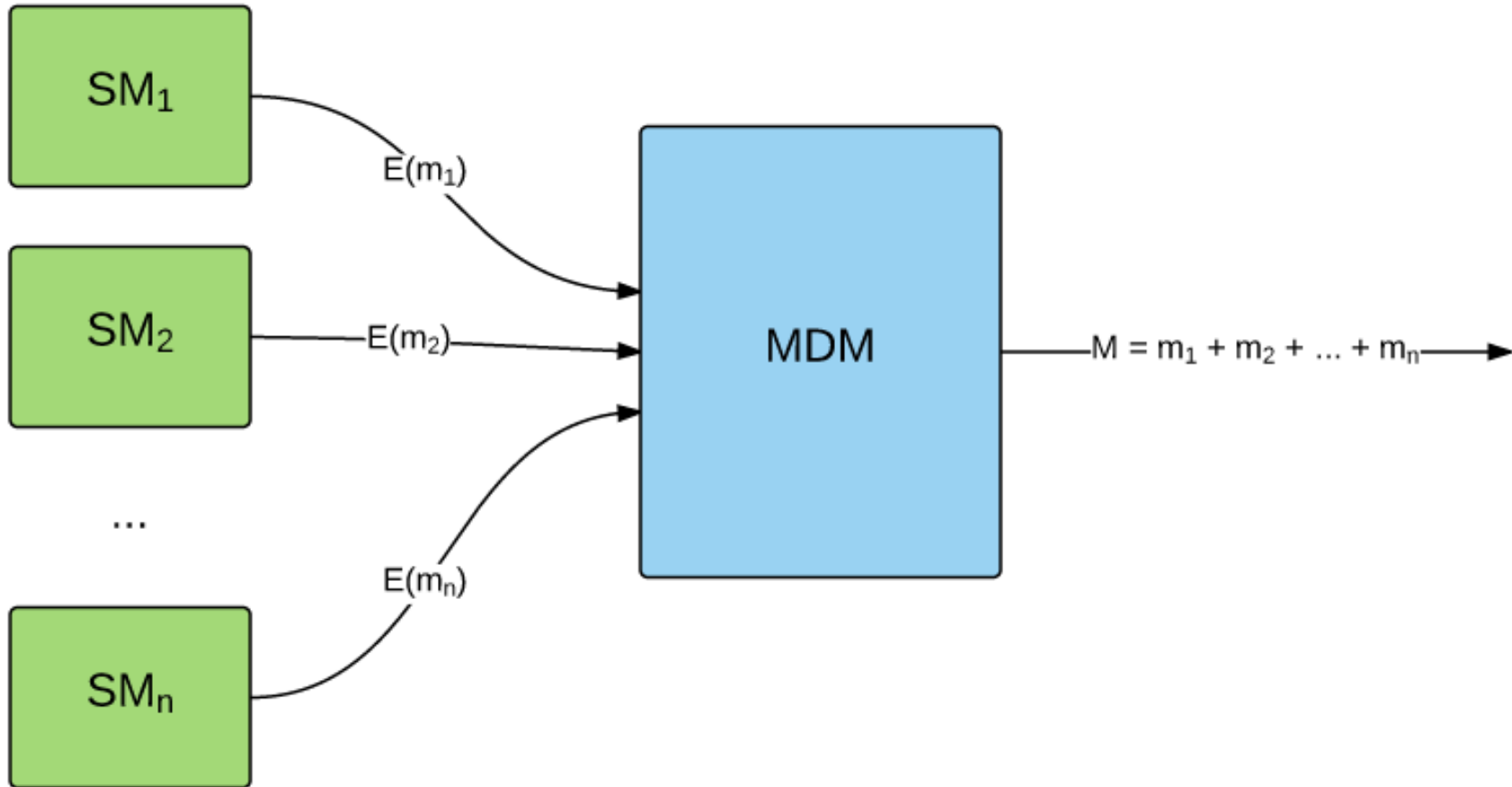


# Criptografia Homomorfica em Smart Meters

Author  
2015

# Problema



- Como calcular  $M$ ?

# Criptografia de Palier

- Seja  $L(u) = (u - 1) / n$
- Selecione  $p$  e  $q$  números primos grandes
- $n = p \cdot q$
- $\lambda = mmc( (p - 1) \cdot (q - 1) )$
- Selecione  $g$  um inteiro de  $1$  a  $n^2$  de forma que
- $b = L(g^\lambda \pmod{n^2})$  seja coprimo a  $n$ , ou seja,  $mdc(b, n) = 1$
- Por fim,  $\mu$  é o inverso multiplicativo de  $b$  módulo  $n$ , ou seja,  $\mu = b^{-1} \pmod{n}$
- $P_k = (n, g)$  e  $S_k = (n, \lambda, \mu)$

# Criptografia de Palier

- Criptografia (de  $m$ ):

- $r = \text{rand}[1, n-1]$

- $c = g^m \cdot r^n \pmod{n^2}$

- Descriptografia (de  $c$ ):

- $m = (L(c^\lambda \pmod{n^2}) \cdot \mu) \pmod{n}$

- Homomorfismo:

- Considerando duas mensagens, temos  $E(m_1 + m_2) = E(m_1) \cdot E(m_2)$ . Logo, o sistema de Paillier apresenta homomorfismo aditivo ao multiplicar blocos cifrados.

# Criptografia de Palier

- Exemplo:

- $p = 7, q=11$

- $n = p \cdot q = 7 \cdot 11 \Rightarrow n = 77$

- $\lambda = mmc((p-1), (q-1)) = mmc(6, 10) \Rightarrow \lambda=30$

- $g = rand[1, n^2], \text{ tal que , } mdc(b, n) = 1, \text{ onde } b = L(g^\lambda \pmod{n^2}) \Rightarrow g = 23 \text{ e } b = 40$

- $\mu = b^{-1} \pmod{n} \Rightarrow \mu \cdot 40 \pmod{77} = 1 \Rightarrow \mu = 52$

- $P_k = (n, g) \Rightarrow P_k = (77, 23)$

- $S_k = (n, \lambda, \mu) \Rightarrow S_k = (77, 30, 52)$

# Criptografia de Palier

- Criptografia:

- $m=14$

- $r = \text{rand}[1, n-1] \Rightarrow r=69$

- $c \equiv g^m \cdot r^n \pmod{n^2} \Rightarrow 2314 \cdot 6977 \pmod{772} \Rightarrow c = 3265$

- Descriptografia:

- $c = 3265$

- $m \equiv (L(c^\lambda \pmod{n^2}) \cdot \mu) \pmod{n} \Rightarrow m = (L(1618) \cdot 52) \pmod{77} \Rightarrow m=14$

# Criptografia de Palier

- Homomorfismo:

- Criptografia:

- $m_1 = 14, m_2 = 3, m_3 = m_1 + m_2 = 17$

- $c_1 \equiv g^{m_1} \cdot r^n \pmod{n^2} \Rightarrow c_1 \equiv 2314 \cdot 6977 \pmod{77^2} \Rightarrow c_1 = 3265$

- $c_2 \equiv g^{m_2} \cdot r^n \pmod{n^2} \Rightarrow c_2 \equiv 233 \cdot 2677 \pmod{77^2} \Rightarrow c_2 = 3503$

- $c_3 = c_1 \cdot c_2 \Rightarrow c_3 = 11437295$

- Descriptografia:

