Final Report of

Two-Pass Privacy Preserving Authenticated Key Agreement Scheme for Smart Grid

Implementation

Submitted by-

Group - 02

Rakin Mohammad Sifullah - 18101003
Sourov Halder - 18101004
Nusrat Afrin Simran - 18101030

Section : A-1

Submitted to -

Dr. A S M Touhidul Hasan
Assistant Professor, Dept. of CSE
University of Asia Pacific (UAP)

What is Smart Grid?

A smart grid is an electricity network enabling a two-way flow of electricity and data with digital communications technology enabling to detect, react and pro-act to changes in usage and multiple issues. Smart grids have self-healing capabilities and enable electricity customers to become active participants.

What is a Password-Authenticated Key Agreement?

In cryptography, a password-authenticated key agreement method is an interactive method for two or more parties to establish cryptographic keys based on one or more party's knowledge of a password.

Objective :

The design of an authenticated key agreement protocol for getting the keen meter correspondences in savvy grid networks has pulled in developing consideration as of late. Acknowledging shared validation and key understanding without dynamic inclusion of a believed outsider is normal in this field. Accordingly, to accomplish this point, a few validated key arrangement plans for smart grid have been introduced; while as examined in the accompanying segment of this work, the vast majority of these plans stay defenseless against different assaults. Thusly, to plan a genuinely secure verified key arrangement convention for shrewd matrix and make a few commitments to this field, this work presents our new validated key agreement plan for smart grid utilizing Elliptic Curve Qu-Vanstone (ECQV) certain declarations as the building block. Our plan is without pairing and simply utilizes two passes to acknowledge common confirmation and key arrangement just as solid certification protection, making it with high productivity in both calculation and correspondence overheads. Additionally, its security is officially demonstrated under the generally acknowledged Canetti and Krawczyk (CK) security model, and is additionally affirmed by a heuristic security examination.

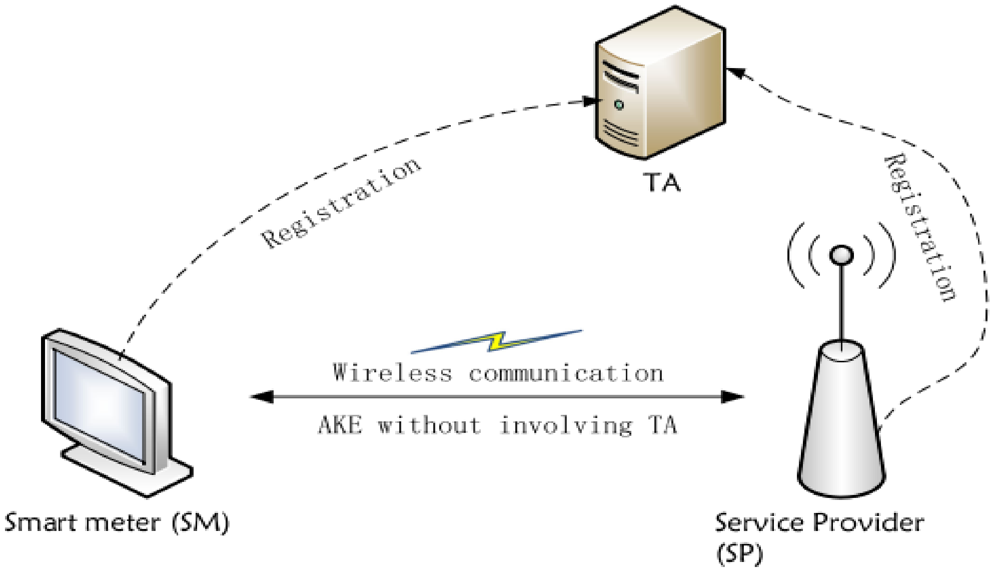


Fig - 01 : Smart metering communication network

By and large, in a smart metering network, a smart meter (SM) and the service provider (SP) should initially enlist to a Trusted Authority (TA), and afterward smart meter can straightforwardly speak with service provider through remote channel without including trusted authority, sending/accepting utilization information or orders, as demonstrated in Fig - 01.

