Privacy-Enhanced Data Aggregation Scheme Against Internal Attackers in Smart Grid

Chun-I Fan (范俊逸), Shi-Yuan Huang, and Yih-Loong Lai

National Sun Yat-sen University
國立中山大學

IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS
VOL. 10, NO. 1, 2014

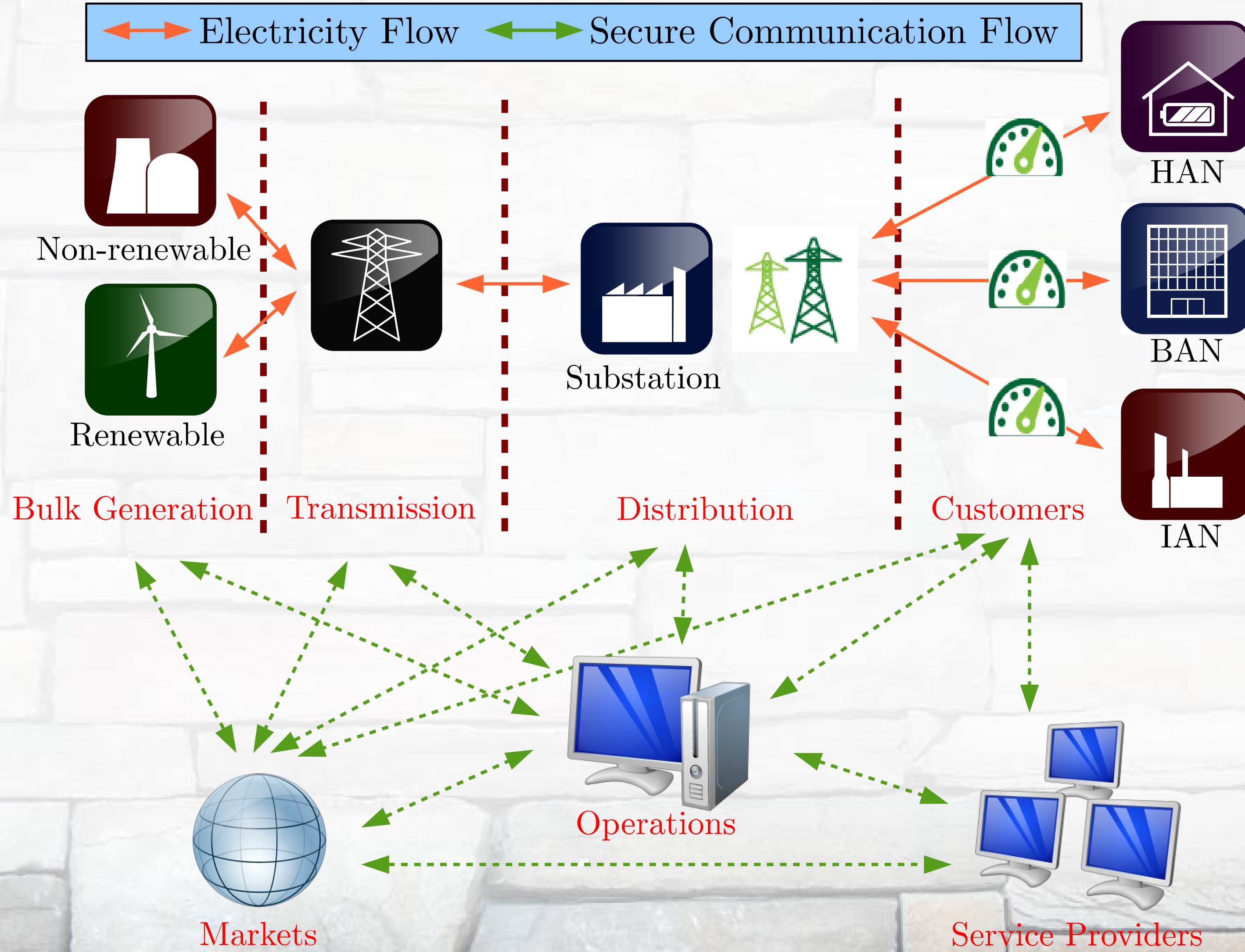
The final publication is available at http://ieeexplore.ieee.org.

A correction to the final paper has been published in ResearchGate (DOI: 10.13140/RG.2.1.4006.8649).

Introduction

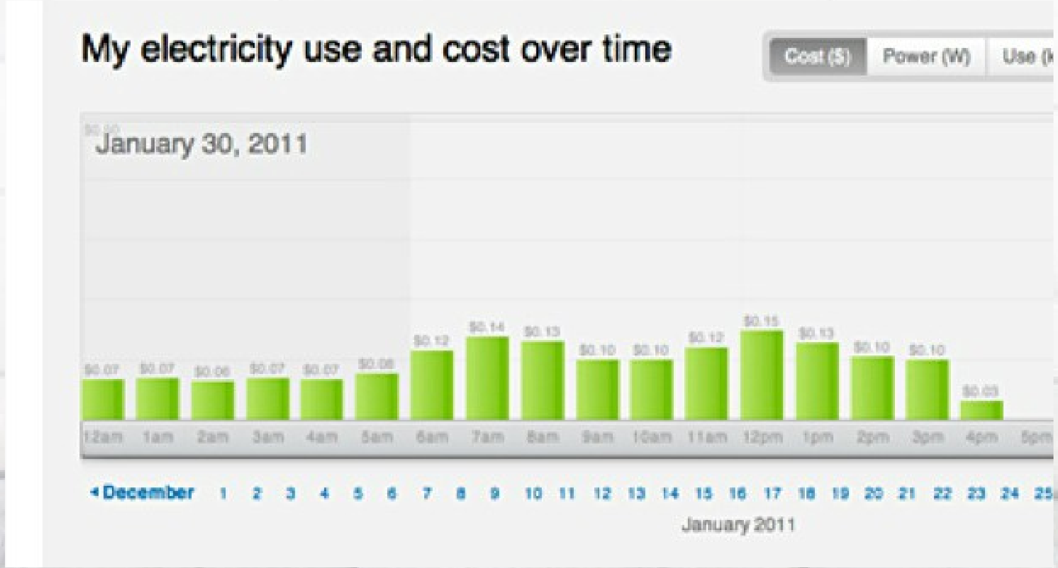
- The development of smart grids becomes a global trend
  - Smart grids can handle bi-directional energy flows better
  - Reduce energy consumption

