China Blockchain Conference

# Blockchain-Based Anonymous Rewarding Scheme for V2G Networks

Huaqun Wang

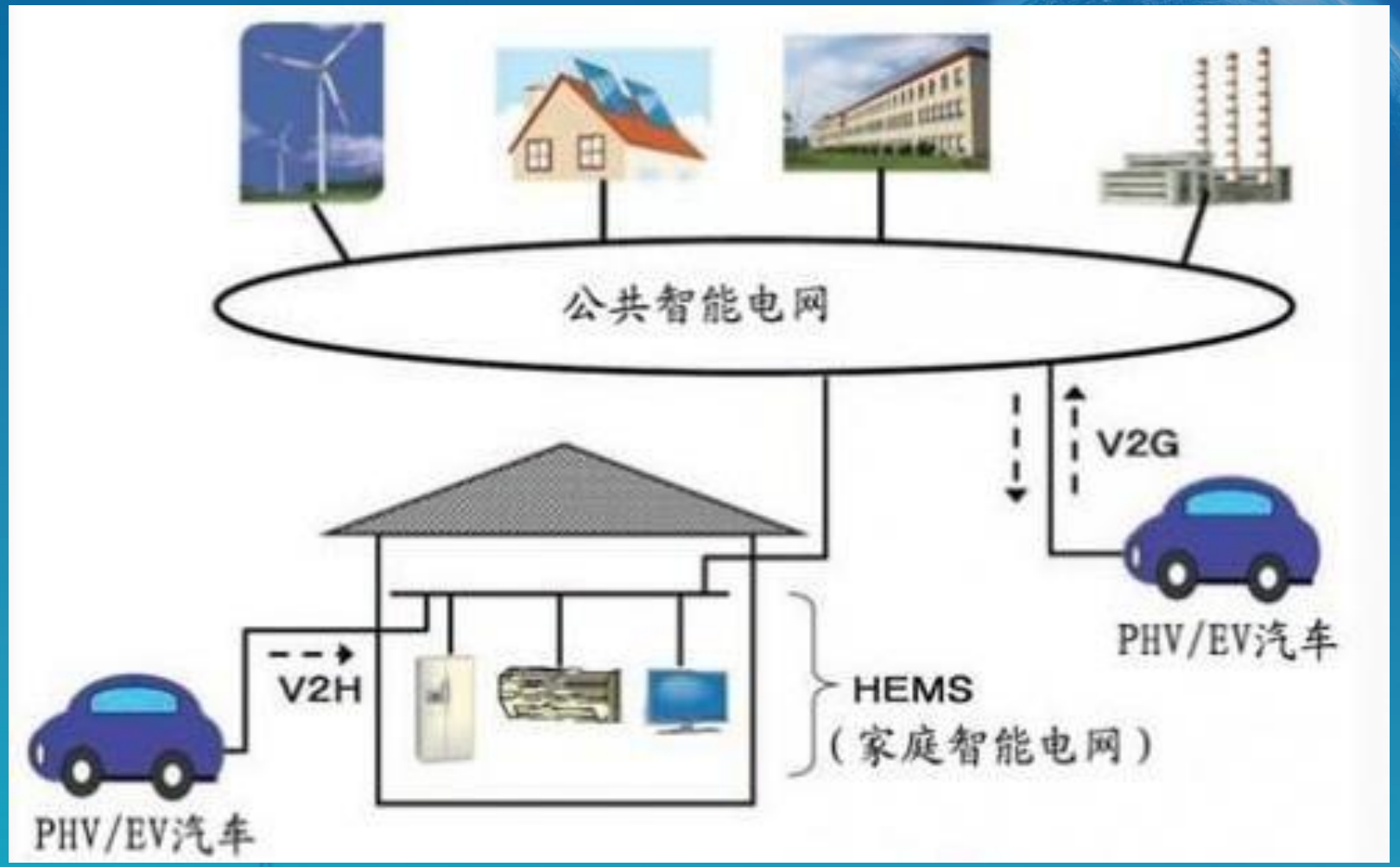
Nanjing University of Posts and Telecommunications

China Blockchain Conference

# Outline

- Motivation
- Two different PKCs
- System model and security model
- The concrete scheme
- Security analysis
- Conclusions

# 1.Motivation




V2G Networks

# 1. Motivation

- In order encourage more BVs to provide the services, it is necessary to reward the valid BVs.
- Privacy-preservation is an important concern for the BVs.

  → Anonymous rewarding
- Disputes between the BVs and the power grid.
- Furthermore, it is also important to preserve the bank account privacy of power grid.

