A New Privacy-Preserving Smart Grid System

Jiaping Lin[1, a], Xingwen Zhao[2, b]

[1]School of Telecommunication Engineering, Xidian University, XI'AN, 710071, China

[2]School of Network and Information Security, Xidian University, XI'AN, 710071, China

[a]email: jplinseven@sina.com, [b]email:sevenzhao@hotmail.com

Keywords: Smart Grid; Group Signature; Privacy Protection; K-times Authentication

Abstract. Smart grid can be used to continuously measure, monitor, predict and even control electricity consumption. However, the energy consumption information may reveal the privacy of users. In this paper, we propose a new privacy-preserving smart grid system based on a k-times short dynamic group signature supporting the controllable linkability. Only the group manager can confirm the identity of the signer, so our system can protect the user's privacy. Each user can submit the consumption value up to k times in each time period, e.g. each day. The users can report the consumption value regularly and the service provider will not be subject to heavy computation burden. We show that the proposed scheme is as efficient as other smart grid system, while achieving the additional security feather of times limited authentication.

1. Introduction

Smart grid is modernization of the existing traditional electricity grid. With more and more global energy challenges appearing, North America and Europe firstly proposed smart grid plan.

With the help of real-time users' energy consumption information, the smart grid is able to monitor grid, forecast users' demands, and control the energy generation /consumption. However, reporting detailed consumption information to the supplier will incur serious privacy issues. A long-time statistical analysis on the user's data can sketch out the user's family life clearly [1], [2].

Information such as when the customers get up, when they come back from work or the household appliances they used can be extracted from the consumption profiles.

Personal information is more and more publicly accessible due to modern technologies, accordingly, privacy is increasingly becoming an important security property [3]. While the home lives of residents belong to sensitive information, the disclosure of such personal information may be used by malicious attackers. The invasion of privacy has also become obstacles to the development of smart grid [4].

In order to solve the privacy issues, several privacy preserving smart metering schemes have been proposed.

Metke and Ekl [5] had discussed the key security technologies for a smart grid system, including public key infrastructures and trusted computing.

Rial and Danezis [6] proposed that there was a tamper-resistant device with each smart metering, and it enabled the smart metering to locally compute the billing and didn't need to transmit the fine-grained consumption information to collector.

Costas Efthymiou [7] using the the escrow agreement to design the anonymous smart meter ID. Alfredo Rial [8] put forward a scheme by using the knowledge of commitment and zero Proof idea.

Mikhail Lisovich [9] and Patrick McDaniel [10] used the relevant strategy to protect the privacy of users from the view of the hardware of smart meters.

There was a third party escrow mechanism to authenticate metering reading and it's difficult to associate these readings with a particular smart metering in [11].

Lei Yang [15] and others studied cost-effective smart meter data privacy protection by using batteries. They designed a dynamic programming framework for consumers to jointly protect smart meter data privacy and reduce the cost of the electricity.

Ken Birman [16] uses a combination of decentralization and differential privacy techniques to keep customers' private data hidden from the utility.

© 2015. The authors - Published by Atlantis Press

Liu [17] used ad-hoc ring signature to protect the user privacy, and it could also provide traceability of illegal users who signed twice or more. Liu [17]was not scalable in a huge number of users.

Zargar and Yaghmaee [18] protected the user privacy by making group signature on the message, and the collector could only verify the validity of the received message without knowing who was the signer. The collector couldn't find out the illegal smart metering itself, and it must ask the group manager for help.

Fabio Borges and Florian Volk [19] adapt a homomorphic encryption scheme based on elliptic curve cryptography to efficiently protect the data series of measurements that are collected by smart meters.

Daniel et al. [20] use a bucket principle where each participant has an own bucket that is encrypted individually and therefore only accessible by a trusted server and himself.

Feng Diao et al. [4] propose a privacy preserving smart metering scheme based on the new linkable anonymous credential. However, users in their scheme can submit the data as many times as they want, and the signatures will not be rejected if each signature is attached a different timestamp. Thus, the service provider may be subject to attacks that will cause heavy computation or even denial of service.

Our Contribution. In this paper, we proposed a new privacy-preserving smart grid system. A k-times short dynamic group signature scheme with linkable ability and dynamic enrollment is employed.

In our smart grid system, smart meter can compute the electricity consumption on an interval such as every 15 minutes, or every half an hour, or every hour, with the interval depending on the strategy of the service provider. That is to say, each user can submit the consumption value up to k times in each time period where k is controlled by the service provider.

Group signatures can be anonymously linked, but the corresponding linkage information can only be revealed with a linking key. The linking key is secretly managed by a privileged party called a linker who is delegated the link capability by the opener. Note that the capability of linking signatures is placed below the capability of opening the signer identity of the signatures. We can control the anonymity by adding controllable linkability to the controllable anonymity that can identify a signer from signatures using an opening key.