Contributions :

1. We plan another authenticated key agreement protocol for smart grid climate utilizing the Elliptic Curve QuVanstone (ECQV) verifiable endorsement as the structure block. Not at all like some current plans that need to play out the generally tedious pairing activity, our new plan is pairing-free.

2. Our new plan acknowledges mutual authentication with session key agreement just as solid certification security inside two correspondence streams, supporting amazing forward secrecy and opposing different known assaults, though some current plans experience the suffering from certain assaults and need at any rate three correspondence streams to accomplish mutual authentication and session key agreement.

3. The security of our proposed new verified key agreement plot for brilliant

framework climate is officially demonstrated under the CK security model, and furthermore affirmed by heuristic examination. Furthermore, we contrast our new plan and some other as of late introduced confirmed key agreement plans for smart grid in the parts of calculation and correspondence costs, and from which it very well may be seen that our new plan has a significant benefit in proficiency.

Working Steps :

1. **Setup Phase** :

   In this phase, the TA first chooses the secure elliptic curve domain parameters.

2. **Registration Phase** :

   The registration phases for both SMi and SPj are the same with the ECQV certificate scheme.

3. **Authenticated Key Agreement Phase** :

   This section gives a detailed description of the mutual authentication and session key agreement processes of our proposed new scheme.

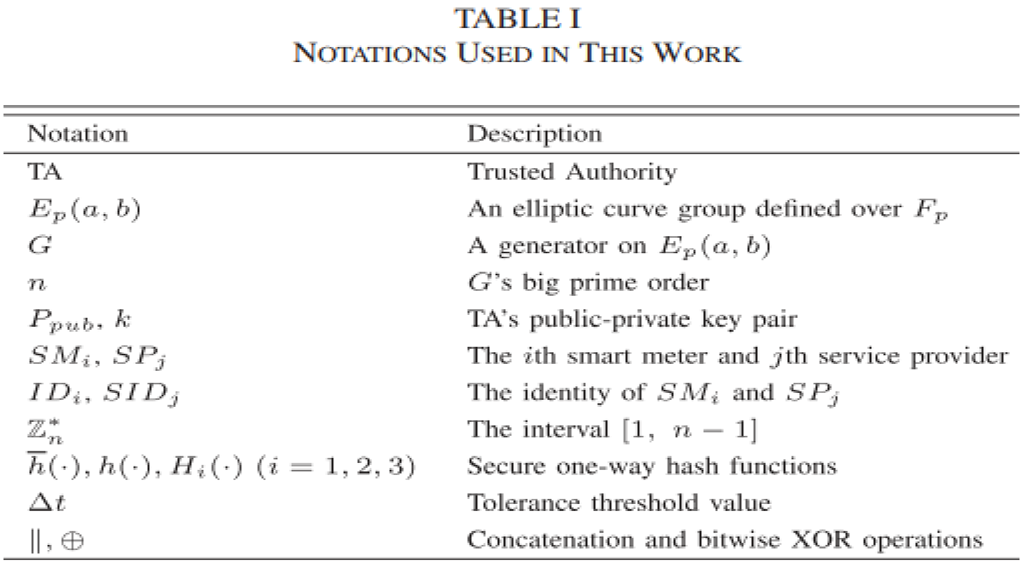
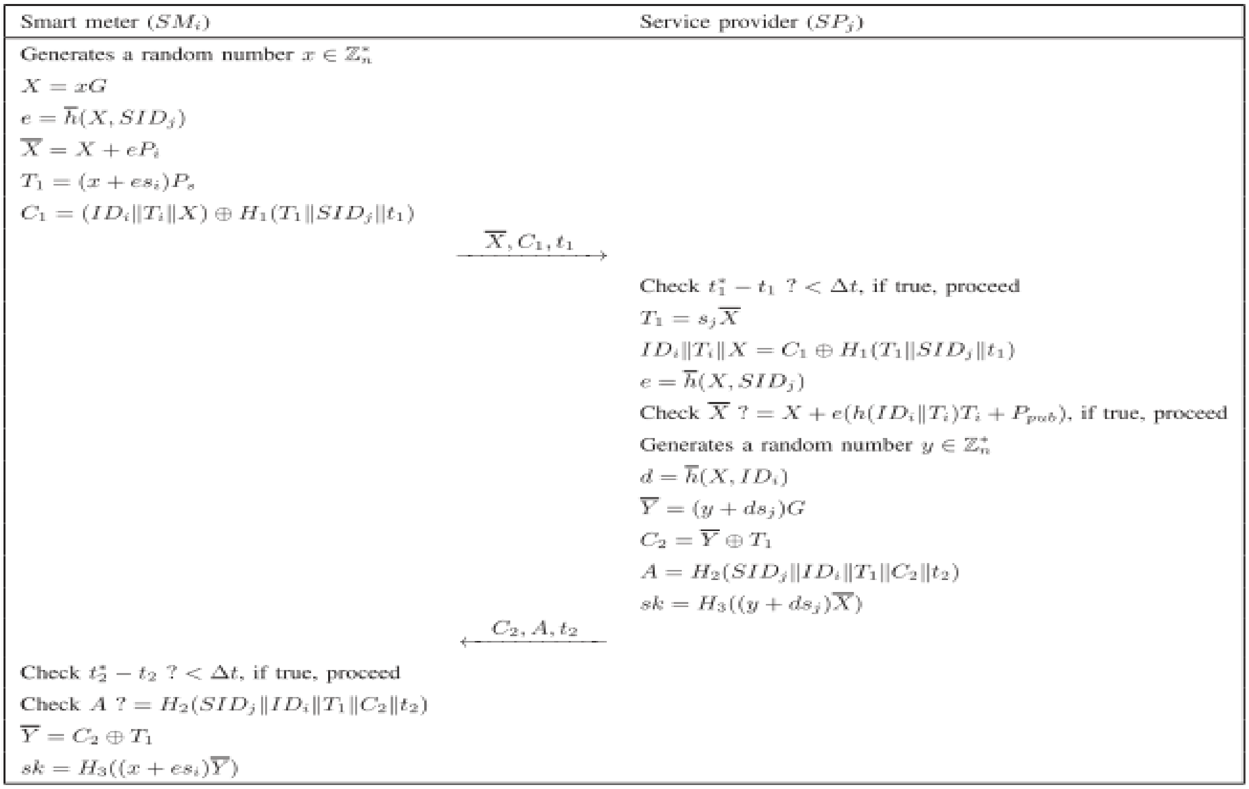