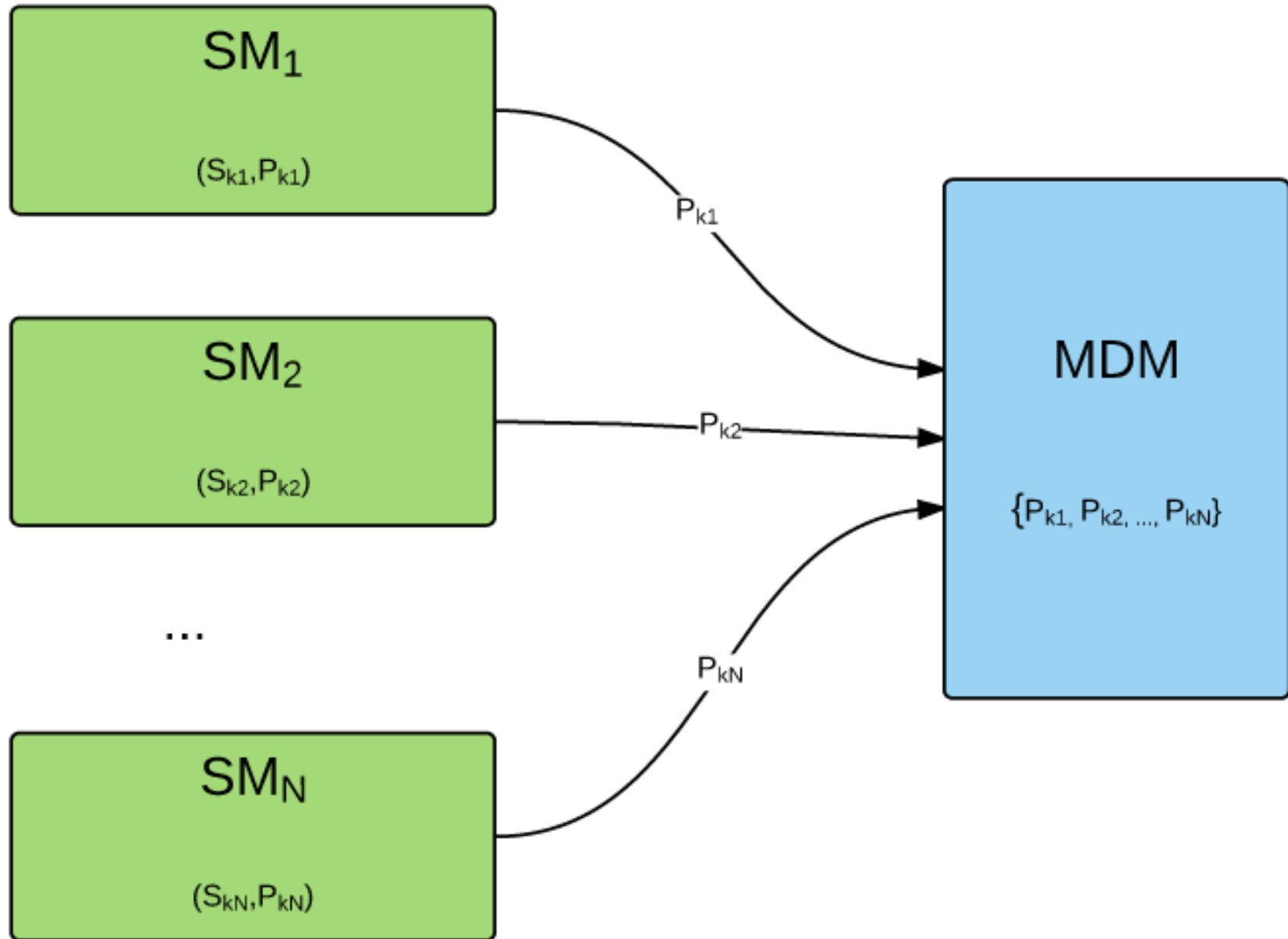
- $m_3 \equiv (L(c_3^\lambda \pmod{n^2}) \cdot \mu) \pmod{n} \Rightarrow m_3 \equiv (L(4929) \cdot 52) \pmod{77} \Rightarrow m_3 = 17$

# Privacidade em Smart Metering por Garcia et al.

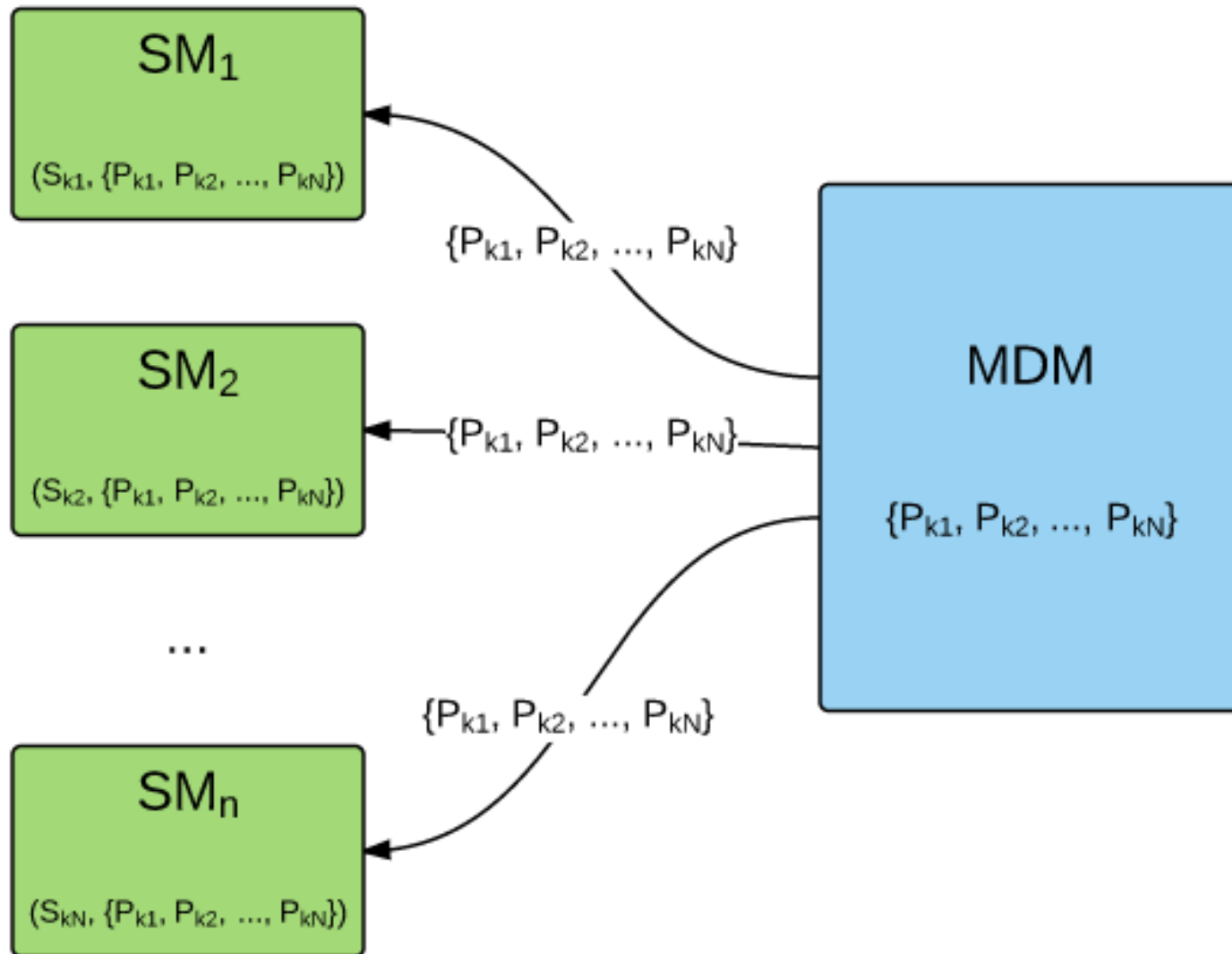
- 1- Cada medidor possui uma chave pública  $P_{ki}$  e uma privada  $S_{ki}$  de Pailier
- 2- A concessionária obtém as chaves públicas de todos os medidores e compartilha este grupo de chaves públicas com todos os medidores.
  - Cada medidor fica com a sua chave privada  $S_{ki}$ , e todas as chaves públicas  $\{P_{k1}, \dots, P_{kN}\}$ .