The scheme supports a dynamic group membership where a user can join or leave a group. Leaving a group is also referred as to be revoked. However, the linking capability can be consistently preserved regardless of changes to the membership status of the signer. In addition, the controllable linkability property does not expose the history of the joining and revocation.

2. Preliminaries

We now review bilinear maps and computational assumptions for our scheme.

A. Bilinear Maps

Let G_1 and G_2 be multiplicative groups of prime order

p and e : $G_1 \times G_2 \to G_T$ be a bilinear map which satisfies the following properties:

(1) Bilinear: $e(g^a, h^b) = e(g,h)^{ab}$ for all $g \in G_1, h \in G_2$ and $a, b \in Z_p$.

(2) Non-degenerate: There exists $h \in G_1$ such that $e(g,h) \neq 1$.

(3) Computable: There exists an efficient algorithm to compute (g', h') for all $g' \in G_1$ and $h' \in G_2$.

The readers can refer to [12], [13] for more about bilinear pairing.

B. The Modified q-Strong Diffie-Hellman ($q - SDH^+$) Problem

The $q - SDH^+$ problem is to compute a pair $((g_1 g_2{}^y w^z)^{\frac{1}{r+x}}, x, y, z)$ for a given $(q+5) - tuple(g_1, g_2, w, g_1{}^r, ..., g_1{}^{r^q}, h_1, h_1{}^r)$ where $r \xleftarrow{R} Z_p{}^*$ and $y, z \in Z_p{}^*$. The readers can refer to [14] for more discussion about $q - SDH^+$ Problem.

C. The Modified ZKPK Protocol for SDH^+

We present an honest-verifier zero-knowledge proof of knowledge (ZKPK) protocol called the modified $ZKPK^{SDH^+}$ which is evolved from $ZKPK^{SDH^+}$ of [14]. This protocol proves possession of an SDH^+tuple and the equality of three discrete logarithms related to this tuple. The concrete protocol is described as follows. Let $e: G_1 \times G_2 \to G_T$ be a bilinear pairing, and $g, u, w, d, g_1, g_2 \xleftarrow{R} G_1, h_1 \xleftarrow{R} G_2, h_\theta = h_1^\theta$ for random $\theta \in Z_p^*$. The public parameters of the modified $ZKPK^{SDH^+}$ are $(e, g, h_1, h_\theta, g_1, g_2, u, w, d)$.

Assume that the prover has an $SDH^+tuple(A, x, y, z)$ where $A^{x+\theta} = g_1 Y^{-1} w^{-z} \in G_1$ for $x, y, z \xleftarrow{R} Z_p^*$. Let $Y = g_2^{\,y}$ and $\hat{Y} = g^y$. The goal of prover P is to show that $A^{x+\theta} = g_1 Y^{-1} w^{-z}$, $\log_g Y = \log_g \hat{Y}$ and $E_i = h_i^{\,y}$ hold. That is to say, P should generate a proof, $PK[(A, x, y, z): A^{x+\theta} = g_1 g_2^{\,-y} w^{-z} \wedge y = \log_g \hat{Y} \wedge (E_i = h_i^{\,y})]$

Prover P performs the following protocol with verifier V.

•**(Prover)** P picks $\alpha \xleftarrow{R} Z_p^*$ and computes $E_i = h_i^{\,y}, D_1 = u^\alpha, D_2 = Aw^\alpha, D_3 = g^y d^\alpha \in G_1$ and $r = x\alpha - z \bmod p$. It also picks $r_\alpha, r_x, r_y \xleftarrow{R} Z_p^*$ and compute $E_i^{'} = h_i^{r_y}$, $R_1 = u^{r_\alpha}$, $R_2 = e(D_2, h_1)^{r_x} e(w, h_\theta)^{-r_\alpha} e(w, h_1)^{-r_y} e(g_2, h_1)^{r_y}$, $R_3 = g^{r_y} d^{r_\alpha}$. $(D_1, D_2, D_3, E_i, R_1, R_2, R_3, E_i^{'})$ are sent to V.

•**(Verifier)** V picks $c \xleftarrow{R} Z_p^*$ and sends it as a random challenge to P.

•**(Prover)** P computes $s_\alpha = r_\alpha + c\alpha, s_x = r_x + cx, s_r = r_r + cr$, and $s_y = r_y + cy$ under module p, and then sends $(s_\alpha, s_x, s_r, s_y)$ to V.

•**(Verifier)** V checks if the followings hold $R_1 \overset{?}{=} u^{s_\alpha} D_1^{-c}$, $R_2 \overset{?}{=} e(D_2, h_1)^{s_x} e(w, h_\theta)^{-s_\alpha} e(w, h_1)^{-s_y} e(g_2, h_1)^{s_y} \cdot (e(D_2, h_\theta) / e(g_1, h_1))^c$ $and R_3 \overset{?}{=} g^{s_y} d^{s_\alpha} D_3^{-c}$. If all equality tests hold then output 1 (it means the signature is valid), and otherwise, 0 (invalid).