TABLE I
NOTATIONS USED IN THIS WORK

Notation	Description
TA	Trusted Authority
$E_p(a,b)$	An elliptic curve group defined over F_p
G	A generator on $E_p(a,b)$
n	G's big prime order
P_{pub}, k	TA's public-private key pair
SM_i, SP_j	The ith smart meter and jth service provider
ID_i, SID_j	The identity of SM_i and SP_j
$\mathbb{Z}_n^*$	The interval $[1,\ n-1]$
$\overline{h}(\cdot), h(\cdot), H_i(\cdot)\ (i = 1,2,3)$	Secure one-way hash functions
Δt	Tolerance threshold value
$\|, \oplus$	Concatenation and bitwise XOR operations

Fig - 02 : Notations used in this work

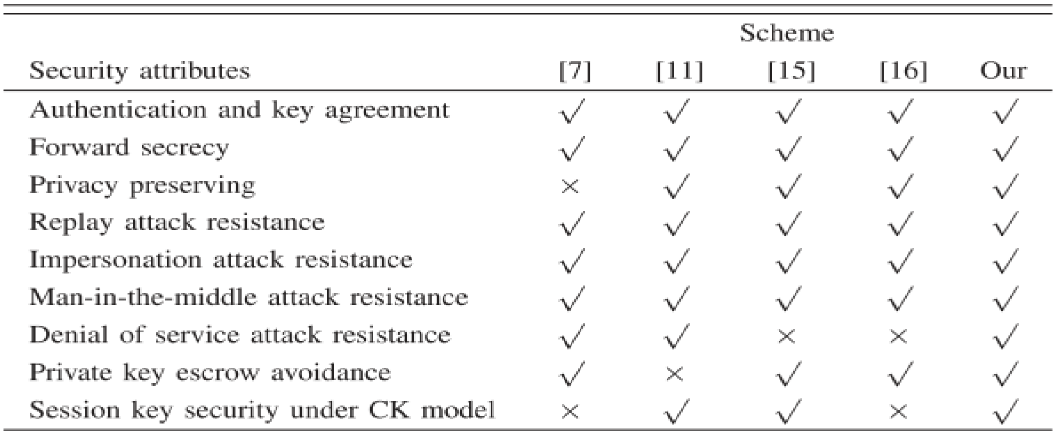


Smart meter (SM_i)		Service provider (SP_j)
Generates a random number $x \in \mathbb{Z}_n^*$		
$X = xG$		
$e = \overline{h}(X, SID_j)$		
$\overline{X} = X + eP_i$		
$T_1 = (x + es_i)P_s$		
$C_1 = (ID_i \| T_i \| X) \oplus H_1(T_1 \| SID_j \| t_1)$		
	$\xrightarrow{\overline{X}, C_1, t_1}$	
		Check $t_1^* - t_1$? $< \Delta t$, if true, proceed
		$T_1 = s_j \overline{X}$
		$ID_i \| T_i \| X = C_1 \oplus H_1(T_1 \| SID_j \| t_1)$
		$e = \overline{h}(X, SID_j)$
		Check $\overline{X}$? $= X + e(h(ID_i \| T_i)T_i + P_{pub})$, if true, proceed
		Generates a random number $y \in \mathbb{Z}_n^*$
		$d = \overline{h}(X, ID_i)$
		$\overline{Y} = (y + ds_j)G$
		$C_2 = \overline{Y} \oplus T_1$
		$A = H_2(SID_j \| ID_i \| T_1 \| C_2 \| t_2)$
		$sk = H_3((y + ds_j)\overline{X})$
	$\xleftarrow{C_2, A, t_2}$	
Check $t_2^* - t_2$? $< \Delta t$, if true, proceed		
Check A ? $= H_2(SID_j \| ID_i \| T_1 \| C_2 \| t_2)$		
$\overline{Y} = C_2 \oplus T_1$		
$sk = H_3((x + es_i)\overline{Y})$		

Fig - 03 : Summary of authenticated key agreement phase of our scheme.

Security Analysis :

Here is the security comparison between our proposed model and other models.



	Scheme				
Security attributes	[7]	[11]	[15]	[16]	Our
Authentication and key agreement	√	√	√	√	√
Forward secrecy	√	√	√	√	√
Privacy preserving	×	√	√	√	√
Replay attack resistance	√	√	√	√	√
Impersonation attack resistance	√	√	√	√	√
Man-in-the-middle attack resistance	√	√	√	√	√
Denial of service attack resistance	√	√	×	×	√
Private key escrow avoidance	√	×	√	√	√
Session key security under CK model	×	√	√	×	√

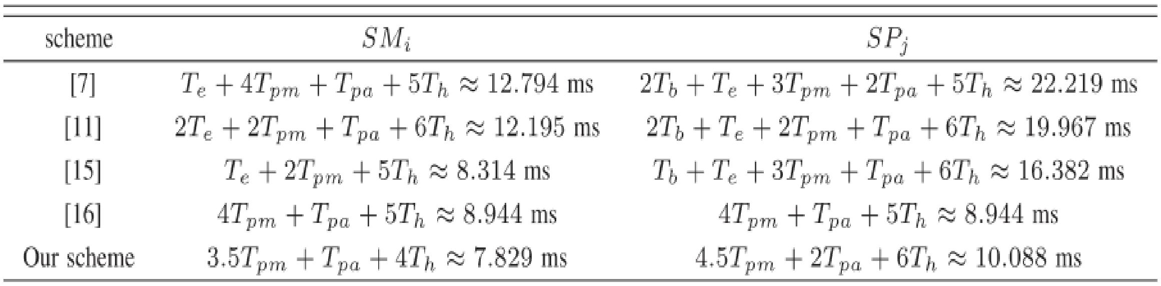
Fig-04 : SECURITY ATTRIBUTES COMPARISON

[7] An anonymous key distribution scheme designed by Tsai and Lo.