# Privacidade em Smart Metering por Garcia et al.



# Privacidade em Smart Metering por Garcia et al.



# Privacidade em Smart Metering

## por Garcia et al.

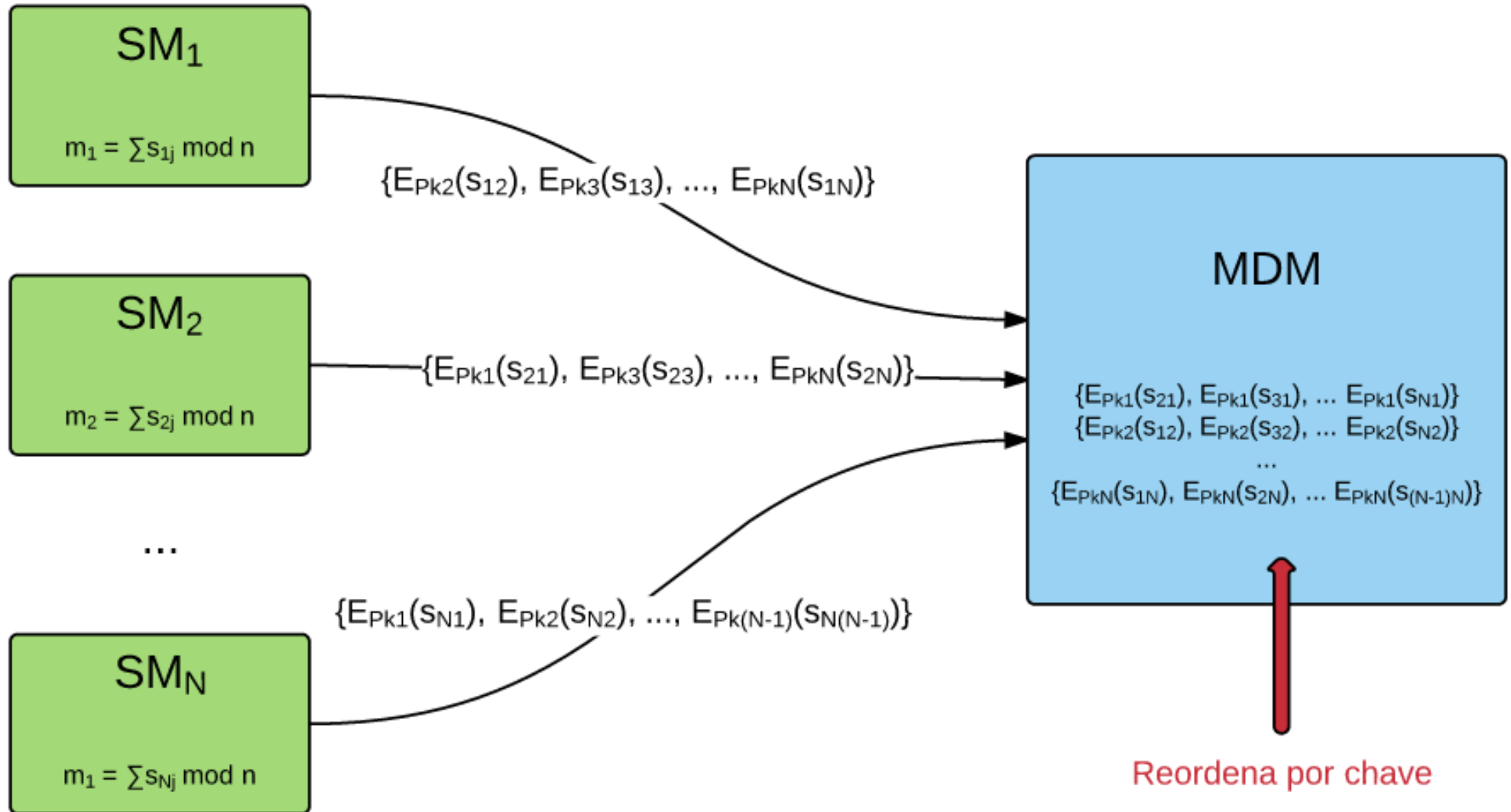
**3-** Cada medidor  $\mathbf{SM}_i$  calcula para a sua medição  $\mathbf{m}_i$ ,  $N$  segredos compartilhados, de forma que

$$m_i = \sum_j s_{ij} \quad \text{para um } n \text{ grande.}$$

**4-**  $\mathbf{SM}_i$  armazena  $s_{ij}$  para si mesmo e envia para a concessionária os outros segredos compartilhados criptografados com as chaves públicas dos outros  $N-1$  medidores

– Ou seja, envia  $E_{PKj}(s_{ij})$  para  $j = 1, \dots, i-1, i+1, \dots, N$ .

# Privacidade em Smart Metering por Garcia et al.



# Privacidade em Smart Metering por Garcia et al.

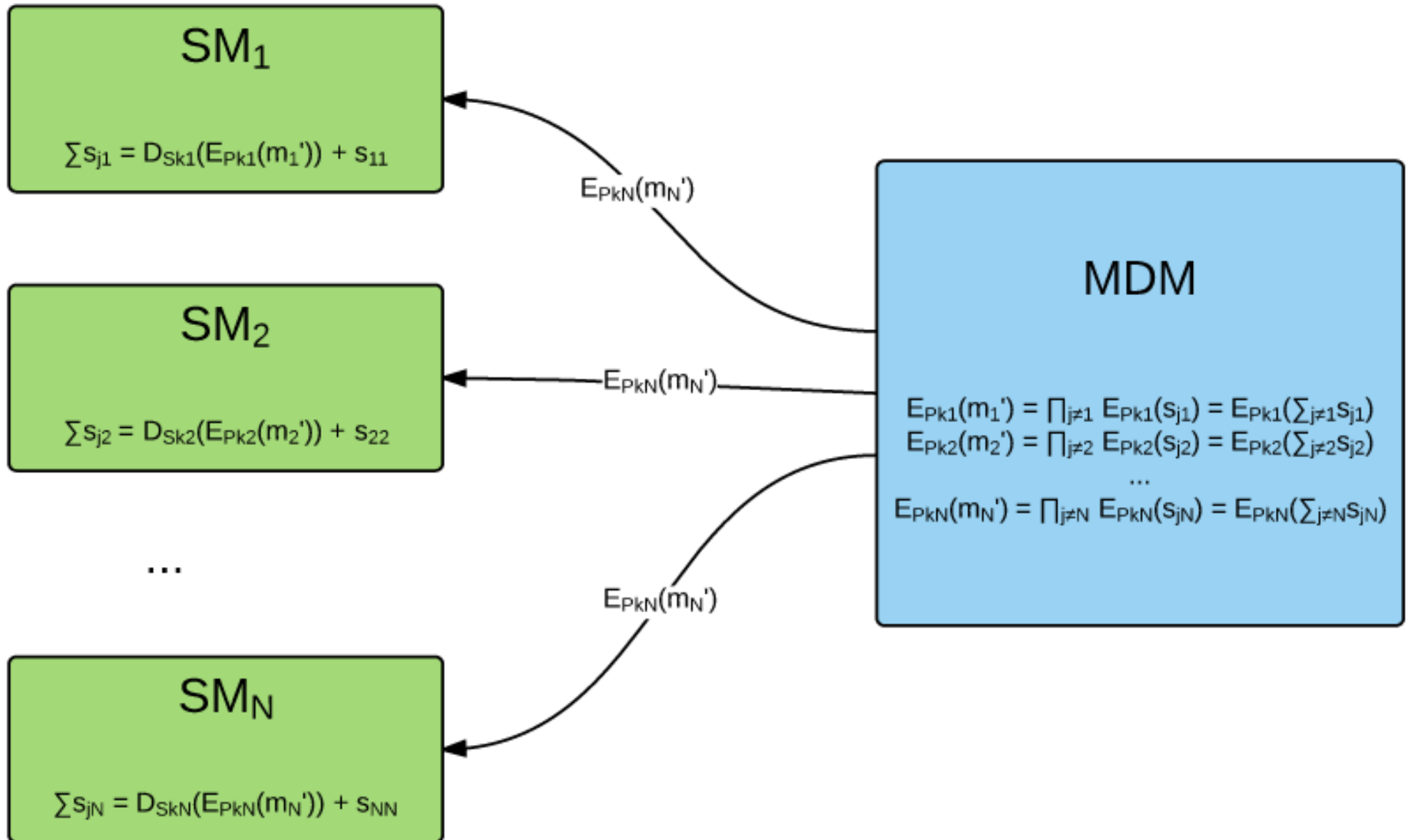
**5-** Ao receber todos os segredos compartilhados criptografados, a concessionária multiplica os que foram criptografados com a mesma chave pública. Devido a propriedade homomórfica, tem-se para cada  $i$ :

$$E_{P_{ki}}(m_i') = \prod_{j \neq i} E_{P_{ki}}(s_{ji}) = E_{P_{ki}}\left(\sum_{j \neq i} s_{ji}\right)$$