In the above protocol, the prover and the verifier execute a proof of knowledge of (α, r, x, y, z) which satisfy the following equations:

(1) $u^\alpha = D_1$,

(2) $e(D_2, h_1)^x e(w, h_\theta)^{-\alpha} e(w, h_1)^{-r} e(g_2, h_1)^y = e(g_1, h_1) e(D_2, h_\theta)^{-1}$

(3) $g^y d^\alpha = D_3$

(4) $h_i^y = E_i$.

3. System Model

Electricity data statistics and charging agreement mainly includes four entities: service providers, trusted third party (TTP), smart electric meter, users (include their PCs and mobile devices).

a. **Service providers** made electricity tariff and collected the electricity data from each smart meters, and charge to users.

b. The **trusted third party** is an institution which the user and the service providers trust at the same time. It is responsible for managing the users. It authenticates user's smart meters and generates the group signature. When there are problems, it is responsible for solving the dispute justly.

c. After **smart meters** got the TTP certification and had the group signature, smart meters sent the electricity data to service providers and users with an interval which is controlled by service provider.

d. A **user** can have one or more smart meter and pay the corresponding fees to the service provider. The relationship between the entities is shown in Fig.1.

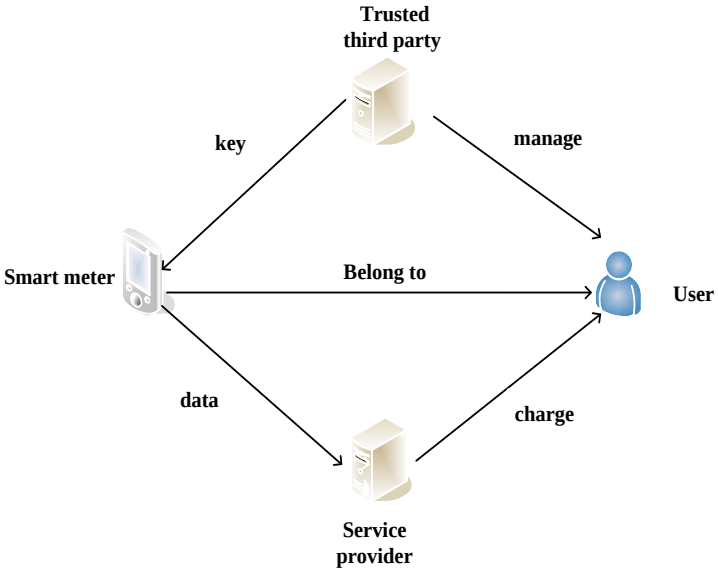

Fig.1. Relationship between the entities

We make an overall introduction for the protocol.

1. Order and establish the service.

A user signs a contract with service providers at first, and in the process the users and service providers use their public key to ensure the safety of the communication. When users join a group controlled by the trusted third party, the trusted third party specifies a corresponding group by their geographic position. The user's smart meters generates group private key used to generate group signature at this stage. Then TTP notifies the service provider that the user (with the group public key) is in the group.

2.Collect electricity data

Smart meter computes the electricity consumption every 15 minutes, or every half an hour, or every hour. The interval depends on the requirement of the service provider. So each user can submit the consumption value up to k times in each time period where k is controlled by the service provider. When receiving the consumption value, each user uses the group private key to make the group signature for it. Users make a signature on message and each user is allowed to sign up to k times in every period. Then smart meter sent the electricity data and the group name to service provider. The service provider stored the data in the local database.

3.Users pay the electricity bills

The users connect with the service provider from user interface and pay the bills. The service provider needs to announce a price list. Users can download the price list. Each user calculates the sum of the electricity consumption, and makes a signature by using his group private key. Finally users send the signature to service provider. Service provider opens the signature to verify, if the total electric charge of all users in a group is different from the charge that all users pay, the service provider will let the trusted third party to check what happened.

4.Solve the dispute

When the service provider finds that the total electric charge of all users in a group is different from the charge that all users pay, the trusted third party will find out the leakage of electricity users. The service provider sends all the group signature of the problem group and the signature of users paying fees to the trusted third party. The trusted third party will open all signatures to see which user cheated.

4. Our Scheme

In this section we present a scheme using traceable commitment and privacy-preserving short group signatures. Basically our construction relies on the modified $ZKPK^{SDH^+}$. A signer will obtain a valid $SDH^+ tuple$, $(x, y, z, A = (g_1 g_2{}^{-y} w^{-z})^{1/\theta+x})$ as a signing key at the joining process. To

generate a signature, the signer proves possession of a valid $SDH^{+}tuple$ and also the equality of three exponents, i.e., $\log_{g_2} g_2^{\,y}, \log_g g^y and \log_{h_i} h_i^y$. In other words, a signature includes an $SDH^{+}tuple$ and a linking value y associated with the $SDH^{+}tuple$. The signatures will not reveal any identity of a user, and they will not be linked unless with an opening key or a linking key.