Tsai and Lo's scheme was pointed out by Odelu et al. [11] that it is unable to ensure the session key security and preserve the privacy of SM
credential.

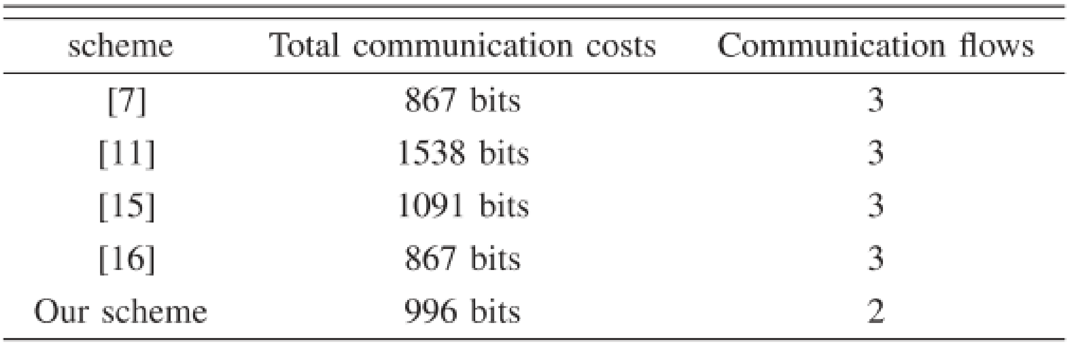
Chen et al. [15] and Abbasinezhad-Mood and Nikoohgadam [16] also, respectively, presented an authenticated key agreement scheme and a self-certificated key distribution scheme for smart grid.

Performance Evaluation :



scheme	SM_i	SP_j
[7]	$T_e + 4T_{pm} + T_{pa} + 5T_h \approx 12.794$ ms	$2T_b + T_e + 3T_{pm} + 2T_{pa} + 5T_h \approx 22.219$ ms
[11]	$2T_e + 2T_{pm} + T_{pa} + 6T_h \approx 12.195$ ms	$2T_b + T_e + 2T_{pm} + T_{pa} + 6T_h \approx 19.967$ ms
[15]	$T_e + 2T_{pm} + 5T_h \approx 8.314$ ms	$T_b + T_e + 3T_{pm} + T_{pa} + 6T_h \approx 16.382$ ms
[16]	$4T_{pm} + T_{pa} + 5T_h \approx 8.944$ ms	$4T_{pm} + T_{pa} + 5T_h \approx 8.944$ ms
Our scheme	$3.5T_{pm} + T_{pa} + 4T_h \approx 7.829$ ms	$4.5T_{pm} + 2T_{pa} + 6T_h \approx 10.088$ ms

Fig - 05 : COMPUTATION COSTS COMPARISON



scheme	Total communication costs	Communication flows
[7]	867 bits	3
[11]	1538 bits	3
[15]	1091 bits	3
[16]	867 bits	3
Our scheme	996 bits	2

Fig - 06 : COMMUNICATION COSTS COMPARISON

Implementation

To implement our work we use python programming language and Google colab as IDE.

```
import random
x = int(input("Enter a value for 'x': "))
z = int(input("Enter a value for 'z': "))
sm_pub = random.randint(x, z)
sp_pub = random.randint(x, z)

sm_private = random.randint(x, z)
sp_private = random.randint(x, z)

smart_meter_pub = sm_pub
service_prodiver_pub = sp_pub

print('')

print(f'Public Key for Smart Meter : {smart_meter_pub}')
print(f'Public Key for Service Provider :
{service_prodiver_pub}\n')

smart_meter_private = sm_private
print(f'The Private Key for Smart Meter :
{smart_meter_private}')

# Gets the generated key for Smart Meter
generated_key_sm = int(pow(service_prodiver_pub,
smart_meter_private, smart_meter_pub))

service_prodiver_private = sp_private
print(f'The Private Key for Service Provider :
{service_prodiver_private}\n')
```

```
# Gets the generated key for Service Provider.
generated_key_sp = int(pow(service_prodiver_pub,
service_prodiver_private, smart_meter_pub))

# Secret Key for Smart Meter
secret_key_sm = int(pow(generated_key_sp, smart_meter_private,
smart_meter_pub))

# Secret Key for Service Provider
secret_key_sp = int(pow(generated_key_sm,
service_prodiver_private, smart_meter_pub))

print(f'Secret Key for Smart Meter : {secret_key_sm}')
print(f'Secret Key for Service Provider : {secret_key_sp}')
```