Introduction



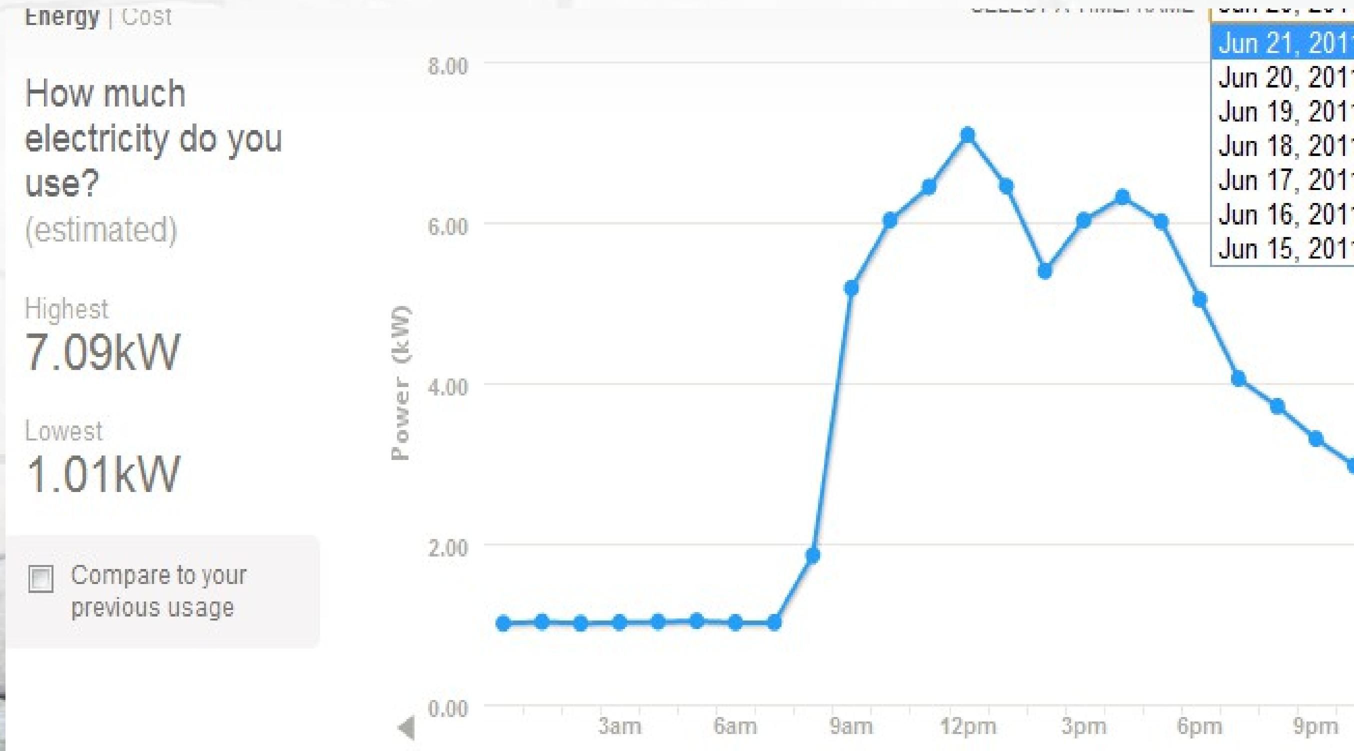
Introduction

- Smart grid applications
  - Know how much electricity users have consumed
  - Get the average electricity consumption data



Introduction

- The privacy issues of smart grid communication
  - Meter readings are sensitive
  - Attackers may catch the consumption data to derive users' lifestyles



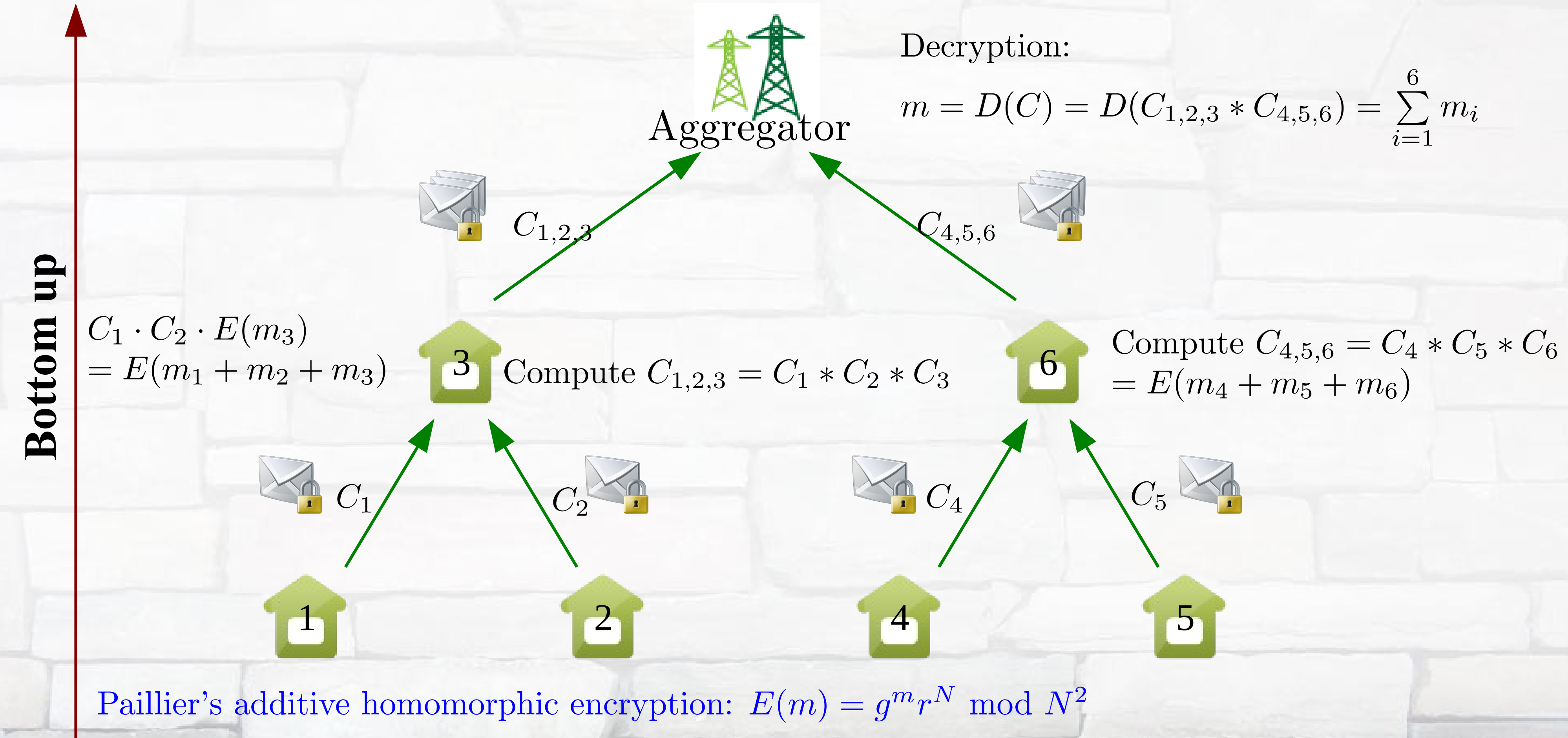
Introduction

- Prevent anyone from stealing or tampering with the data
  - External attackers: Hackers
  - Internal attackers: Electricity suppliers