Our scheme consists of a setup algorithm, a group membership issuing algorithm which involves two sub-algorithms User Join and Issue to generate a user signature key, and signing, verifying, opening, linking, and revocation algorithms.

A. Set up

For given a security parameter, proceed as follows. TTP generates a tuple of algebraic groups of prime order p, i.e., (G_1, G_2, G_T) and a bilinear map $e: G_1 \times G_2 \to G_T$. Pick random $h_1 \xleftarrow{R} G_2 \setminus \{1_{G_2}\}$ and $g, g_1, g_2, u \xleftarrow{R} G_1 \setminus \{1_{G_1}\}$ and $\eta, \xi, \theta \xleftarrow{R} Z_p^{\,*}$. TTP also computes $w = u^{\eta}, d = u^{\xi}, L = h_1^{\xi}, h_\theta = h_1^{\theta}$. Then it selects two cryptographic hash functions $\mathrm{H}_1 : \{0,1\}^* \to Z_p^{\,*}, \mathrm{H}_2 : \{0,1\}^* \to G_T$.

An initial group public key is $gpk_0 = (e, g, h_1, h_\theta, H_1, H_2, g_1, g_2, u, w, d)$ and the master issuing key, $mik = \theta$, the master opening key, $mok = (\eta, \zeta)$, and the master linking key, $mlk = L$. In the group public key, g_1, g_2, u, w, d will be updated per revocation.

B. User join

User join, which is run by a joining user, and Issue, which is run by the Issuer, interactively perform the following protocol to generate a user signature key $usk[i] = (x_i, y_i, z_i, A_i = (g_1 g_2^{-y_i} w^{-z_i})^{\frac{1}{\theta + x_i}})$.

Step1: User picks a secret key $sk_{ID} = z \xleftarrow{R} Z_p^{\,*}$ and computes its corresponding public key $Z = upk_{ID} = w^z$. User also picks $\tilde{r} \xleftarrow{R} Z_p^{\,*}$ and computes $\tilde{W} = w^{\tilde{r}}$. Then he computes $\tilde{c} = \mathrm{H}_1(M_{JR}, w, Z, \tilde{W})$ and $\tilde{s} = \tilde{r} + \tilde{c} z (\mathrm{mod}\, p)$ where M_{JR} is a pre-defined message that is used to request a joining process. Let $\tilde{Y}_{ID} = (Z, \tilde{s}, \tilde{c})$. User sends $(JOIN - REQUEST, ID, \tilde{Y}_{ID})$ to issuer. Assume that $\mathrm{Z} = \mathrm{upk}_{\mathrm{ID}}$ is cryptographically bounded with the identity ID through an additional technique such as a certificate.

Step2: Upon the receipt of the join-request message $(JOIN - REQUEST, ID, \tilde{Y}_{ID} = (Z, \tilde{s}, \tilde{c}))$, issuer computes $\tilde{c}' = \mathrm{H}_1(M_{JR}, w, Z, w^{\tilde{s}} Z^{-\tilde{c}})$ and checks if $\tilde{c} = \tilde{c}'$. If the equality holds, issuer checks whether the ID is registered, that is, there exists j with $REG[j] = (ID, ...)$ on the registration list $REG = (REG[1], ..., REG[n-1])$. If no such j exists then issuer picks $x, y \xleftarrow{R} Z_P^{\,*}$ and computes $A = (g_1 g_2^{-y} w^{-z})^{\frac{1}{\theta + x}}$. Otherwise, issuer finds y from $REG[j] = (ID, g^y, A, x, y, upk[j])$, and picks $x \xleftarrow{R} Z_p *$ to compute $A = (g_1 g_2^{-y} Z^{-1})^{\frac{1}{\theta + x}}$. Issuer clear $REG[j]$ and adds $REG[n] = (ID, g^y, A, x, y, upk[n] = upk_{ID,} Y_n = g_2^{\,y}, Xn = h_1^{\,x})$ to REG. Finally, issuer sends (n, A, x, y) to user.

Step3: After receiving the message (n, A, x, y) for some non-negative integer n, if the equality $e(A, h_\theta h_1^{\,x}) = e(g_1 g_2^{-y} w^{-z}, h_1)$ holds the user keeps $usk[i] = (\lambda = 0, x, y, z, A = (g_1 g_2^{-y} Z^{-1})^{\frac{1}{\theta + x}})$ secret as its private signing key.

C. Sign

Given a user private signing key $usk[i] = (\lambda, x, y, z, A)$ and a message M, this algorithm proceeds as follows:

Step1: Calls Update-usk of the revocation algorithm (shown below) with $usk[i]$ and obtain $gpk_{\lambda} = (e, g, h_1, h_\theta, H_1, H_2, g_1, g_2, u, w, d)$ and its corresponding user signature key $usk[i] = (\lambda, x, y, z, A)$.

