1035* standard [8], we define K= ($ID_A|ID_B|PW$). Furthermore, we assume that both parties have knowledge of the EC parameters set and hash function. Table I presents the list of parameters and their description used in our schema.

Table I: NOTATIONS LIST

Notation	Description
|	The string concatenation operation
×	An elliptic curve scalar multiplication
H ()	Secure hash functions in the random oracle model
Gp	Cyclic group of prime order n of P
P	Large prime generator of group
n	Order of elliptic curve
H_1 ()	Secure one-way hash function H1: $\{0,1 \rightarrow Z_p^*\}$
T	The time stamp
PW	Pre shared key between SM and server
G	Hash function family in the random oracle model
G_h	$\{\mathbf{0},\mathbf{1}\}^{m2} \rightarrow [\mathbf{0}, \boldsymbol{n}-\mathbf{1}]$ is an instance in the hash function family G selected with an indicator h
Zp	Finite field of order p with integer

A. Description of Mutual Authentication Protocol

As shown in figure 1, the protocol has the following steps:

Step1) SM initiates a mutual authentication process with sending a request to MDMS as the first packet $\{ID_{sm}, Sig, Sn=1\}$, Sn is the sequence number of packet. Firstly, SM generates t SEALs (S_1, S2… S_t), and computes mask for timestamp as $t_{mask\ 1}=H_1(T_1)$. Then utilizes initial password PW shared with MDMS for calculating $Q= (PW|\ t_{mask\ 1}) \times P$, where $Q= (Q^x, Q^y)$, Q^x and Q^y are the coordinates of Point Q. It sends parameters Q^x and t_{mask1} to Algorithm 1 to find a possible signature. C in Algorithm 1 is a counter. If once could not find signature then increments C and recomposes h. When Algorithm 1 finds a 3-way collision for h where $i \neq j \neq k$ then it forms the first packet and send it to the server.

Algorithm 1

1. for c←1 to 100 do
2. Compute $h=H(Q^x | t_{mask} | C)$
3. Pick G_h from Hash Function Family G
4. for i← 0 down to t -1 do
5. for j← 0 down to t -1 do
6. for k← 0 down to t -1 do
7. if $G_h(S_i) = G_h(S_j) = G_h(S_k)$
8. Return sig= {Si, Sj, S_k, T, C}
9. else increments C

Step2) In this step, first the MDMS via ID_{sm}, picks PW from database and computes, $t_{mask1}=H_1(T_1)$, $Q= (PW|\ t_{mask\ 1}) \times P$, and obtains Q^x. Then it performs $h=H(Q^x |t_{mask1}|C)$, checks $S_i \neq S_j \neq S_k$ and $G_h(S_j)$ is equal with $G_h(S_j)$ and $G_h(S_k)$ then MDMS can authenticate the identity of the SM. Subsequently MDMS should proof its identity to SM because it is a mutual authentication and if MDMS do not authenticate for SM, impersonate server attack can happen. Hence MDMS computes $t_{mask2}=H_1(T_2)$, $Q= (PW|\ t_{mask2})$ and sends parameter t_{mask2} and Q^x to Algorithm1 to achieve a signature. After that it performs second packet $\{ID_{MDMS}, Sig, Sn=2\}$ and sends it to SM.

Step3) SM receives the second packet and verifies the signature in the packet. Hence it picks C and T_2 from signature set and composes $t_{mask2}=H_1(T_2)$, $Q= (PW|\ t_{mask2}) \times P$ after that build $h=H(Q^x | t_{mask2} | C)$. Then computes $G_h(S_i)$ for all S_i in signature set, if all $G_h(S_i)$ in signature set are equal where $i \neq j \neq k$ then the identity of MDMS is authenticate for SM.

After that three steps process between SM and MDMS to authenticate each other is done. But still we need message authentication method to assurance of message integrity.