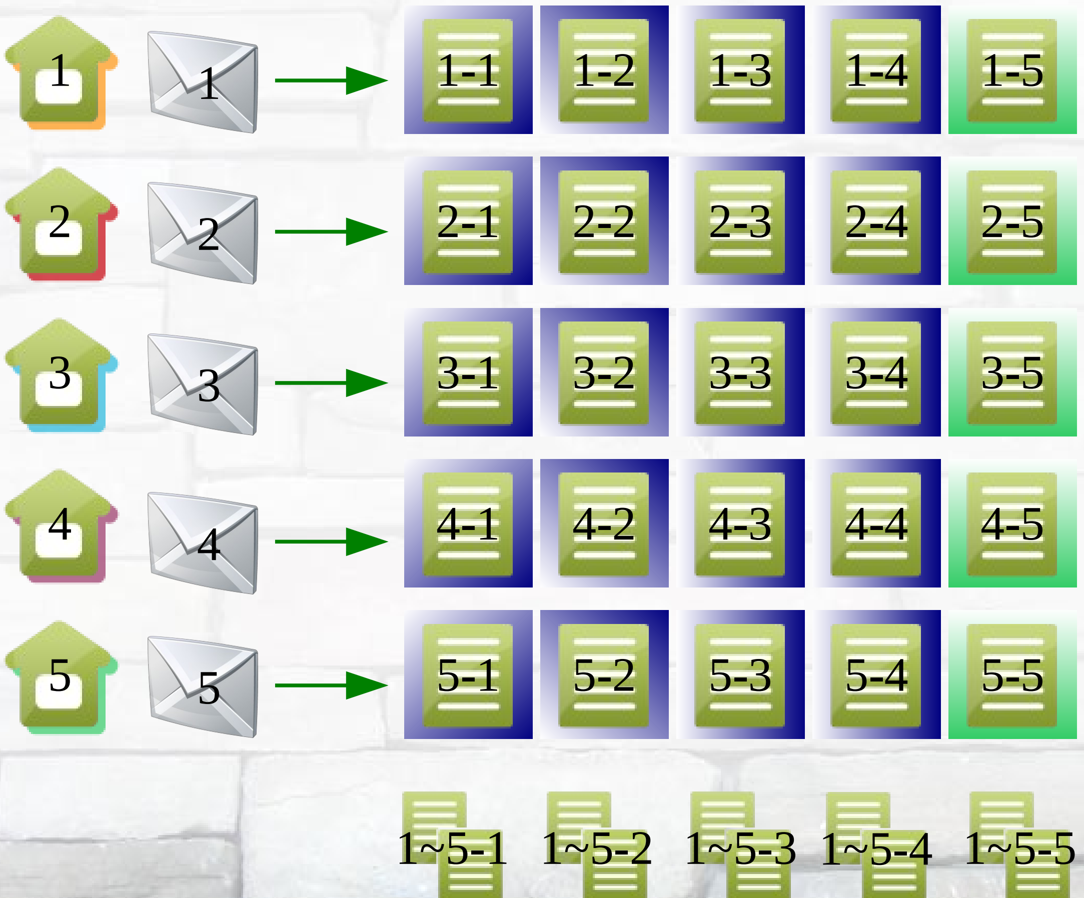
Related Works

- Li *et al.*'s *Scheme*
  - Secure Information Aggregation for Smart Grids Using Homomorphic Encryption
- Garcia *et al.*'s *Scheme*
  - Privacy-Friendly Energy-Metering Via Homomorphic Encryption
- Lu *et al.*'s Scheme
  - EPPA: An Efficient and Privacy-Preserving Aggregation Scheme for Secure Smart Grid Communication
- Petrlic's Scheme
  - A Privacy-Preserving Concept for Smart Grids

Li *et al.*'s Scheme



Garcia *et al.*'s Scheme



$$m_1 \equiv_N m_{1-1} + m_{1-2} + m_{1-3} + m_{1-4} + m_{1-5}$$

$$m_2 \equiv_N m_{2-1} + m_{2-2} + m_{2-3} + m_{2-4} + m_{2-5}$$

$$m_3 \equiv_N m_{3-1} + m_{3-2} + m_{3-3} + m_{3-4} + m_{3-5}$$

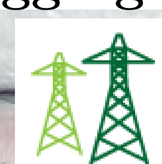
$$m_4 \equiv_N m_{4-1} + m_{4-2} + m_{4-3} + m_{4-4} + m_{4-5}$$

$$m_5 \equiv_N m_{5-1} + m_{5-2} + m_{5-3} + m_{5-4} + m_{5-5}$$

Garcia *et al.'s* Scheme

$$
\begin{array}{ccccc}
E_1(m_{1-1}) & E_2(m_{1-2}) & E_3(m_{1-3}) & E_4(m_{1-4}) & E_5(m_{1-5}) \\
E_1(m_{2-1}) & E_2(m_{2-2}) & E_3(m_{2-3}) & E_4(m_{2-4}) & E_5(m_{2-5}) \\
E_1(m_{3-1}) & E_2(m_{3-2}) & E_3(m_{3-3}) & E_4(m_{3-4}) & E_5(m_{3-5}) \\
E_1(m_{4-1}) & E_2(m_{4-2}) & E_3(m_{4-3}) & E_4(m_{4-4}) & E_5(m_{4-5}) \\
E_1(m_{5-1}) & E_2(m_{5-2}) & E_3(m_{5-3}) & E_4(m_{5-4}) & E_5(m_{5-5}) \\
\Downarrow & \Downarrow & \Downarrow & \Downarrow & \Downarrow \\
\prod_{Y=1}^{5} E_1(m_{Y-1}) & \prod_{Y=1}^{5} E_2(m_{Y-2}) & \prod_{Y=1}^{5} E_3(m_{Y-3}) & \prod_{Y=1}^{5} E_4(m_{Y-4}) & \prod_{Y=1}^{5} E_5(m_{Y-5}) \\
\Downarrow & \Downarrow & \Downarrow & \Downarrow & \Downarrow \\
D_1(\cdot) & D_2(\cdot) & D_3(\cdot) & D_4(\cdot) & D_5(\cdot) \\
\Downarrow & \Downarrow & \Downarrow & \Downarrow & \Downarrow \\
\sum_{Y=1}^{5} m_{Y-1} & \sum_{Y=1}^{5} m_{Y-2} & \sum_{Y=1}^{5} m_{Y-3} & \sum_{Y=1}^{5} m_{Y-4} & \sum_{Y=1}^{5} m_{Y-5} \\
\searrow & \searrow & \downarrow & \swarrow & \swarrow
\end{array}
$$

Aggregator



$$\sum_{W=1}^{5} \left(\sum_{Y=1}^{5} m_{Y-W} \right) \Rightarrow \sum_{i=1}^{5} m_i$$

Lu *et al.*'s Scheme

1 $C_1 = g_1^{d_{1,1}} g_2^{d_{1,2}} \dots g_L^{d_{1,L}} r_1^N \bmod N^2$ $\searrow$

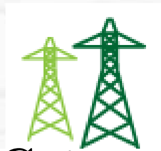
2 $C_2 = g_1^{d_{2,1}} g_2^{d_{2,2}} \dots g_L^{d_{2,L}} r_1^N \bmod N^2$ $\searrow$

3 $C_3 = g_1^{d_{3,1}} g_2^{d_{3,2}} \dots g_L^{d_{3,L}} r_1^N \bmod N^2$ $\longrightarrow$ Gateway $C = \prod_{i=1}^{5} C_i \bmod N^2$

4 $C_4 = g_1^{d_{4,1}} g_2^{d_{4,2}} \dots g_L^{d_{4,L}} r_1^N \bmod N^2$ $\nearrow$

5 $C_5 = g_1^{d_{5,1}} g_2^{d_{5,2}} \dots g_L^{d_{5,L}} r_1^N \bmod N^2$ $\nearrow$

Lu *et al.*'s Scheme



Aggregate

$$C = g^{M} R^{N} \bmod N^2$$

$\Downarrow$ (Paillier's decryption)

$$M = a_1 \sum_{i=1}^{5} d_{i,1} + a_2 \sum_{i=1}^{5} d_{i,2} + \ldots + a_L \sum_{i=1}^{5} d_{i,L}$$

(super-increasing
$\Downarrow$ sequence decoding)