**6-** A concessionária envia  $E_{P_{ki}}(m_i')$  para o medidor  $SM_i$ , que pode descriptografar com a sua chave privada  $S_{ki}$  e adicionar o segredo  $s_{ji}$ . Assim, o medidor obtém:

$$\sum_j s_{ji}$$

# Privacidade em Smart Metering por Garcia et al.

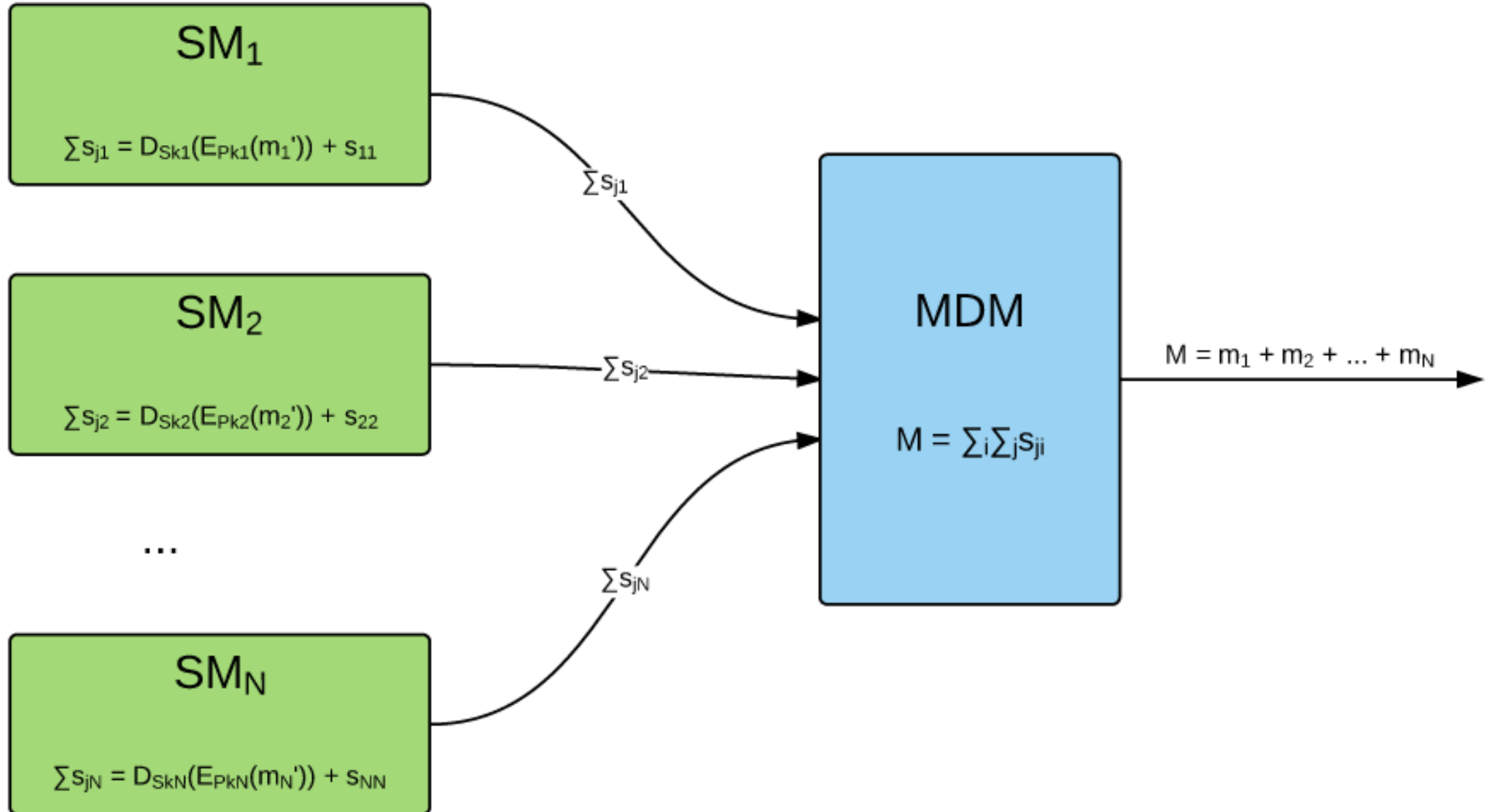


# Privacidade em Smart Metering por Garcia et al.

**7-** O medidor  **$SM_i$**  então envia  $\sum_j s_{ji}$  para a concessionária, que ao somar os resultados recebidos de todos os medidores obterá o consumo total  **$M$**  da região

$$M = \sum_i \sum_j s_{ji} \mod n$$

# Privacidade em Smart Metering por Garcia et al.





# Privacidade em Smart Metering por Garcia et al.

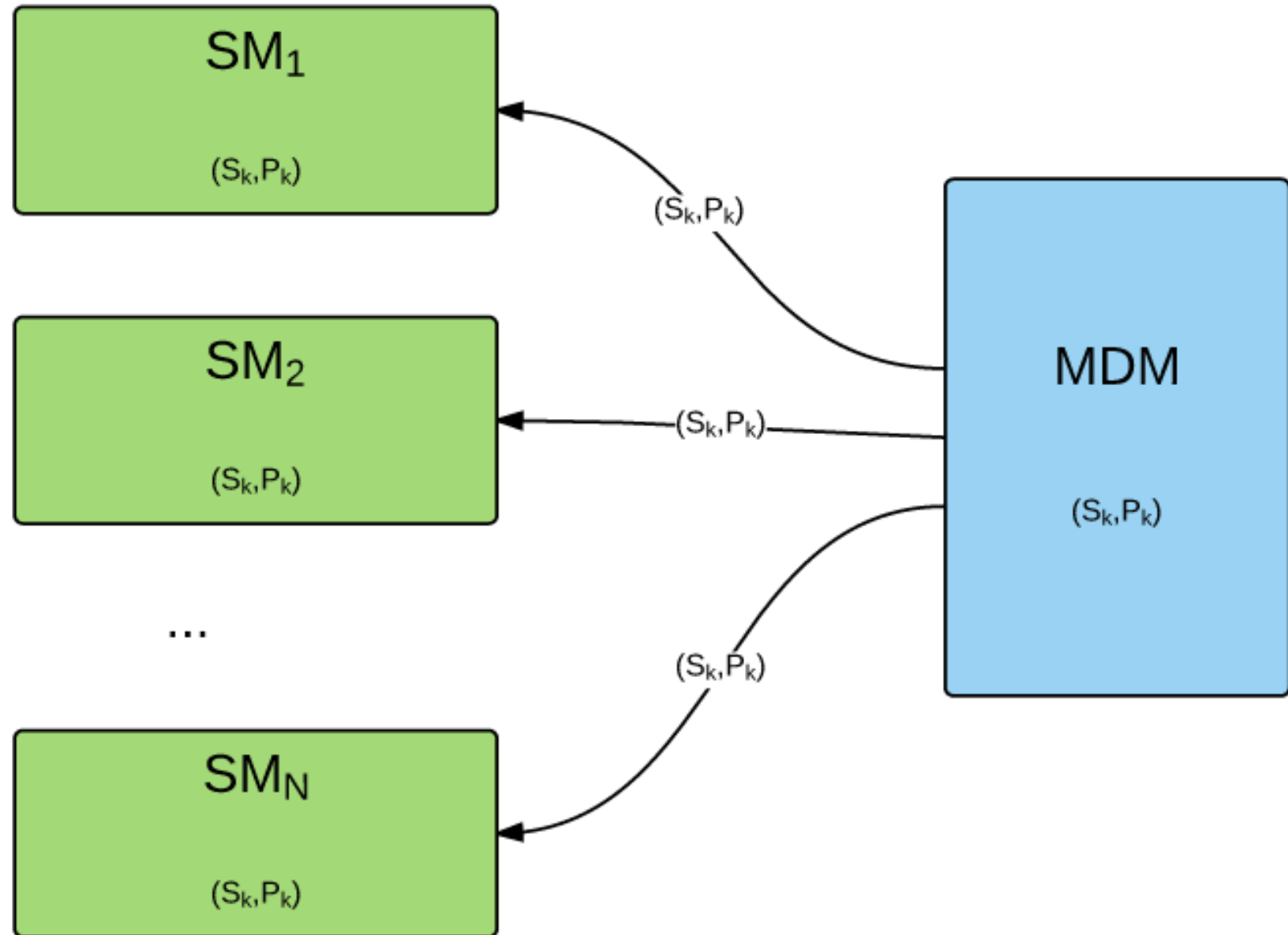
- Problemas:

- Muita troca de mensagens entre um medidor e a concessionária (5 mensagens)
- Não é escalável:
  - Se um único medidor falhar durante dos passos, não se obtém nem mesmo uma aproximação
- Considerando geração de chaves, criptografia, descriptografia e geração de segredos como  $O(1)$ , temos:
  - $O(N)$  para cada medidor (passos 3 e 4)
  - $O(N^2)$  para a concessionária (passo 5)

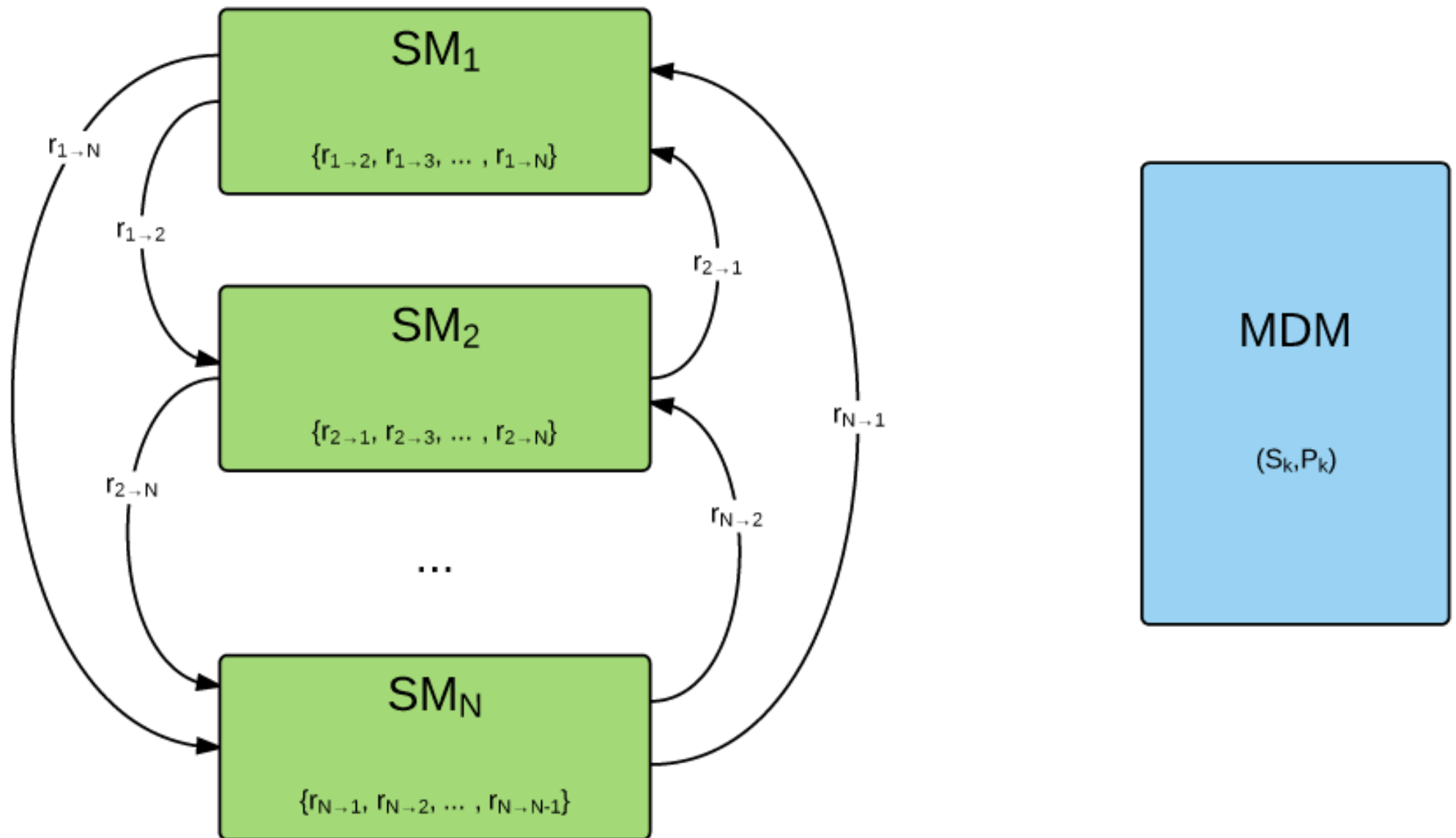
# Privacidade em Smart Metering por Erkin et al.

- 1-** Um único par de chaves do esquema de Paillier é compartilhado entre todos os  $N$  medidores (i.e, todos os medidores possuem  $P_k$  e  $S_k$ )
- 2-** Para uma medição  $m_i$ , o medidor  $SM_i$  gera  $N - 1$  números aleatórios, um para cada outro medidor, e os distribui utilizando canais seguros (ex, criptografia assimétrica RSA na comunicação entre medidores).

# Privacidade em Smart Metering por Erkin et al.



# Privacidade em Smart Metering por Erkin et al.



# Privacidade em Smart Metering por Erkin et al.

**3-** Após receber todos os números aleatórios gerados pelos outros medidores,  $SM_i$  computa:

$$R_i = n + \sum_{j=1, i \neq j}^N r_{(i \rightarrow j)} - \sum_{j=1, i \neq j}^N r_{(j \rightarrow i)}$$

– Onde  $n$  é o módulo usado em Paillier e  $r_{i \rightarrow j}$  é o número aleatório gerado de  $SM_i$  para  $SM_j$

**4-** Em seguida computa-se um hash  $h_t$  utilizando o timestamp da medição atual ( $m_i$ ). Este hash precisa ser coprimo a  $n$ , ou seja,  $mdc(h_t, n) = 1$ .

– Como o timestamp é sincronizado para todos os medidores, espera-se que o hash obtido seja o mesmo para todos.

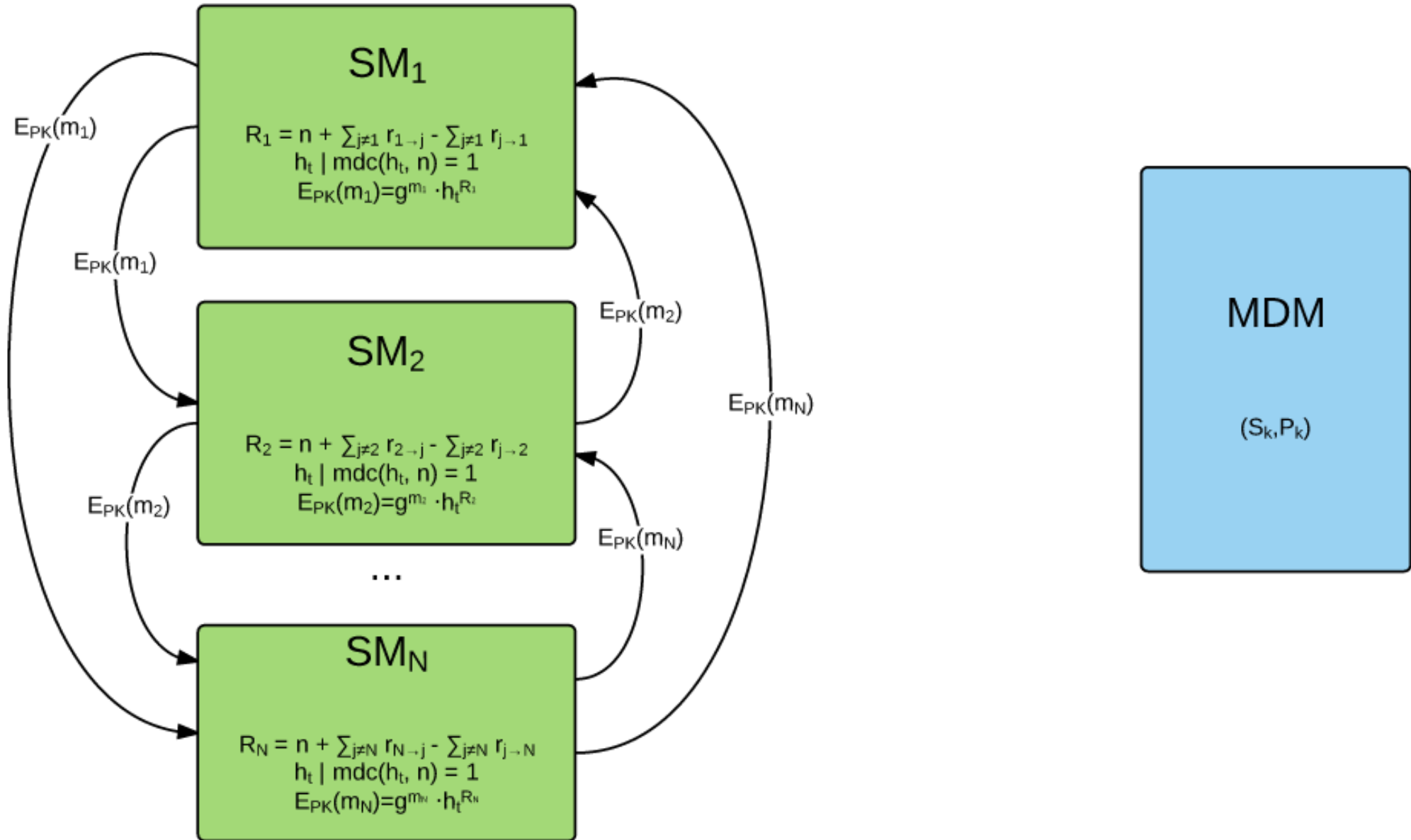
# Privacidade em Smart Metering por Erkin et al.