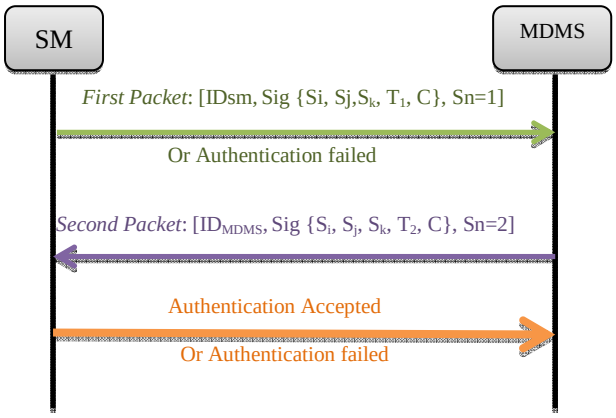

Figure 1: Mutual authentication protocol

B. Description of Message authentication Protocol

Phase 1: Key Generation and Signing

The message authentication is used by any smart meter that wants to send a message to MDMS or any MDMS that wants send a message to smart meter. So, similar to BiBa method [7] signer has t 1-bit random string named SEAL: S_1, S_2… S_t. First signer computes mask for timestamp as $t_{mask1}=H_1(T_1)$, and executes Algorithm 2 to achieve the signature for the message, then execute Algorithm 3 to get a temporary public key for signature verification. Signer forms packet: (m, T, C, sig [Si, Sj, S_k], PK_t [Vi, Vj, V_k]} and sends it to verifier.

Algorithm 2

1. for c←1 to 100 do
2. Compute h=H (m | t_{mask} | C)
3. Pick G_h from Hash Function Family G
4. for i← 0 down to t -1 do
5. for j← 0 down to t -1 do
6. for k← 0 down to t -1 do
7. if $G_h(S_i) = G_h(S_j) = G_h(S_k)$
8. Return sig= {Si, Sj, S_k}

Algorithm 3

1. Take t_{mask} as timestamp mask,
   and K= (ID_{sm}| PW |ID_{MDMS})
2. Compute R=K· t_{mask} × P
3. for i←1 to 3 do (for 3-way collision)
4. $V_i=S_i \times R$ add to PKt set
5. return PK_t

Phase 2: Verification

Verifier receives packet with message m′, signature {Si′, Sj′, S_k′} and temporary public key {Vi, Vj, V_k}. Verifier knows about H1, H, P, and hash family function G. Firstly, he/she should authenticate SEALs, hence calculates these values: $t_{mask1}=H_1(T_1)$, K= (ID_{sm} |PW| ID_{MDMS}), R=K· t_{mask1} × P, then checks $Si' \neq S_j' \neq S_k'$, $V_i=R\times S_i'$, $V_j=R\times S_j'$, $V_k=R\times S_k'$.

Besides forms h=H (m′ | $t_{mask\ 1}$ | C), picks G_h from hash family function G and apply that on all SEALs on signature set. If $S_i \neq S_j \neq S_k$ and $G_h(S_j)$ is equal with $G_h(S_j)$ and $G_h(S_k)$, then verifier can authenticate the message and will make sure the message is not forge or any change. For message authentication we just use one hash function three times and this operation is very light weight and fast.

C. Brief Analysis of the Protocol

Comparing to the BiBa schema, we eliminate the SEALs authentication phase. In the BiBa SEALs should authenticate with a way like a Merkle tree. In the Merkle tree method SEALs are denoting with leaves and signature denotes root of the tree. In this authentication t=1024, hence we need to operate 1023 hash function. To achieve a security of at least O (2^{80}), a hash function must have at least 160 bits. Therefore, as a computational cost we have: 1023 × 160 = 163,680 bits or 20,460 bytes. But in our schema we do not authenticate SEALs in separate way, instead we build a temporary public key, which is built in Algorithm 3. We have one multiplication as a password mask and timestamp mask, and just two scalar multiplications as multiplication of point P in (K · t_{mask}), and S_i in point R. If we use the 3-way collision for compute signature, then we need 1 multiplication and 4 scalar multiplications.

D. Security for signature

In this section, we describe the occupancy problems for bins and balls algorithm and explain the security of our proposed algorithm. Occupancy problems deal with pairings of objects. The basic occupancy problem is about placing t balls into n bins. Let Xi be the random variable which counts the number of balls in the bin i (so Xi is not an indicator). Plainly

$$\sum_{i=1}^{n} X_i = t \qquad (2)$$

Xi has the binomial distribution [6]. To find this out, let X_{ij} be the indicator random variable for ball j going into the bin i, so that $X_i = \sum X_{ij}$ and

$$X_{ij} \begin{cases} 1 \text{ if ball j goes into bin i} \\ 0 \quad \text{otherwise} \end{cases}$$

Then each Xij represents a Bernoulli trial with probability p = 1/n, which is the probability of ball j going into bin i. Since Xi is a sum of Bernoulli trials, it has the binomial distribution.

Specifically, for the probability of a particular bin having exactly k balls, it has a distribution of the form:

$$Pr[X_i = k] = \binom{t}{k} P^k (1-P)^{t-k} = \binom{t}{k} \left(\frac{1}{n}\right)^k \left(1-\frac{1}{n}\right)^{t-k} \qquad (3)$$

But here we need to calculate the probability of a particular bin having at least k balls. If we look at any

subset of balls of size k, then the probability that the subset of balls falls into the bin i is equal to $\left(\frac{1}{n}\right)^k$.