The aggregated data

$$C = \prod_{i=1}^{5} C_i \bmod N^2$$

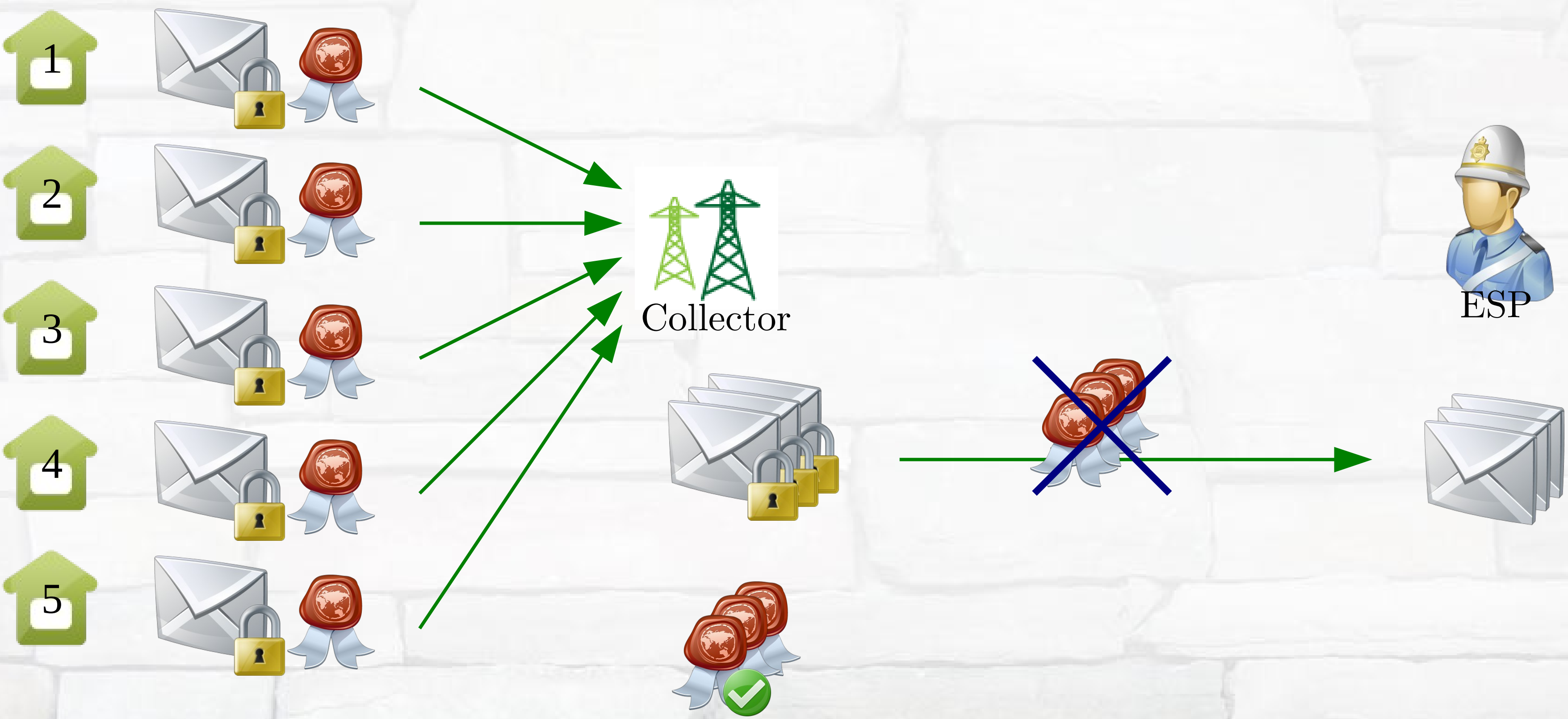
$$= \prod_{i=1}^{5} g_1^{d_{i,1}} g_2^{d_{i,2}} \ldots g_L^{d_{i,L}} r_i^N \bmod N^2$$

$$= g_1^{\sum_{i=1}^{5} d_{i,1}} g_2^{\sum_{i=1}^{5} d_{i,2}} \ldots g_L^{\sum_{i=1}^{5} d_{i,L}} \left(\prod_{i=1}^{5} r_i\right)^N \bmod N^2$$

$$= g^{a_1 \sum_{i=1}^{5} d_{i,1}} g^{a_2 \sum_{i=1}^{5} d_{i,2}} \ldots g^{a_L \sum_{i=1}^{5} d_{i,L}} \left(\prod_{i=1}^{5} r_i\right)^N \bmod N^2$$

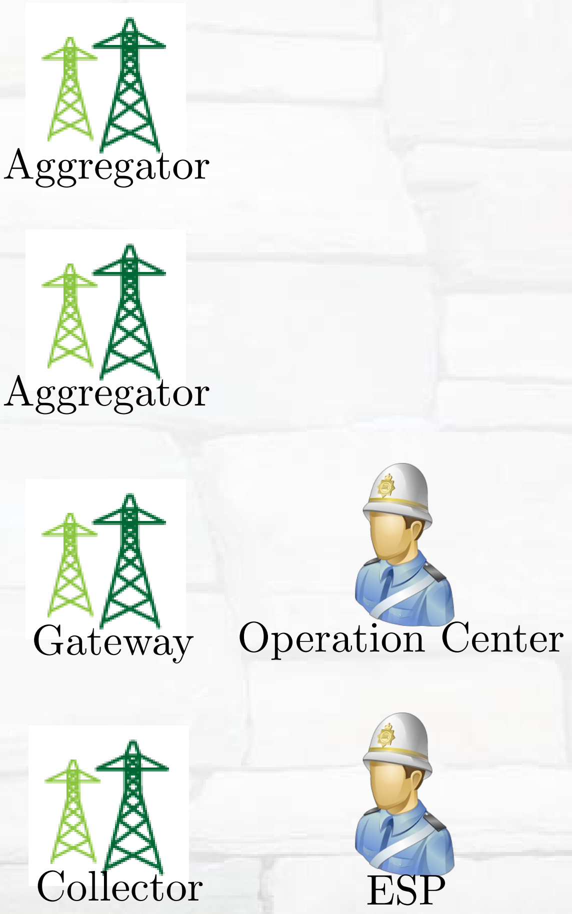
$$= g^{a_1 \sum_{i=1}^{5} d_{i,1} + a_2 \sum_{i=1}^{5} d_{i,2} + \ldots + a_L \sum_{i=1}^{5} d_{i,L}} \left(\prod_{i=1}^{5} r_i\right)^N \bmod N^2$$