# 2. Two different PKCs

- PKC on the Blockchain

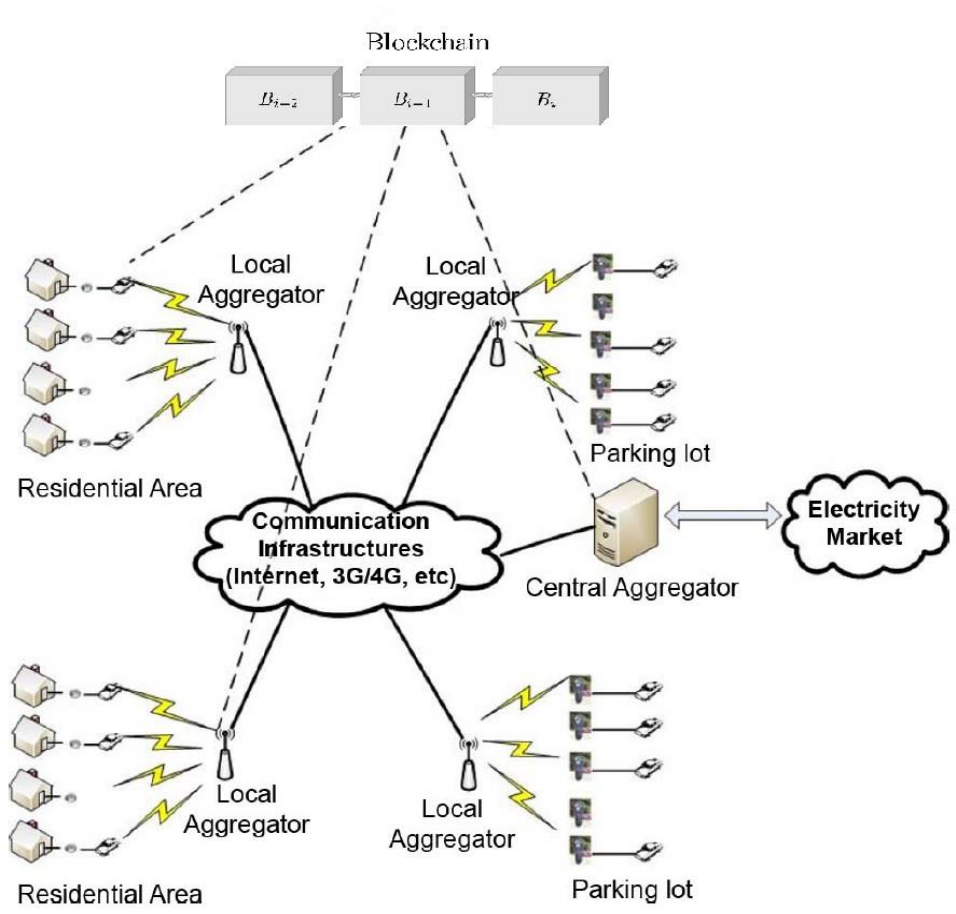
  ⟹ Elliptic curve cryptography for transaction

  ⟹ ring signature, Monero

- PKC in PKI

  ⟹ Bilinear group pair ⟹ Aggregate signature

# 3. System model and security model

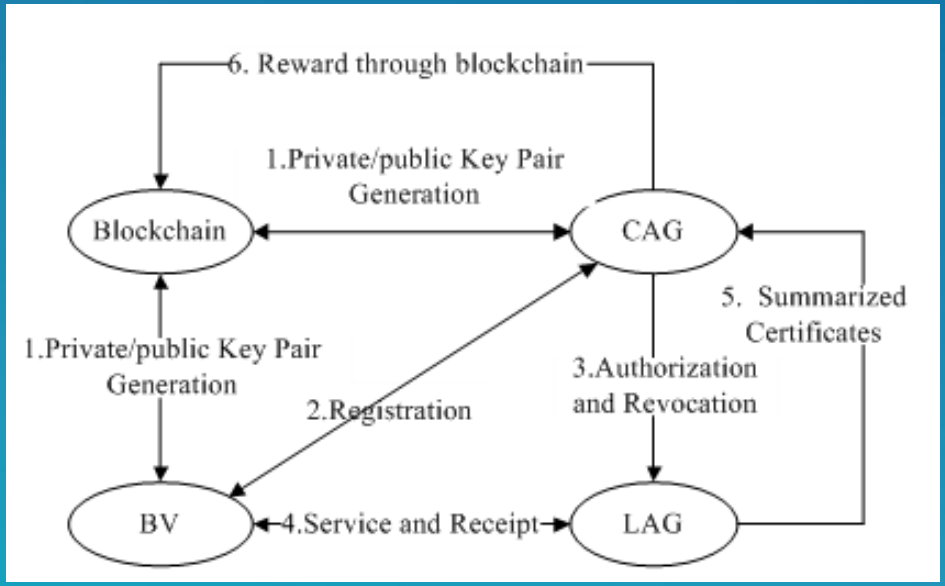



- CAG(central aggregator): It is an operator of the V2G networks.
- LAG (local aggregator) : It is also an operator of the V2G networks.
- BV (battery-powered vehicle)
- Blockchain: CAG sends the rewards to BVs through the blockchain.