Step2: Picks $\alpha \xleftarrow{R} Z_p^*$ and computes $\gamma = x\alpha - z(\text{mod } p)$, $D_1 = \hat{u}^{\alpha}$, $D_2 = \hat{A}\hat{w}^{\alpha}$, $D_3 = g^{y}\hat{d}^{\alpha}$ and $E_i^{'} = v_i^{r_y}$.

If the service provider allows each user to submit up to k times on time period T such as "2015-07-22", the user selects an unused $j \in [1, k]$ and calculates $v_i = H_2(T, i)$. Let $E_i = v_i^{y}$.

Step3: Picks $r_\alpha, r_x, r_r, r_y \xleftarrow{R} Z_p^*$ and computes $R_1 = \hat{u}^{\gamma_\alpha}$, $R_2 = e(D_2, h_1)^{r_x} e(\hat{w}, h_\theta)^{-\gamma_\alpha} e(\hat{w}, h_1)^{-\gamma_\gamma} e(\hat{g}_2, h_1)^{r_y}$ and $R_3 = g^{r_y}\hat{d}^{\gamma_\alpha}$.

Step4: Computes $c = H_1(M, D_1, D_2, D_3, E_{i,} R_1, R_2, R_3, E_i^{'})$, and $s_\alpha = r_\alpha + ca(\text{mod } p)$, $s_x = r_x + cx(\text{mod } p)$, $s_r = r_r + cr(\text{mod } p)$ and $s_y = r_y + cy(\text{mod } p)$.

Step5: Finally, outputs a signature $\sigma = (\hat{\lambda}, D_1, D_2, D_3, E_i, c, s_\alpha, s_x, s_r, s_y)$,which is a zero-knowledge proof for $PK[(A, x, y, z) : A^{x+\theta} = g_1 g_2^{-y} w^{-z} \wedge (y = \log_g \hat{Y}) \wedge (E_i = h_i^{y})]$.

D. Verify

For the given signature, this algorithm proceeds as follows:

Step1: Calls Update-gpk of the revocation algorithm with gpk_0, $(\lambda^{'}, RL)$ and obtain $gpk_{\lambda^{'}} = (e, g, h_1, h_\theta, H_1, H_2, \hat{g}_1, \hat{g}_2, \hat{u}, \hat{w}, \hat{d})$

Step2: Computes $R_1 = \hat{u}^{s_\alpha} D_1^{-c}$,
$R_2 = e(D_2, h_1)^{s_x} e(\hat{w}, h_\theta)^{-s_\alpha} e(\hat{w}, h_1)^{-s_\gamma} e(\hat{g}_2, h_1)^{s_y} \cdot (e(D_2, h_\theta) / e(\hat{g}_1, h_1))^{c}$,
$R_3 = g^{s_y}\hat{d}^{s_\alpha} D_3^{-c}$, $E_i^{'} = h_i^{s_y} E_i^{-c}$.

Step3: If $c = H_1(M, D_1, D_2, D_3, E_{i,} R_1, R_2, R_3, E_i^{'})$, then outputs 1(valid), and otherwise, 0 (invalid).

E. Open

Given $mok = (\eta, \xi)$, REG and valid (σ, M), the algorithm proceeds as follows:

Step1: Calls Verify to check if the given signature σ is valid.

Step2: If the signature is not valid it outputs $\perp$. Otherwise, it proceeds as follows. It recovers $g^{y} = D_3 \cdot D_1^{-\xi}$ (and alternatively $A = D_2 \cdot D_1^{-\eta}$). Using a binary search on the registration list REG, it finds i such that $g^{y} = g^{y_i}$. If i does not exist then return $(i = 0, *)$. Otherwise, find the corresponding $upk[i] = Z_i = w^{z_i}$ and $PI_i = (Y_i = g_2^{y_i}, X_i = h_1^{x_i})$. Next, it picks $r \xleftarrow{R} Z_p^*$ and compute $Kopen = D_1^{\eta}, V_1 = u^{r}, V_2 = D_1^{r}, c_{open} = \mathrm{H}_1(\sigma, g, K_{open}, V_1, V_2)$ and $s_{open} = r + c_{open}\eta(\text{mod } p)$. It outputs i and a proof $\tau = ((K_{open}, c_{open}, s_{open}), PI_i)$ where $PI_i = (Y_i, X_i)$.

Assume that PI_i is cryptographically bounded with a registered identity through an additional technique such as a certificate.

F. Judge

For valid (σ, M), $upk[i] = Z_i$, a proof $\tau = ((K_{open}, c_{open}, s_{open}), PI_i = (Y_i, X_i)$, the algorithm proceeds as follows.