Petrlic's Scheme

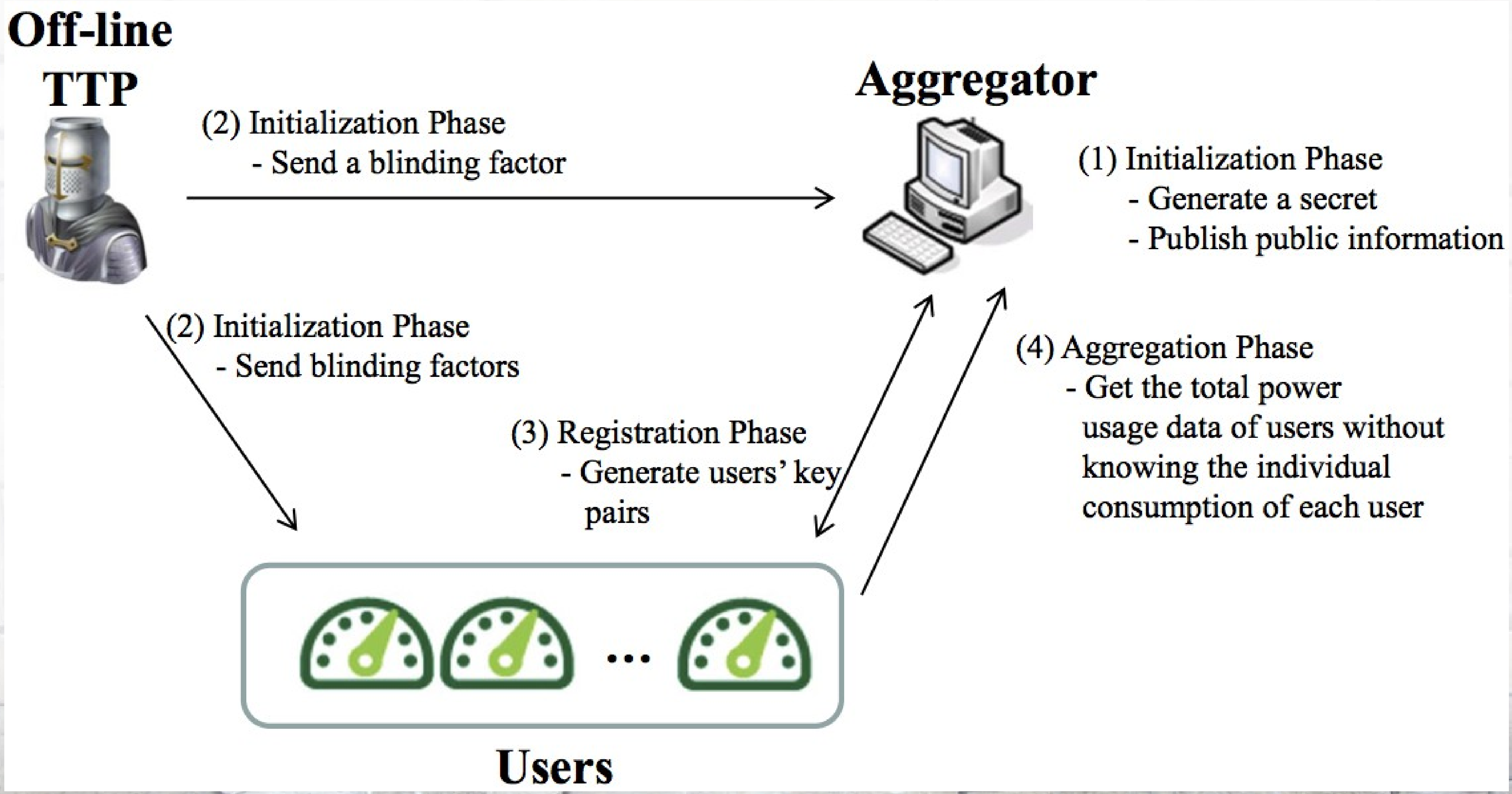


Internal Attackers

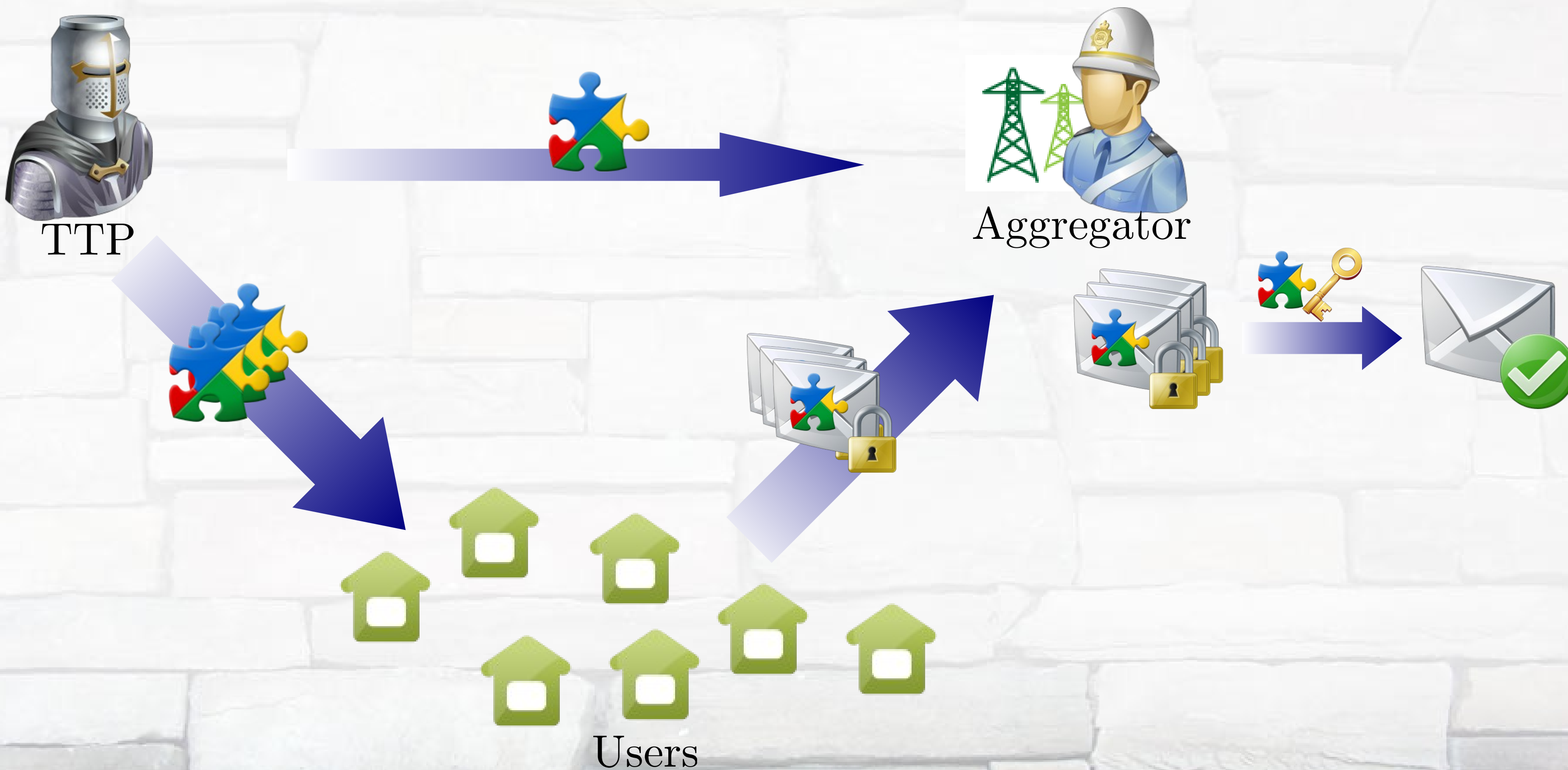
- Li *et al.*'s scheme
- Garcia *et al.*'s scheme
- Lu *et al.*'s scheme
- Petrlic's scheme



The Proposed Scheme



The Core Idea



The Proposed Scheme

- Initialization Phase
- Registration Phase
- Aggregation Phase
- Remark (Tree-Based Aggregation)

Initialization Phase

1. e: $\mathbb{G}_1 \times \mathbb{G}_1 \longrightarrow \mathbb{G}_T$, and $\mathbb{G}_1, \mathbb{G}_T$ are GDH groups with prime order q
2. $\mathbb{G}'_1$: a multiplicative group with order N where $N = q_1 * q_2$
3. H_0: a one-way hash function, $H_0 : \{0,1\}^* \longrightarrow \mathbb{Z}_q^*$
4. H_1: a one-way hash function, H_1: $\{0,1\}^* \longrightarrow \mathbb{G}_1$
5. H_2: a one-way hash function, H_2: $\{0,1\}^* \longrightarrow \mathbb{G}'_1$
6. H_3: a one-way hash function, $H_3 : \mathbb{G}_1 \longrightarrow \mathbb{Z}_q^*$
7. t: the time when the aggregator needs to aggregate the power usage data
8. U_i's: the neighbor users, where $i = 1, 2, \ldots, n$
9. ID_i: the identity of U_i
10. π_i: the blinding factor of U_i
11. x_i: the private key of U_i
12. Y_i: the public key of U_i

Initialization Phase

Aggregator:

q, q_1, q_2: three large primes

$\mathbb{G}_1'$: a group with order $N = q_1 q_2$; $(g_0, u) \in_R \mathbb{G}_1'^2$; $h = u^{q_2} \in \mathbb{G}_1'$

$\mathbb{G}_1$: a GDH group with order q and generator g_1

$\{N, q, g_0, g_1, u, h\}$: public keys; $\{q_1, q_2\}$: secret keys

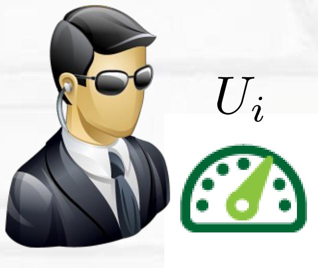
TTP:

Choose $\{\pi_0, \pi_1, \ldots, \pi_n\}$ at random such that $\sum_{i=0}^{n} \pi_i \equiv 0 \pmod{N}$.