# 3. System model and security model

- Mutual authentication among BV, CAG and LAG.
- Anonymity for BV. In the process of receiving BV's service, LAG cannot identify the BV's identity. In the process of authorization and rewarding, CAG cannot identify the BV's identity.
- Anonymity for CAG. Although BV accepts CAG's rewards, BV cannot identify CAG's account address on the blockchain.
- Unlinkability between payee address (i.e., BV) and payer address (i.e., CAG).

# 4. The concrete scheme: Setup




Architecture of our rewarding scheme

- Blockchain parameters: The elliptic curve $E$ on the finite field $\mathrm{F_q}$, base point G with order $\hat{l}$. $\mathrm{BV_i}$ picks $a_i, b_i$ (keep secret) and computes $\mathrm{A_i} = \mathrm{a_i G}, \mathrm{B_i} = \mathrm{b_i G}$. CAG picks its private key $x, y$ and computes its public key $\mathrm{X} = x\mathrm{G}, Y = y\mathrm{G}$.
- PKC in PKI: Let the bilinear map be $\mathrm{e}: \mathrm{G_1} \times G_1 \rightarrow G_2$ where $\mathrm{G_1}, G_2$ have prime order $p$. Let $P$ be a generator of $G_1$. CAG's private key/public key pair are $(z, Z)$. LAGi's private key/public key pair are $(l_i, L_i)$.
- Pick a secure signature/verification algorithm pair (*Sign*, *Verify*).

# 4. The concrete scheme: Contract-based authorization

Denote the agreed contract as $Cont_i$ between CAG and $BV_i$. CAG performs the following procedures to authorize $BV_i$:

- CAG generates the signature $\sigma_i$ for $Cont_i$: $\sigma_i = zH(Cont_i)$ ;
- CAG adds the address $(A_i, B_i)$ and period of service into $Tab$;
- CAG sends $(Cont_i, \sigma_i)$ to $BV_i$ ;
- CAG sends the updated table $Tab$ to all the LAGs.

# 4. The concrete scheme: Anonymous service and receipt

- $BV_i$ presents $(Cont_i, \sigma_i)$ to the $LAG_j$ in the local area. At some moment, $LAG_j$ receives a lot of pairs $(Cont_i, \sigma_i)$ where $i \in I$.
- $LAG_j$ picks $\alpha_i \in F_p$, $i \in I$ and verifies: $e(\sum \alpha_i \sigma_i, P) = e(\sum \alpha_i H(Cont_i), Z)$

If it does not hold, $LAG_j$ finds out the invalid pairs and reject them ; otherwise, goto the following procedure.

- If $BV_i$'s public key $(A_i, B_i)$ belongs to $Tab$ , $LAG_j$ accepts $BV_i$'s service;
- When $BV_i$ provides service, $LAG_j$ generates receipt: Denote the service and corresponding reward as $m_{i,j}$. $LAG_j$ computes $\overline{\sigma_{i,j}} = l_j H(m_{i,j}, A_i, B_i)$. $LAG_j$ sends $(m_{i,j}, A_i, B_i, \overline{\sigma_{i,j}})$ to CAG and $BV_i$, respectively.

# 4. The concrete scheme: Reward from CAG

- CAG receives the certificates $(m_{i,j}, A_i, B_i, \overline{\sigma_{i,j}})$ from $LAG_j$, where $i \in I$, j $\in J$.

CAG picks $\beta_{i,j} \in F_p$ and verifies:$e\left(\sum \beta_{i,j}\, \overline{\sigma_{i,j}},\ P\right) = \prod e(\beta_{i,j} H\left(m_{i,j}, A_i, B_i\right), L_j)$;

- For $BV_i$, CAG picks $r \in F_{\hat{i}}$ and computes $\widehat{P_i} = H_1(rA_i)G + B_i$.
- CAG computes $bal_i = \sum bal_{i,j}$ and $bal_c = bal - \sum bal_i$. CAG will transfer $bal_i$ to the address $\widehat{P_i}$ and $bal_c$ to the new address $\widehat{P_c}$.
- By using the similar method to Monero transaction, CAG realizes the above transactions. The detailed procedures are given below:

# 4. The concrete scheme: Reward from CAG

- CAG computes $A = H_2(Sign_z(m))$. Picks a random subset $S'$ of users' public key $P_i$ where $|S'| = n$, CAG's public key $P_s \in S'$. It computes $I = x_s H_2(P_s)$ , picks the random numbers $\{q_i | i = 1, \cdots, n\}$ and $\{\omega_i | i \neq s\}$ from $F_{\hat{l}}$. Then, it computes:

$$L_i = \begin{cases} q_i G & if \quad i = s \\ q_i G + \omega_i P_i & if \quad i \neq s \end{cases}$$

$$R_i = \begin{cases} q_i H_2(P_i) & if \quad i = s \\ q_i H_2(P_i) + \omega_i I & if \quad i \neq s \end{cases}$$

$$c = H_1(m, A, L_0, \cdots, L_n, R_0, \cdots, R_n)$$

$$c_i = \begin{cases} \omega_i & if \quad i \neq s \\ c - \sum_{i \neq s} c_i & if \quad i = s \end{cases}$$

$$r_i = \begin{cases} q_i & if \quad i \neq s \\ q_s - c_s x_s & if \quad i = s \end{cases}$$

The resulting signature is $\sigma = (I, A, c_0, c_1, \cdots, c_n, r_0, r_1, \cdots, r_n)$. CAG sends $bal_i$ to $\widehat{P_i}$ and $bal_c$ to $\widehat{P_c}$ with the signature $\sigma$. The query is sent to the blockchain.

# 4. The concrete scheme: Verification and Gain

- The verifier computes $L_i' = r_i G + c_i P_i, R_i' = r_i H_2(P_i) + c_i I$ and checks the following formula: $\sum c_i = H_1(m, A, L_0', L_1', \cdots, L_n', R_0', R_1', \cdots, R_n') \ mod \ \hat{l}$

If it does not hold, the signature is rejected; otherwise, the verifier goes to the next step.

- The verifier checks whether $I$ has been used in past signatures. If Yes, the signature is rejected; otherwise, accepts and finishes all the outputs.
- In order to gain the reward $bal_i$, $BV_i$ computes $x_i = H_1(a_i R) + b_i$ which satisfies $P_i = x_i G$. Thus, $BV_i$ gains the reward $bal_i$.

# 4. The concrete scheme: Solve the dispute

When $BV_i$ cannot find its rewards, it sends its receipt to CAG. CAG checks $BV_i$ 's receipt, if it is valid, CAG tells it the transaction information with the ring signature. If $BV_i$ thinks CAG is not the real signer, CAG performs the following procedure to show it is the real signer:

- CAG shows $BV_i$ the pre-image $\mathrm{Sign}_z(m)$;
- $BV_i$ verifies $\mathrm{Sign}_z(m)$. If it is Yes, $BV_i$ admits CAG is the real signer; otherwise, $BV_i$ denies CAG is the real signer.

# 4. The concrete scheme: BV revocation

- Case 1: When $LAG$ wants to reject $BV$'s service, $LAG$ refuses $BV$'s service request. When $CAG$ wants to reject $BV$'s service, $CAG$ updates the table $Tab$. $CAG$ adds the revocation information to $Tab$ and sends it to $LAG$s.
- Case 2: When BV would like to be revoked, BV would refuse to provide the service for GAG. It can also informs this information to CAG. CAG updates the table $Tab$ and sends it to $LAG$s.
- Case 3: When BV's service expired, it realizes the BV revocation.

# 5. Security analysis

*Theorem 1 (Unforgeability):* In our BBARS scheme, $BV_i$ 's authorization from $CAG$, the receipt and certificate from $LAG$, and the ring signature from $CAG$ satisfy the unforgeability.

*Theorem 2 (Anonymity for BV):* In our BBARS scheme, BV is unconditionally anonymous. Even if the adversary's computing power is infinite, it cannot identify BV's real identity.

# 5. Security analysis

*Theorem 3 (Anonymity for CAG): Our BBARS scheme satisfies anonymity for CAG.*

*Theorem 4 (Unlinkability):* Our BBARS scheme can satisfy the unlinkability and solve the dispute.

# 5. Conclusions

In this paper, we studied the privacy protection and precise reward architecture for V2G networks in the smart grid. By taking use of the properties of blockchain, we propose the novel concept of blockchain-based anonymous rewarding for V2G in the smart grid. The system model and security model are formalized. Based on the aggregated signature, ring signature and blockchain, we design the first BBARS scheme. The analysis and implementation show that our BBARS scheme is provably secure and efficient.

China Blockchain Conference

# 谢谢！

wanghuaqun@aliyun.com