Step1: If $i \neq 0$ and $\sigma = (\lambda^{'}, D_1, D_2, D_3, c, s_\alpha, s_x, s_r, s_y)$, it calls Update-gpk of the revocation algorithm with $(gpk_0, \lambda^{'}, RL)$ and obtains $gpk_{\lambda^{'}} = (e, g, h_1, h_\theta, H_1, H_2, \hat{g}_1, \hat{g}_2, \hat{u}, \hat{w}, \hat{d})$.

Step2: Let $h_1 = h_1^{v} \in G_2$ where $v = \log_{g_1} \hat{g}_1$ and $g_1 \in gpk_0$. If $c_{open} = H_1(\sigma, g, K_{open}, u^{s_{open}} w^{-c_{open}}, D_1^{s_{open}} K_{open}^{-c_{open}})$ and $e(D_2 K_{open}^{-1}, X_i h_\theta) = e(g_1 Y_i^{-1} Z_i^{-1}, \hat{h}_1)$ then outputs 1 (valid), and otherwise outputs 0 (invalid).

G. Link

For the master linking key $mlk = L$ and two given pairs of signatures and messages, (σ', M') and (σ'', M''), the algorithm proceeds as follows: Let $\sigma' = (\lambda', D_1', D_2', D_3', c', s_\alpha', s_x', s_r', s_y')$ and $\sigma'' = (\lambda'', D_1'', D_2'', D_3'', c'', s_\alpha'', s_x'', s_r'', s_y'')$.

Step1: Calls Verify to check if the given signatures are valid.

Step2: If any of them is not valid then it outputs $\perp$. Otherwise, it computes $B_1 = e(D_3', h_1)e(D_1', L)^{-1}$ and $B_2 = e(D_3'', h_1)e(D_1'', L)^{-1}$. If $B_1 = B_2$ then it outputs 1(i.e., linked) and otherwise, 0 (i.e., unlinked).Alternatively, $e(D_3'/D_3'', h_1) \stackrel{?}{=} e(D_1'/D_1'', L)$ can be used for the above test.

H. Revocation

The revocation algorithm consists of two subalgorithms, Update-gpk which updates a group public key, and Update-usk which updates a user signature key. Let the initial group public key be $gpk_0 = (T, g_1, g_2, u, w, d)$ where $Q = (e, g, h_1, h_\theta, H)$. Define $v_k = (\theta + x_{j_1})(\theta + x_{j_2}) \cdots (\theta + x_{j_k}) \bmod p$ where k is a positive integer. Let the revocation list denote $RL = \{(S_{1,k}, S_{2,k}, S_{3,k}, S_{4,k}, S_{5,k}, x_{j_k}) \mid k = 1,..., \hat{\lambda}\}$, $S_{1,k} = g_1^{1/v_k}$, $S_{2,k} = g_2^{1/v_k}$, $S_{3,k} = u^{1/v_k}$, $S_{4,k} = w^{1/v_k}, S_{5,k} = d^{1/v_k}$ and x_{j_k} corresponds to the j_k^{th} registered but revoked user.

• Update-usk. For given $usk[i] = (\lambda, x, y, z, A)$, proceed as follows: Assume that $x \neq x_{j_k}$ for any $k = 1,..., \hat{\lambda}$.

- If $\hat{\lambda} = \lambda$ then return (λ, x, y, z, A).

- Else if $\hat{\lambda} = \lambda$, update the key as follows. Compute $gpk_{\hat{\lambda}}$ by calling Update-gpk with $(gpk_0, -1, RL)$.

- Let $\pi = \hat{\lambda} - \lambda$ and $t_k = j_{\lambda+k}$ for k = 1, . . . ,

- Compute $\hat{A} = A^{\frac{(-1)^\pi}{\delta_\pi}} \Pi^\pi{}_{i=1}[S_{1,i} S_{2,i}{}^{-y} S_{4,i}{}^{-z}]^{(-1)^{\pi-i}\delta_{i-1}/\delta_\pi}$, where is $\delta_0 = 1$ and is $\delta_i = \Pi^i{}_{k=1}(x - x_{t_k}) \bmod p$. Return $(\hat{\lambda}, x, y, z, \hat{A})$.

• Update-gpk. Given a tuple (gpk_0, ρ, RL), the algorithm proceeds as follows.

- If $\hat{\lambda} \leq \rho$ then abort.

- Else if $\rho = -1$ then update gpk_0 to the latest revoked key as follows. Set $\hat{g}_1 \leftarrow S_{1,j_{\hat{\lambda}}}, \hat{g}_2 \leftarrow S_{2,j_{\hat{\lambda}}}, \hat{u} \leftarrow S_{3,j_{\hat{\lambda}}}, \hat{w} \leftarrow S_{4,j_{\hat{\lambda}}} and \hat{d} \leftarrow S_{5,j_{\hat{\lambda}}}$ and output $gpk_{\hat{\lambda}} = (Q, \hat{g}_1, \hat{g}_2, \hat{u}, \hat{w}, \hat{d})$ where is $T = (e, g, h_1, h_\theta, H)$.