Send π_0 to the aggregator in a secure manner.

Send π_i to U_i securely for $i = 1, 2, \ldots, n$.

Registration Phase



U_i

Aggregator

Private key: $x_i \in_R \mathbb{Z}_q^*$

Public key: $Y_i = g_1^{x_i}$

$r_i \in_R \mathbb{Z}_q^*$

$\alpha_i = g_1^{H_0(r_i \| ID_i)}$

$\beta_i = H_0(r_i \| ID_i) - x_i H_3(\alpha_i || Y_i) \bmod q$

$\{Y_i, \alpha_i, \beta_i, ID_i\}$ →

Check $\alpha_i = g_1^{\beta_i} Y_i^{H_3(\alpha_i || Y_i)}$

Publish $\{Y_i, \alpha_i, \beta_i, ID_i\}$

*Correction:

$\{Y_i, \alpha_i, \beta_i, r_i, ID_i\} \rightarrow \{\mathrm{Y}_i, \alpha_i, \beta_i, ID_i\}$

*ResearchGate (DOI: 10.13140/RG.2.1.4006.8649)

Aggregation Phase

U_i

$r'_i \in \mathbb{Z}_N^*$

$CT_i = g_0^{m_i}(H_2(t)h^{r'_i})^{\pi_i}$

$\sigma_i = H_1(t\|CT_i)^{x_i}$

$\{\sigma_i, CT_i\}$ →

Aggregator

Collect $\{\sigma_i, CT_i\}$ for $i = 1, 2, \ldots, n$

Generate $\delta_i \in_R \mathbb{Z}_q^*$ where $i = 1, 2, \ldots, n$

Check $e(\prod_{i=1}^{n} \sigma_i^{\delta_i}, g_1) = \prod_{i=1}^{n} e(H_1(t\|CT_i)^{\delta_i}, Y_i)$

Compute $CT = \prod_{i=1}^{n} CT_i$ and $\bar{g} = g_0^{q_1}$

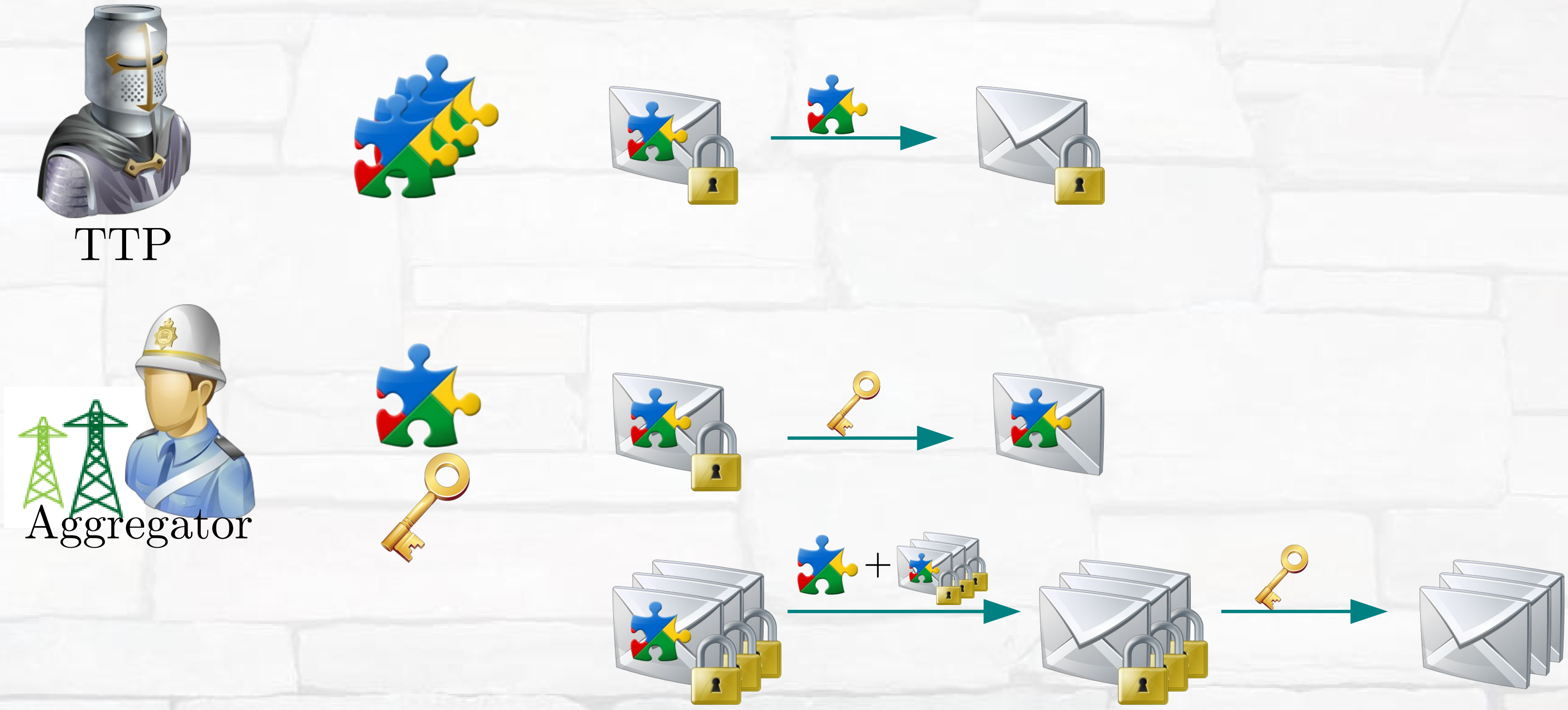
$$V = H_2(t)^{\pi_0} \cdot \prod_{i=1}^{n} CT_i$$

$$= H_2(t)^{\pi_0+\pi_1+\ldots+\pi_n} \cdot \prod_{i=1}^{n} g_0^{m_i} \cdot h^{r'_i \cdot \pi_i}$$