**5-** Após o medidor computar  $R_i$  e  $h_t$ , criptografa-se a medição  $m_i$  utilizando o seguinte esquema modificado de Paillier:

$$E_{P_k}(m_i) = g^{m_i} \cdot h_t^{R_i}$$

- Em seguida esta mensagem criptografada é disseminada para todos os outros  **$N-1$**  medidores.

# Privacidade em Smart Metering por Erkin et al.



# Privacidade em Smart Metering por Erkin et al.

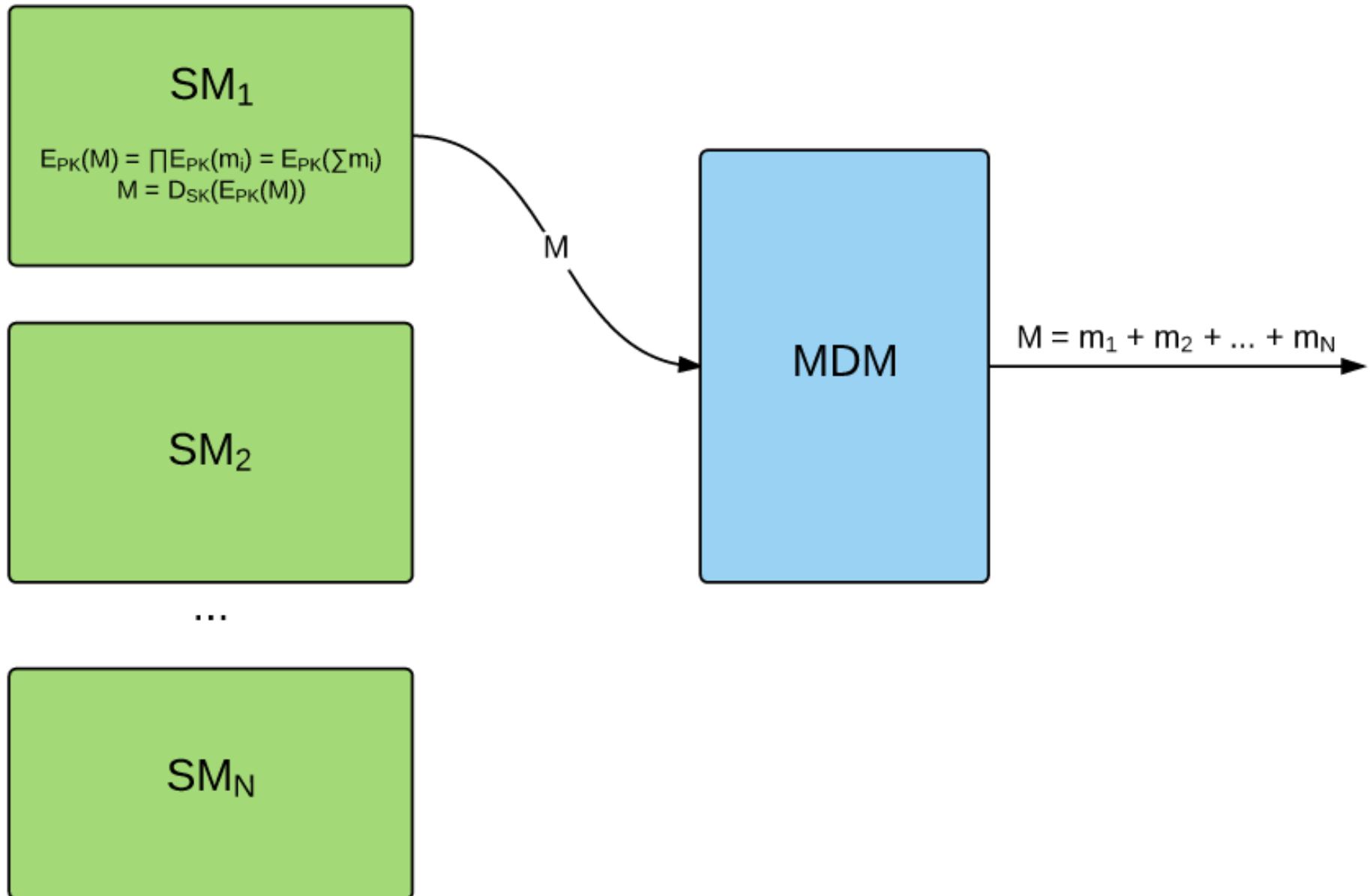
**6-** Quando um medidor receber todas as medições criptografadas dos outros medidores, ele poderá calcular o consumo total na região devido a propriedade homomórfica:

$$E_{P_k}(M) = \prod_{i=1}^N E_{P_k}(m_i) = E_{P_k}\left(\sum_{i=1}^N m_i\right)$$

- Ao descriptografar  $E_{P_k}(M)$ , o medidor poderá enviar o valor  $M$  para a concessionária.



# Privacidade em Smart Metering por Erkin et al.



# Privacidade em Smart Metering por Erkin et al.

- Por que funciona?
  - Um medidor não consegue descriptografar medições individuais dos outros
    - Ele só possui os números aleatórios gerados por ele e os gerados para ele

$$E_{P_k}(m_i) = g^{m_i} \cdot h_t^{R_i} \quad R_i = n + \sum_{j=1, i \neq j}^N r_{(i \rightarrow j)} - \sum_{j=1, i \neq j}^N r_{(j \rightarrow i)}$$

- A descriptografia do agregado funciona, pois:

$$E_{P_k}(M) = g^{m_1 + m_2 + \dots + m_N} \cdot h_t^{\sum_{i=1}^N n + \sum_{i=1}^N \sum_{j=1, i \neq j}^N r_{(i \rightarrow j)} - \sum_{i=1}^N \sum_{j=1, i \neq j}^N r_{(j \rightarrow i)}}$$

# Privacidade em Smart Metering por Erkin et al.

$$E_{P_k}(M) = g^M \cdot h_t^{N \cdot n}$$

- Ora, sendo  $r = h_t^N$  como um número aleatório, temos a configuração de Paillier original

# Privacidade em Smart Metering

## por Erkin et al.

- Problemas:

- Muita troca de mensagens entre medidores ( $2N - 2$  mensagens) e duas com a concessionária
- Não é escalável:
  - Se um único medidor falhar durante dos passos, não se obtém nem mesmo uma aproximação
- Considerando geração de chaves, criptografia, descriptografia e geração de segredos como  $O(1)$ , temos:
  - $O(N)$  para cada medidor (passos 2, 3 e 6)
  - $O(N)$  para a concessionária (passo 1)
    - Mas poderia ser  $O(1)$  (cada medidor salvaria o par de chaves)

# Criptografia de ElGamal

- Problema do logaritmo discreto:
  - Seja  $\mathbf{G}$  um grupo multiplicativo de ordem  $\mathbf{p}$  e seja  $\mathbf{g}$  um gerador de  $\mathbf{G}$
  - Dado  $\mathbf{g}$  e  $\mathbf{y}$ , como encontrar o inteiro  $\mathbf{x}$  tal que  $\mathbf{g}^{\mathbf{x}} = \mathbf{y}$ ?
  - Tal inteiro  $\mathbf{x}$  é o logaritmo discreto de  $\mathbf{y}$  na base  $\mathbf{g}$  ( $\log_{\mathbf{g}} \mathbf{y} = \mathbf{x}$ )
- Problema computacional de Diffie-Hellman:
  - Seja  $\mathbf{G}$  um grupo multiplicativo.
  - Dado  $\mathbf{g}^{\mathbf{a}}$  e  $\mathbf{g}^{\mathbf{b}}$  com  $\mathbf{a}$  e  $\mathbf{b}$  desconhecidos, como computar  $\mathbf{g}^{\mathbf{ab}}$ ?