- If $0 < \rho < \hat{\lambda}$ then update gpk_0 up to the ρ^{th} revoked key as follow. Set $\hat{g}_1 \leftarrow S_{1,\rho}, \hat{g}_2 \leftarrow S_{2,\rho}, \hat{u} \leftarrow S_{3,\rho}, \hat{w} \leftarrow S_{4,\rho} and \hat{d} \leftarrow S_{5,\rho}$. Output the updated $gpk_\rho = (Q, \hat{g}_1, \hat{g}_2, \hat{u}, \hat{w}, \hat{d})$.

5. Security Analysis of Our Scheme

Unforgeability. Only group members can make group signature on the message on behalf of the group.

Sketch Proof: A message which the user sent to the service provider consists of one commitment $E_i = h_i^{\,y}$, a group signature from [3] and a standard non-interactive zero-Knowledge proof $\pi_i = PK[(A, x, y, z) : A^{x+\theta} = g_1 g_2{}^{-y} w^{-z} \wedge (y = \log_g \hat{Y})] \wedge (E_i = h_i{}^Y)$.

If anyone can forge a valid message, he/she provides a valid forgery of the group signature for [3]. Since the group signature scheme [3] is proved to be unforgeable, so our scheme is unforgeable.

Anonymity. For a valid group signature, no one except for the group manager can confirm the identity of the signer.

Sketch Proof: Our scheme relies on group signature [3] to be anonymous. If anyone can confirm the identity of the user from the signature, our scheme will not have anonymity, and the signature [3] must not have anonymity. Since the group signature scheme [3] is proved to have anonymity, so our scheme has anonymity.

Traceability. When there are problems and disputes, the group manager can open the group signature to identify the signer.

Sketch Proof: The proposed group signature is traceable due to the proof that the group signature scheme is traceable [3].

Correctness. Signatures generated by an honest user should be verified correctly. With inputs of a message and signature, Open should correctly identify the signer. Link should link the signatures from a signer.

Sketch Proof: If user U_i and group manager are honest, U_i will carefully select a secret key y_i and group manager will make sure that is different from others' public keys. So the secret key y_i is also different from the others. For each unused $k_i \in [1,k]$ during period *T*, there will not exist a value the same as $E_i = h_i^{y_i} = (H_1(T,k_i))^{y_i}$ for index k_i during period *T*, because y_i is different from others' secret keys. Then the verifier will not reject such an execution.

Times Limited Authentication. No one can authenticate more than announced number to the honest verifier.

Sketch Proof: For each user on period *T*, only k bases can be used for generating traceable tags during the execution, namely $v_1 = H_1(T,1),...,v_k = H_1(T,k)$. If the user uses additional base, it can be easily detected by the verifier, because the user has to tell the verifier which base he is using in the execution.

Using these *k* bases, each user can perform only *k* times successful executions, and *w* users can perform *kw* times. After *kw* times normal executions, if they collude together and want to sign the messages for one more time, they need a new secret key to create a different tracing tag other than the *kw* used tags. And they also need to prove knowing a message-signature pair for the secret key, which is not obtained from normal interaction with the group manager. It cannot be fulfilled due to the unforgeability of our scheme

6. Performance

We analyze the performance of our scheme in terms of dynamic user, traceability, times limited authentication. This analysis includes the comparison between five privacy-protecting smart grid system and our proposed system.

TABLE I DIFFERENT APPROACHES COMPARISON

	[17]	[18]	[19]	[20]	[4]	Ours
Dynamic User	Yes	Yes	No	Yes	Yes	Yes
Traceability	Yes	Yes	Yes	No	Yes	Yes
Times limited authentication	No	No	No	No	No	Yes

A. Dynamic User

The system can add new member or revoke some existing user dynamically. This property is needed in real life smart grid where houses are dynamic with the development of local area.

B. Traceability

The service provider sends all the group signature of the suspected users to the trusted third party, so it can open all signatures to find out which user cheated. This property is needed in real life smart grid where some users may be dishonest.

C. Times limited authentication

In our smart grid system each user can submit the consumption value k times in each time period, therefore reporting regularly is convenient to calculate the consumption value. And no one can authenticate more than announced number to the honest verifier. This property is desirable that it can avoid unnecessary heavy computation for the service provider.

TABLE II SIGNATURE LENTH COMPARISON

	[18]	[4]	Ours
Signature Length	$3G_1 + G_T + 5Z_p$	$3N+160 + 5Z_p$	$3G_1 + G_T + 5Z_p + 2l$

G_1 : element in group G_1 ; G_T : element in group G_T ; Z_p : element in group Z_p ; N : the key length in RSA; l :a very small integer.

In table II we compare the signature length of [18], [4] with that of our system, and we don't consider the length of message.