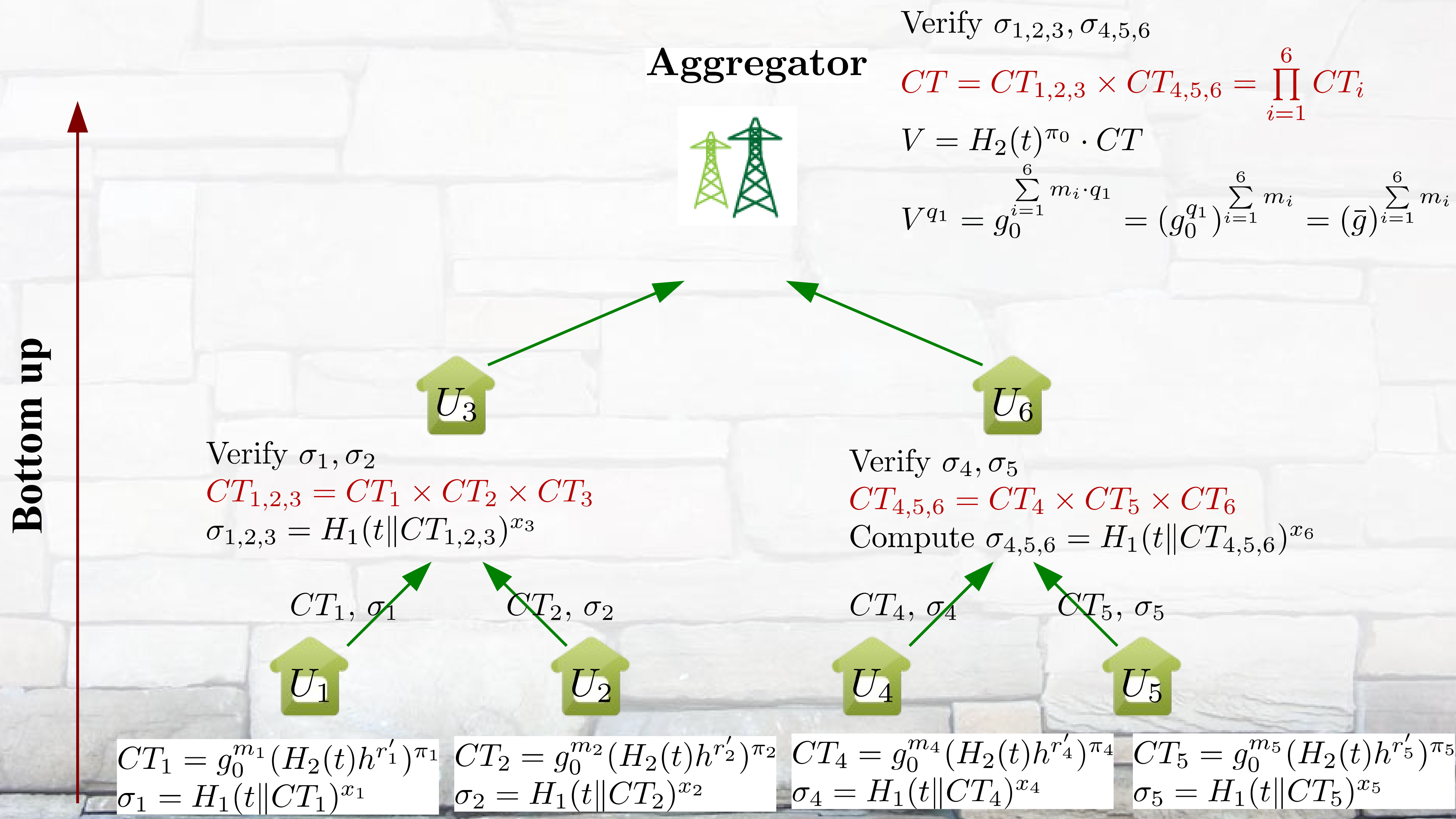
$$V^{q_1} = g_0^{\sum_{i=1}^{n} m_i \cdot q_1} = (g_0^{q_1})^{\sum_{i=1}^{n} m_i} = (\bar{g})^{\sum_{i=1}^{n} m_i}$$

Get $\sum_{i=1}^{n} m_i$ by Pollard's lambda method

Resisting Internal Attackers



Remark (Tree-Based Aggregation)



Comparison

	Our Scheme	Li *et al.*'s Scheme	Garcia *et al.*'s Scheme	Lu *et al.*'s Scheme	Petrlic's Scheme
Against External Attackers	Yes	Yes	Yes	Yes	Yes
Against Internal Attackers	**Yes**	No	No	No	No †
Data Integrity	Yes	No	No	Yes	Yes
Secure Batch Verification	**Yes**	N/A ‡	N/A ‡	No	N/A ‡
On/Off-line TTP	Off-line	No	No	On-line	No
Formal Proof	Yes	No	Yes	Yes	No

† : The author claimed that it can resist internal attackers, but it used an administration approach, not a cryptographic technique.
‡ : No batch verification in the scheme

Security Proofs

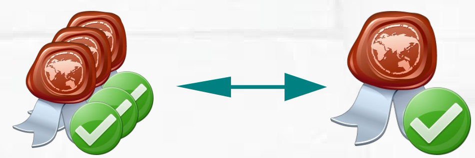
- Semantic Security

Ciphertext: $CT_i = g_0^{m_i}(H_2(t)h^{r'_i})^{\pi_i}$

- Unforgeability

Signature: $\sigma_i = H_1(t\|CT_i)^{x_i}$

- Batch Verification Security



Batch Verification:

$$e(\prod_{i=1}^{n} \sigma_i^{\delta_i}, g_1) = \prod_{i=1}^{n} e(H_1(t\|CT_i)^{\delta_i}, Y_i)$$

Semantic Security

$\mathcal{G}(\tau) \rightarrow (q_1, q_2, \mathbb{G}'_1, \mathbb{G}'_2, e')$ where $\mathbb{G}'_1, \mathbb{G}'_2$ are with order $N = q_1 q_2$

The subgroup decision problem :

- Given $\{N, \mathbb{G}'_1, \mathbb{G}'_2, e'\}$ and an element $x \in \mathbb{G}'_1$,
  - if the order of x is q_1, output "1"
  - otherwise, output "0"

The problem is to decide if an element x is in a subgroup of $\mathbb{G}'_1$ without knowing the factorization of the group order N.

Semantic Security

Adversary

Simulator

$N, \mathbb{G}_1', \mathbb{G}_2', e', g_0, x, H_2$ (Simulator → Adversary)

m_0, m_1 (Adversary → Simulator)

Choose $b \in_R \{0,1\}$

$C \leftarrow g_0^{m_b} H_2(t)^{\pi} x^{r\pi}$ (Simulator → Adversary)

Output $b' \in \{0,1\}$

b' (Adversary → Simulator)