- First we import a library called “random”. This library is used for generating Random numbers.

- Take two integer values from the user that will be used to set a range for the Random number.

- After that we have assigned different random numbers as public keys for the smart meter and the service provider.

- In the same way we assign different random numbers as private keys for the smart meter and the service provider.

- Then we print the public key of the smart metre and service provider.

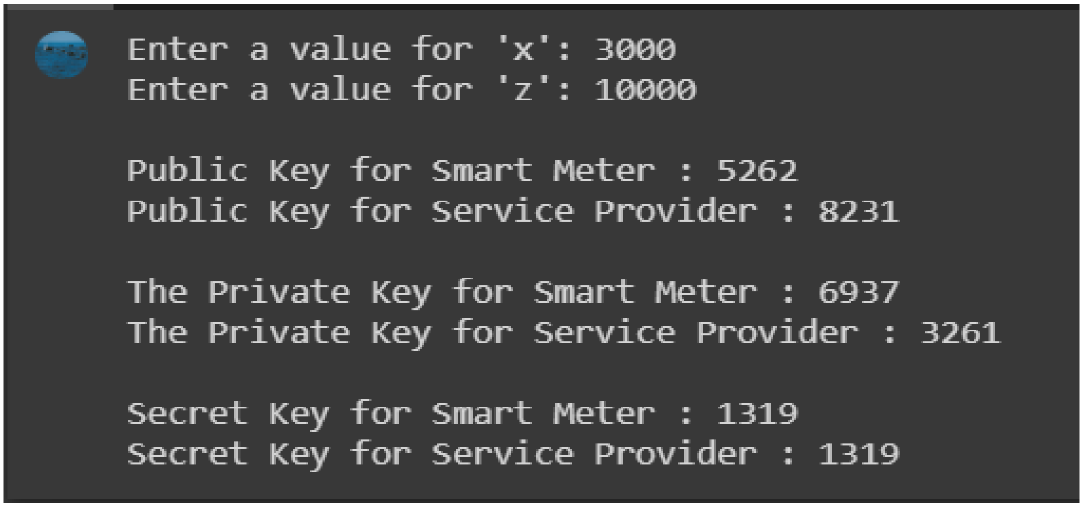
- In the same way we print the smart metre private key and also service providers private key.

- After that we have generated a special key for the smart metre and in the same way we generate another special key for the service provider.

- After that we have generated the secret keys for the smart metre and the service provider, these are the authenticated keys.

- And finally we print secret key values.

Output :



```
Enter a value for 'x': 3000
Enter a value for 'z': 10000

Public Key for Smart Meter : 5262
Public Key for Service Provider : 8231

The Private Key for Smart Meter : 6937
The Private Key for Service Provider : 3261

Secret Key for Smart Meter : 1319
Secret Key for Service Provider : 1319
```

So here in the output first we declare two integer values 3000 and 10000, which means the random number will be between 3000 to 10000. so we get our public key 5262 for the smart metre and we get our public key 8231 for the service provider. we get our private key 6937 for the smart metre and we get our private key 3261 for the service provider. And we get our secret key for the smart meter and service provider 1319.

Conclusion :

- Able to provide session key security in the widely accepted CK adversary model.
- Offer various anticipated security features.
- Behaves quite well in terms of the computation and communication overhead.
- More efficient than other public key cryptography based relevant schemes.