We notice that the signature length in our scheme is almost the same as the scheme in [18] if the two small integers are neglected, and less than that in [4]. Moreover, our scheme can limit the times of data submission so that we can avoid heavy computation burdening in service provider, while each user in the scheme of [18]and [4] can submit data to the service provider as many times as they want if different time-stamps are attached.

7. Conclusion

We describe a new privacy-preserving smart grid system based on a k-times short dynamic group signature supporting the controllable linkability. We also formally proved that the constructed scheme achieves unforgeability, anonymity, traceability, correctness and times limited authentication. Comparing with other smart grid system shows that our system is very versatile and useful in many privacy-enhancing applications with limited resources.

Acknowledgements

This work is supported by the National Natural Science Foundation of China (No.61272457), the National 111 Program (No.B08038) and the Research Fund for the Doctoral Program of Higher Education of China (No.20130203120003). The authors gratefully acknowledge the anonymous reviewers for their valuable comments.

References

[1]G. W. Hart, "Nonintrusive appliance load monitoring," Proc. IEEE,vol. 80, no. 12, pp. 1870–1891, Dec. 1992.

[2]C. Laughman et al., "Power signature analysis, Power Energy," Mag.,vol. 1, no. 2, pp. 56-63, 2003.

[3]Jung Yeon Hwang, Liqun Chen, Hyun Sook Cho, and DaeHun Nyang, Short Dynamic Group Signature Scheme Supporting Controllable Linkability.VOL.10,NO.6,June 2015.

[4]Feng Diao, Fangguo Zhang, Xiangguo Cheng:A Privacy-Preserving Smart Metering Scheme Using Linkable Anonymous Credential. IEEE Trans. Smart Grid 6(1): 461-467 2015.

[5]A. R. Metke and R. L. Ekl, “Security technology for smart grid networks,” IEEE Trans. Smart Grid, vol. 1, no. 1, pp. 99-107, Jun. 2010.

[6]A. Rial and G. Danezis,“Privacy-preserving smart metering, in Proc.10th Annu. ACM Workshop Privacy Electron. Soc., Chicago, IL, USA, pp. 49-60, 2011.

[7]The Smart Grid Interoperability Panel. Smart Grid Cyber Security Strategy and Requirements.Technical Report 7628, National Institute of Standards and Technology.vol2, September 2009.

[8]A.Rial, K.U.Leuven & IBBT, Leuven, Belgium G.: Privacy-preserving smart metering.Technical Report. MSRTR- 2010-150, Microsoft Research , pp.49-60,November 2010.

[9]M. Lisovich and S. Wicker. Privacy concerns in upcoming residential and commercial demand-response systems. In 2008 Clemson University Power Systems Conference. Clemson University, pp. 553-568, March 2008.

[10]P. McDaniel and S. McLaughlin. Security and privacy challenges in the smart grid. IEEE Security and Privacy, pp. 75-77, 2009.

[11]J. C. L. Cheung et al., “Credential-based privacy-preserving power request scheme for smart grid network,” in IEEE GLOBECOM 2011, Houston, TX, USA, pp. 1-5, 2011.

[12] I. F. Blake, G. Seroussi, and N. P. Smart, Advances in Elliptic Curve Cryptography, vol. 317. Cambridge, U.K.: Cambridge Univ.Press, ch. 9, pp. 183-213,2005.

[13]K. G. Paterson, “Cryptography from pairings,” in Advances in Elliptic Curve Cryptography, vol. 317. Cambridge, U.K.: Cambridge Univ.Press, ch. 10, pp. 215-251, 2005.

[14]J. Y. Hwang, S. Lee, B.-H. Chung, H. S. Cho, and D. Nyang, “Group signatures with controllable linkability for dynamic membership,” Inf. Sci., vol. 222, pp. 761-778, Feb. 2013.

[15]Lei Yang, Xu Chen, Junshan Zhang, H. Vincent Poor: Cost-Effective and Privacy-Preserving Energy Management for Smart Meters. IEEE Trans. Smart Grid 6(1): 486-495 2015.

[16]Ken Birman, Márk Jelasity, Robert Kleinberg, Edward Tremel: Building a Secure and Privacy-Preserving Smart Grid.Operating Systems Review 49(1): 131-136 2015.

[17]W. Liu, “Smart meter data transmission security mechanism”,M.S. thesis, School Sci. Technol., Sun Yat-sen Univ., Guangzhou, China, 2013.

[18]S. H. M. Zargar and M. H. Yaghmaee, “Privacy preserving via group signature in smart grid,” in Proc. 1st Congr. Elect. Ind.Autom, 2013 .

[19]Fábio Borges, Florian Volk, Max Mühlhäuser:Efficient, verifiable, secure, and privacy-friendly computations for the smart grid. ISGT 2015: 1-5.

[20]Daniel Brettschneider, Alfred Scheerhorn, Daniel Hölker, Peter Roer, Ralf Tönjes: Privacy-friendly distributed algorithm for energy management in smart grids.NetSys pp. 1-8, 2015.