Note that we no longer have the $\left(1-\frac{1}{n}\right)^{t-k}$ factor, because we don't care about where the rest of the balls fall. We then take a union bound of these probabilities over all $\binom{t}{k}$ subsets of size k. The events we are summing over, though, are not disjoint. Therefore, we can only show that the probability of a bin having at least k balls is $\binom{t}{k}(\frac{1}{n})^k$.

As an example, assume n=1000 as bins or the result of hash family function G_h range [1, n-1], t=1024 as balls or SEALs and k=2 as a two-way collision. So the probability of finding at least one two-way collision is equal to:

$Pr[k]=\binom{1024}{2}(\frac{1}{1000})^2=0.523776$ or 52%

So it is 52% chance to find a signature in first examine. Now we assume 10 SEALs have been revealed. So $Pr[forge]=\binom{10}{2}(\frac{1}{1000})^2=0.000045$ or 0%, it means an attacker has 0% chance to forge a signature. In figure 2 study probability of finding signature with different SEALs is shown. In blue line when we have 1000 SEALs and 1000 n, the probability of finding a signature with a two-way collision in first try is 50% in average. In red line when we have 1400 SEALs and 1000 n, the probability of finding a signature with a three-way collision in first try is 45% in average. Therefore we should decrease the number of n and SEALs because with three-way collision we have more security and we can use less n and SEALs. So in green line, we have a three-way collision, 500 n and 700 SEALs with probability 50% in first try in average. As a consequence, for performance analyses, we assume t= 700.

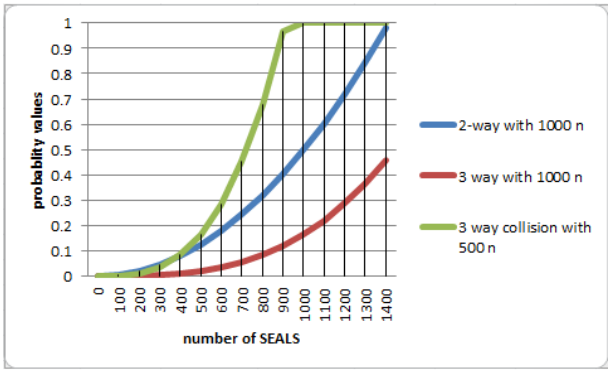

Figure2. Probability of finding signature

IV. ANALYSIS

A. Security Analysis

Since the proposed protocol is based on BiBa and ECC, it utilizes of their security. In this section, we introduce some attacks and analysis our model security against them.

1) Replay attack: since protocol uses timestamp for each packet, a hostile cannot perform a replay attack on messages. Also, each party use of random strings and secure perfect hash function that it protects protocol against replay attack.

2) *Key Privacy & Insider Attack Resilience:* in this approach there is not a pre build private and public key, the signature and the temporary public key is created from a message that is supposed to send in every time and there is not a shared information between MDMS and all smart meters. So an insider attack cannot happen, and other smart meters cannot exploit a smart meter information to arrange a local attack.

3) Off-line Guessing Attack Resilience: let assume an attacker eavesdrops channel and obtains signature, then he/she run a dictionary attack on Q^x and finds this parameter, but based on ECDLP, the attacker is not capable to find PW. Also attacker could not use of Q^x in the next packet because it updates with timestamp mask every time.

4) Denning-Sacco Attack Resilience: our hash functions are resilience against second pre-image attack and also we do not use of session key. So the Denning Sacco attack could not perform on our protocol.

5) MITM Attack: in mutual authentication protocol and in the first step we send the first packet with ID_{sm}, signature and a sequence number. Suppose a MITM capture a packet and change the signature with his/her information, but in destination, MDMS first authenticates the signature and if it failed, it aborts the process and SM should initialize the process again. This operation exactly happens for second packet in SM side.

6) Denial of Service (DOS) attack: If a valid SM gets malicious can arrange a DOS attack against MDMS with an initial mutual authentication request repeatedly. To prevent this attack, MDMS can restrict the number of mutual authentication request for a certain SM in a period of time.