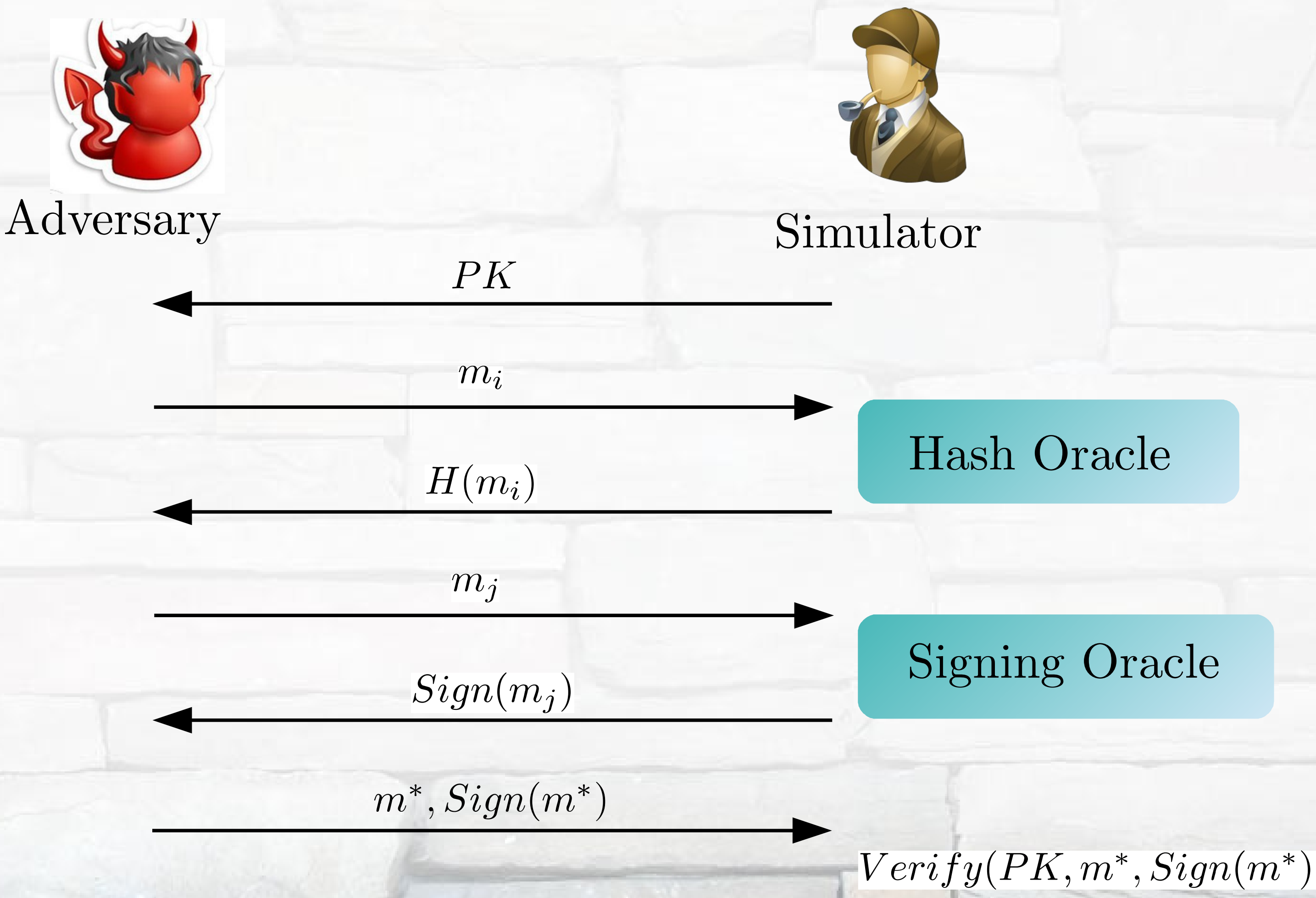
$\Pr[b' = b] = 1/2 + \epsilon(\tau)$

Unforgeability

Consider a generator g in a multiplicative cyclic group $\mathbb{G}$ with prime order p. We discuss two problems on $\mathbb{G}$:

- Decisional Diffie-Hellman Problem
  For $a, b, c \in \mathbb{Z}_p^*$, given (g, g^a, g^b, g^c), determine whether $c = ab$
- Computational Diffie-Hellman Problem
  For $a, b \in \mathbb{Z}_p^*$, given (g, g^a, g^b), compute g^{ab}

Unforgeability

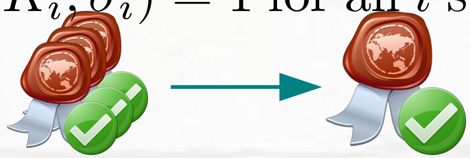


Unforgeability

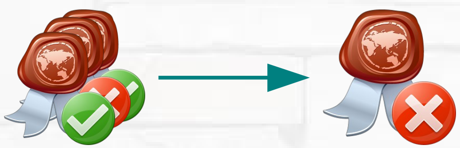
	PK	Hash Oracle	Signing Oracle	Challenge	Probability
Game 1	$Y \leftarrow g^a$	$h_i \leftarrow g^{r_i}$	$\sigma_i \leftarrow (g^a)^{r_i}$	$Verify(Y, M^*, \sigma_i^*)$	ϵ
Game 2	$Y \leftarrow g^a$	$h_i \leftarrow g^{r_i}$	$\sigma_i \leftarrow (g^a)^{r_i}$	$Verify(Y, M^*, \sigma_i^*) \wedge s_i^* = 1$	$\zeta\epsilon$
Game 3	$Y \leftarrow g^a$	$h_i \leftarrow g^{r_i}$	$s_i \in \{0, 1\}$ $\sigma_i \leftarrow (g^a)^{r_i}$	$Verify(Y, M^*, \sigma_i^*) \wedge s_i^* = 1$ $\wedge$ all of $s_i = 0$	$\zeta\epsilon \cdot (1-\zeta)^{q_s}$
Game 4	$Y \leftarrow g^a$	$h_i \leftarrow g^{r_i}$	$\sigma_i \leftarrow (g^a)^{r_i}$ $s_i = 1$: halt	$Verify(Y, M^*, \sigma_i^*) \wedge s_i^* = 1$ $\wedge$ all of $s_i = 0$	$\zeta\epsilon \cdot (1-\zeta)^{q_s}$
Game 5	$Y \leftarrow g^a$	$s_i = 0$: $h_i \leftarrow g^{r_i}$ $s_i = 1$: $h_i \leftarrow g^b g^{r_i}$	$s_i = 0$: $\sigma_i \leftarrow (g^b)^{r_i}$ $s_i = 1$: halt	$Verify(Y, M^*, \sigma_i^*) \wedge s_i^* = 1$ $\wedge$ all of $s_i = 0$	$\zeta\epsilon \cdot (1-\zeta)^{q_s}$
Game 6	$Y \leftarrow g^a$	$s_i = 0$: $h_i \leftarrow g^{r_i}$ $s_i = 1$: $h_i \leftarrow g^b g^{r_i}$	$s_i = 0$: $\sigma_i \leftarrow (g^b)^{r_i}$ $s_i = 1$: halt	$Verify(Y, M^*, \sigma_i^*) \wedge s_i^* = 1$ $\wedge$ all of $s_i = 0$ Output $\sigma_i^* / (g^a)^{r_i^*}$=g^{ab}	$\zeta\epsilon \cdot (1-\zeta)^{q_s}$

Batch Verification Security

- If $Verify(m_i, PK_i, \sigma_i) = 1$ for all i's in $[1, n]$, $Batch((m_i, PK_i, \sigma_i)$, for $i \in [1, n]) = 1$

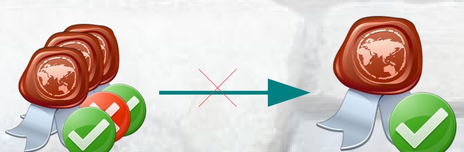


- If $Verify(m_i, PK_i, \sigma_i) = 0$ for some i in $[1, n]$, $Batch((m_i, PK_i, \sigma_i)$, for $i \in [1, n]) = 0$



Assume that an adversary tampers with some valid signatures and let the batch verification be valid (event E) as follows:

- When $Verify(m_i, PK_i, \sigma_i) = 0$ for some i's in $[1, n]$, $Batch((m_i, PK_i, \sigma_i)$, for $i \in [1, n]) = 1$ is with negligible probability



Conclusion

- The proposed scheme is the first one that can resist internal attackers in smart grids
- It ensures data integrity and provides secure batch verification for efficient verification
- We have also designed a tree-based aggregation variant for the wireless mesh network architecture

Future Works

- Eliminate the offline trusted third party
- Integrate the proposed scheme into the time-of-use billing system to protect user consumption information
- Apply the proposed approach to the other privacy-preserving protocols in smart grids

Applied Cryptology Laboratory
Department of Computer Science and Engineering
National Sun Yat-sen University