B. *Performance Analysis*

Fast packet delivery: First we present execute time of several cryptographic operations: a scalar multiplication of a point time ($T_{s.mul}$), execute a hash function time (T_{hash}) and multiplication operations (T_{multi}) in a practical environment. The operations were built with a standard cryptography library named MIRACLE and the hardware platform was 32bit operation system and an Intel A80386-16MHz processor with 256-MB memory for a constraint-resource smart meter and INTEL Pentium 4.3.2 GHz with 1G memory for MDMS. For achieving to 1024 bit RSA security, we utilize the ECC group on Koblitz EC. The results are shown in table II for each operation. In table III we analyze operation time for each protocol. E.T 1 refers to

execute time for the mutual authentication protocol. As it can be seen, the operation is very light-weight and fast. E.T 2 refers to execute time for message authentication protocol. Execute time of this protocol is more than mutual authentication. The reason is that we use of ECDLP for increase security of temporary public key. If we just use of message authentication code (MAC) for signature, an adversary could perform a password guessing attack on MAC and forge signatures.

Table II. Cryptographic operation time (in milliseconds)

Device	$T_{s.mul}$	T_{multi}	T_{hash}
SM	15.21	<0.002	<0.005
MDMS	0.49	<0.0002	<0.0001

Table III. Performance results for the proposed protocols

Device	*Mutual Authentication*	*Message authentication*	*E.T 1*	*E.T 2*
SM	$7T_h+tT_h+2T_{s.mul}$	$7T_h+tT_h+8T_{s.mul}+2T_{multi}$	33.955	64.391
MDMS	$7T_h+tT_h+2T_{s.mul}$	$7T_h+tT_h+8T_{s.mul}+2T_{multi}$	1.0507	4.9711

Low Implementation Cost: To show how much our implementation cost is low, we classify and summarize the performance of two protocols in tables IV and V.

Table IV. Implementation cost: mutual authentication protocol

	Hash function	*Random string*	*Concatenation*	*Packet*	*Step*
First protocol	2t+13	t	4	2	3

Table V. Implementation cost: message authentication protocol

Phase	Hash function	Random string	Concatenation	Scalar multiplication	Multiplication
Signing	t+2	t	2	2	2
Verification	4	-	2	2	2

V. CONCLUSION

In this paper, we studied a mutual authentication and a message authentication protocol for smart grid. Since smart grid as modern network for grid power needs more efficient approach for mutual authentication, because most of current solutions do not consider mutual authentication, As a result, a hostile is able to impersonate devices or manipulate data without hacking a SM or a MDMS. So we proposed a lightweight mechanism for mutual authentication between SM and MDMS. Our proposed mechanism inherits advantages of ECDLP, probability theory, basic like birthday problems and hash based protocols.

REFERENCES

[1] Hasen Nicanfar, Victor C. M. Leung, 2013, "Multilayer Consensus ECC-Based Password Authenticated Key-Exchange (MCEPAK) Protocol for Smart Grid System", IEEE Trans. Smart Grid, Vol. 4, 253-264

[2] R. L. Rivest, A. Shamir, L. Adleman, 1978, "A Method for Obtaining Digital Signatures and Public-Key Cryptosystems", Communications of the ACM 21, 120–126.

[3] M. Kgwadi and T. Kunz, 2009, “Securing RDS broadcast messages for smart grid applications,” Dept. Syst. Comput. Eng. Carleton Univ., Ottawa, Canada, Tech. Rep. SCE-09-06.

[4] M. M. Fouda, Z. MD. Fadlullah, N. Kato, R. Lu, and X. Shen, 2011, “A lightweight message authentication scheme for smart grid communications,” *IEEE Trans. Smart Grid*, Vol. 2, 675–685.

[5] Kim. Young-Sam, Heo. Joon, 2012, "Device authentication protocol for smart grid systems using homomorphic hash", IEEE communication and networks, Vol. 14, p 606-613

[6] Alfred J. Menezes, Paul C. Van Oorschot, Scott A. Vanstone, 1997, “Handbook of Applied Cryptography”, CRC Press LLC, USA, p 52, 53

[7] Adrian Perrig, 2001, “the BiBa One-Time Signature and Broadcast Authentication Protocol”, *Eighth ACM Conference on Computer and Communication Security*, pages 28–37

[8] ITU-T Study Group 17, 2007, “Password-Authenticated Key Exchange (PAK) Protocol,” *Telecommunication Standardization Sector of International Telecommunication Union (ITU-T)*, Recommendation X.1035.

[9] Oded Goldreich, Shafi Goldwasser, Silvio Micali, 1986, "How to Construct Random Functions", *Journal of the ACM*, Vol. 33, no. 4, p. 792-807

[10] Neal. Koblitz, 1987, “Elliptic curve cryptosystems”, *Math. Comput, 48* 203–209.

[11] Victor. S. Miller, 1986, “Use of elliptic curves in cryptography”, Advances in Cryptology Crypto '85: Proceedings, Springer Berlin, Heidelberg, p. 417-426