# Criptografia de ElGamal

- Seja  $q$  um número primo grande
- Seja  $g$  um gerador do grupo multiplicativo  $G$  de ordem  $q$  (i.e,  $\mathbb{Z}_q^*$ )
- Obter  $x = rand(G)$  e  $y = g^x$
- $S_k = x$  e  $P_k = (G, g, y)$
- Para criptografar uma mensagem  $m \in G$ , obtém-se  $r = rand(G)$  e computa-se  $c = g^r$  e  $d = m.y^r$ . Assim,  $E(m) = (c, d)$
- Para descriptografar  $E(m)$  computa-se  $m = d.c^{-x}$

# Criptografia de ElGamal

- Propriedade homomórfica:

- Seja  $E(m_1) = (c_1, d_1) = (g^{r_1}, m_1 \cdot y^{r_1})$  e  $E(m_2) = (c_2, d_2) = (g^{r_2}, m_2 \cdot y^{r_2})$

- Assim,  $E(m_1) \cdot E(m_2) = (c_1 \cdot c_2, d_1 \cdot d_2) = (g^{r_1 + r_2}, m_1 \cdot m_2 \cdot y^{r_1 + r_2}) = E(m_1 \cdot m_2)$

- Ou seja, ElGamal é um sistema homomórfico multiplicativo

# Criptografia de ElGamal

- Dado  $E(g^{m1})$  e  $E(g^{m2})$ , então  $E(g^{m1}) \cdot E(g^{m2}) = E(g^{m1} \cdot g^{m2}) = E(g^{m1 + m2})$
- Assim, também pode-se obter a propriedade homomórfica aditiva se conseguir computar  $m$  em  $m' = g^m$ .
  - Por exemplo, na Espanha o consumo de um consumidor durante 30 min fica em torno de 0 – 7500 Wh. Para 128 medidores é necessário números de 20-bits. Em um Intel Core 2.5GHz e 4GB RAM, levou-se em média 0.046s.
  - Assim, sabendo-se o range, o cálculo do logaritmo discreto não fica tão custoso.



# Privacidade em Smart Metering por Busom et al.

## 1- Cada medidor possui:

- O número primo grande  $q$  e o gerador  $g$  (já vem em hardware e de fábrica)
- Uma chave privada  $x_i$
- Uma chave pública  $y_i = g^{x_i}$  e um certificado  **$Cert_i$**
- A chave pública da autoridade certificadora que verifica os certificados dos outros medidores.

# Privacidade em Smart Metering por Busom et al.

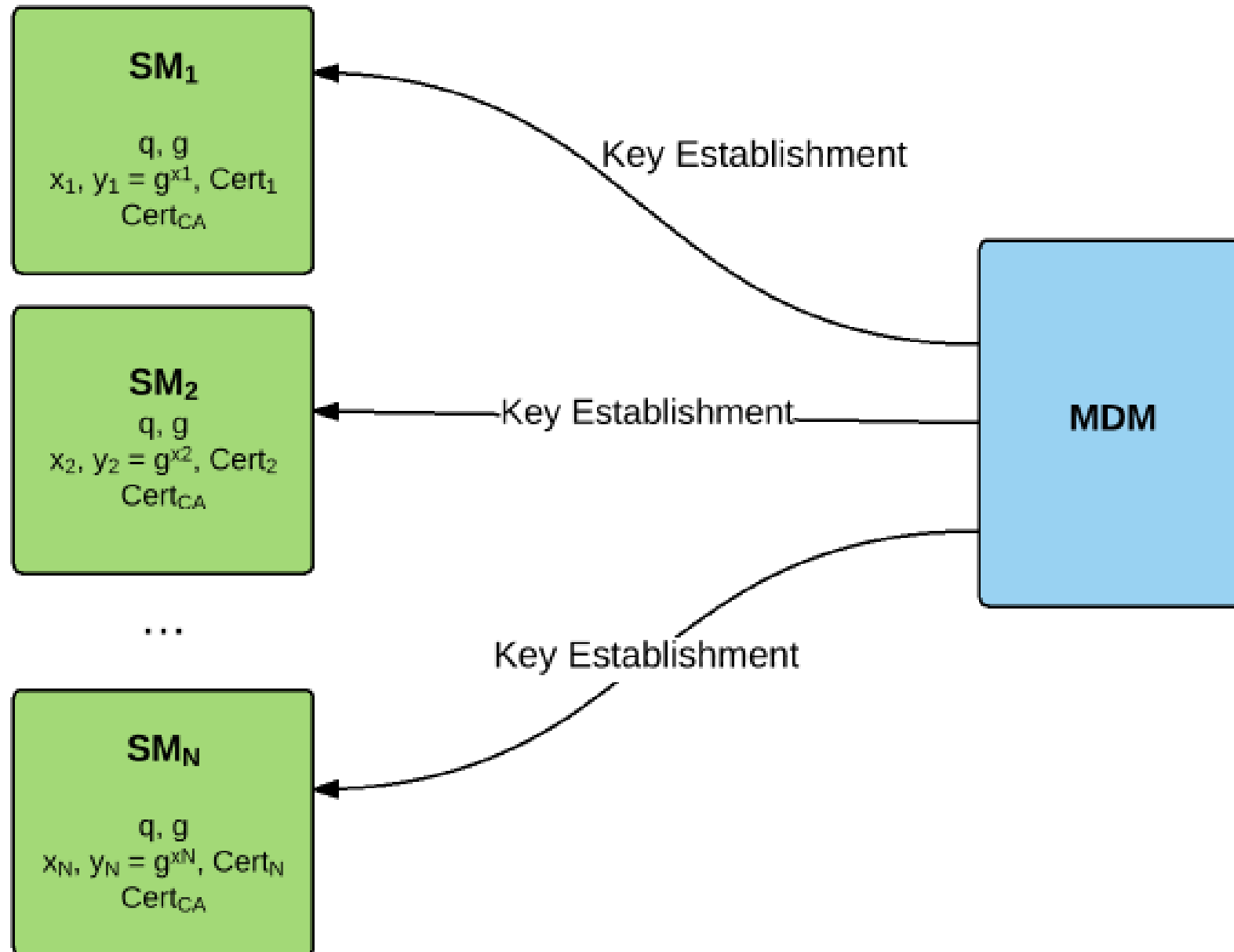
**2-** Quando se tem uma nova configuração (ex, novo medidor na região), a concessionária inicia um procedimento de estabelecimento de chaves:

- A concessionária envia uma mensagem de key establishment para cada um dos medidores
- Cada medidor envia  $y_i$  e **Cert<sub>i</sub>** para a concessionária
- A concessionária verifica a validade de cada **Cert<sub>i</sub>** e envia  $\{y_1, \dots, y_n\}$  e  $\{\mathbf{Cert}_1, \dots, \mathbf{Cert}_n\}$  para cada medidor
- Finalmente, cada um dos medidores verifica a validade de cada **Cert<sub>i</sub>** e computa uma chave pública global:

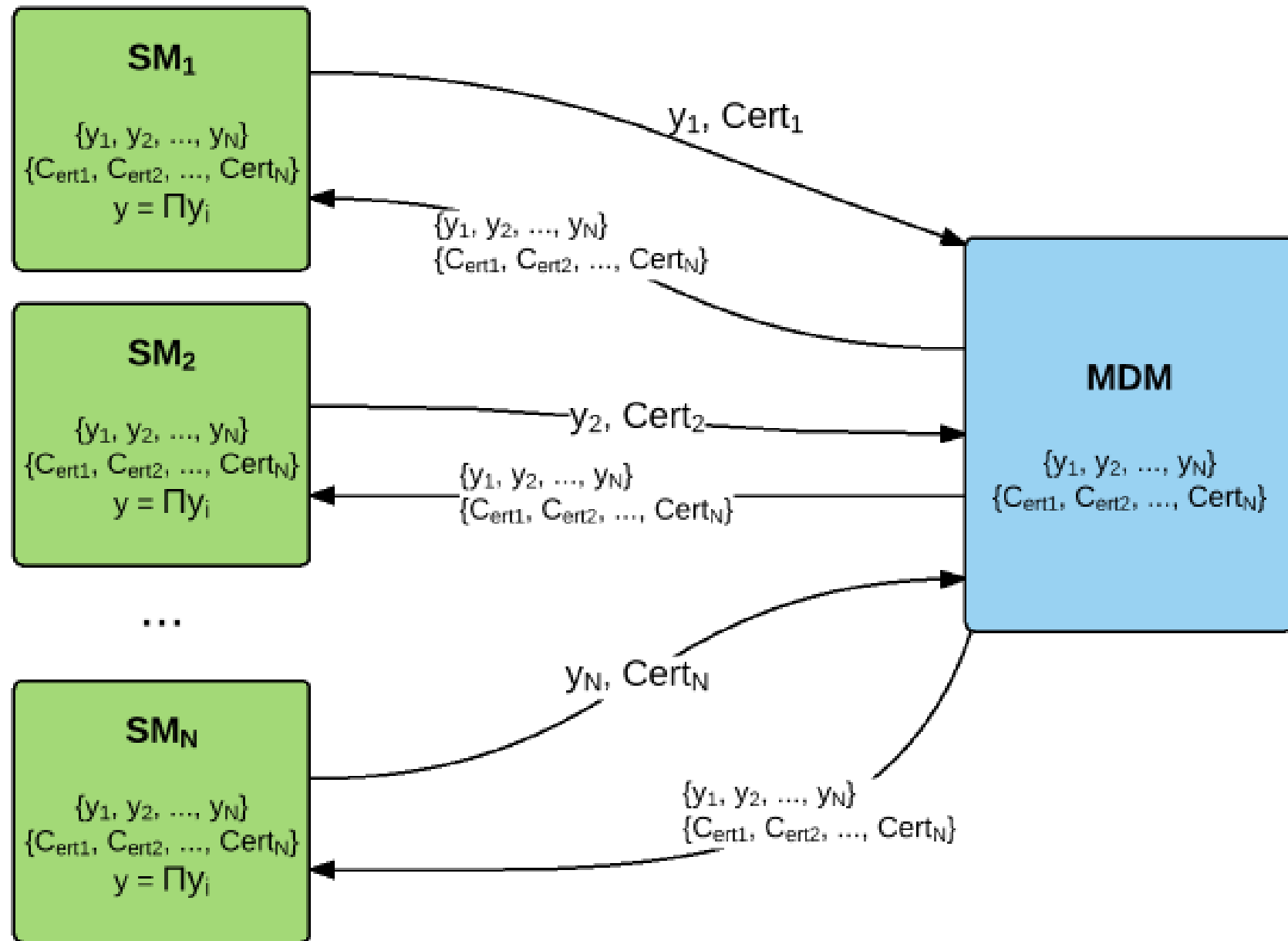
$$y = \prod_{i=1}^N y_i$$

# Privacidade em Smart Metering

por Busom et al.



# Privacidade em Smart Metering por Busom et al.



# Privacidade em Smart Metering por Busom et al.

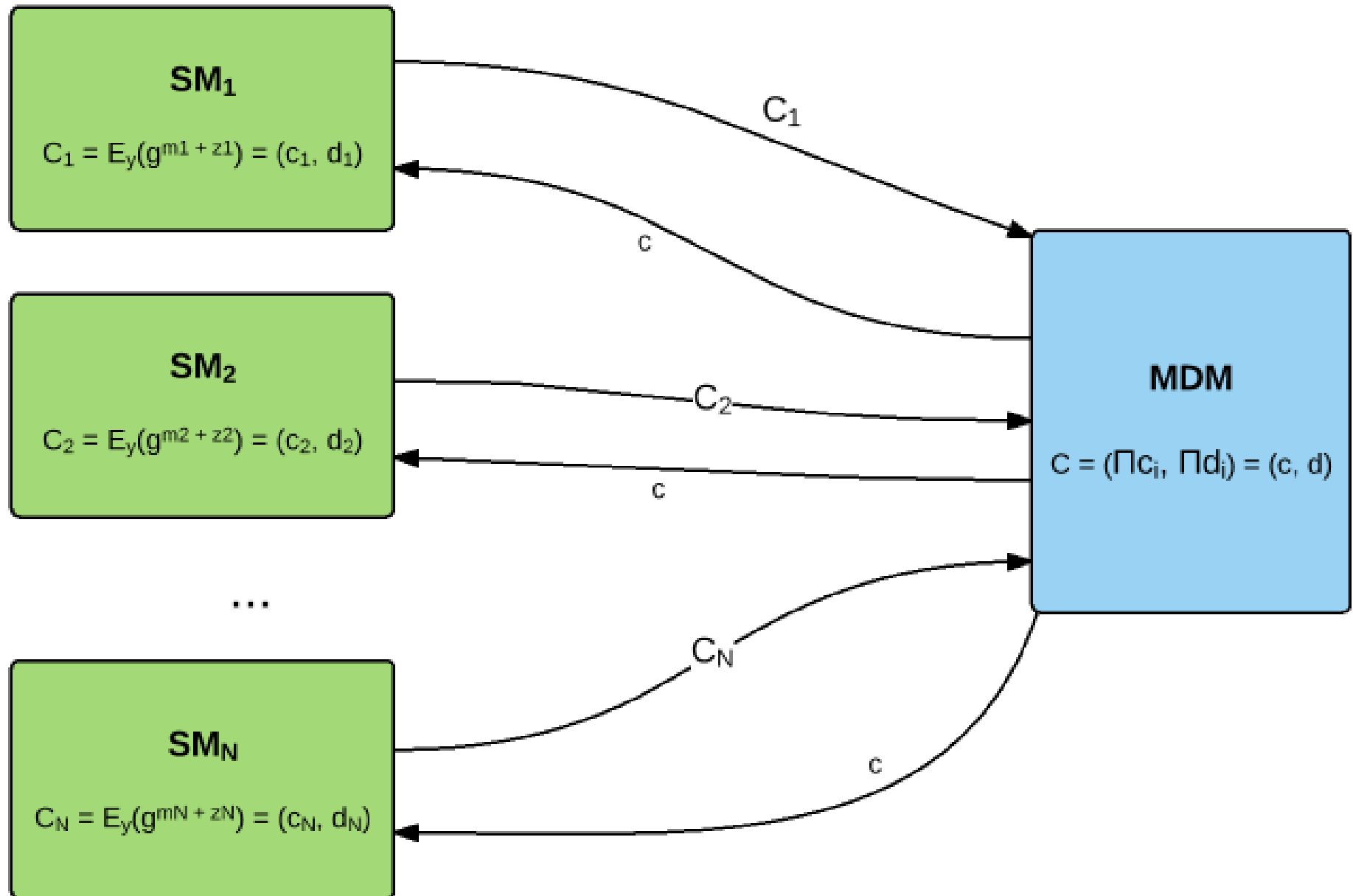
**3-** A cada instante, a concessionária envia para os medidores requisições de medição e cada medidor gera um número aleatório  $z_i \in \mathbb{Z}_q^*$ , computa  $\mathbf{C}_i = E_y(g^{mi} + z_i) = (\mathbf{c}_i, \mathbf{d}_i)$  e envia  $\mathbf{C}_i$  para a concessionária

**4-** A concessionária agrega todos os  $\mathbf{C}_i$  recebidos:

$$C = \left( \prod_{i=1}^N c_i, \prod_{i=1}^N d_i \right) = (c, d)$$

e envia  $\mathbf{c}$  para cada medidor.

# Privacidade em Smart Metering por Busom et al.



# Privacidade em Smart Metering por Busom et al.

**5-** Cada medidor computa  $T_i = \mathbf{c}^{xi} \cdot \mathbf{g}^{zi}$  e envia o resultado para a concessionária

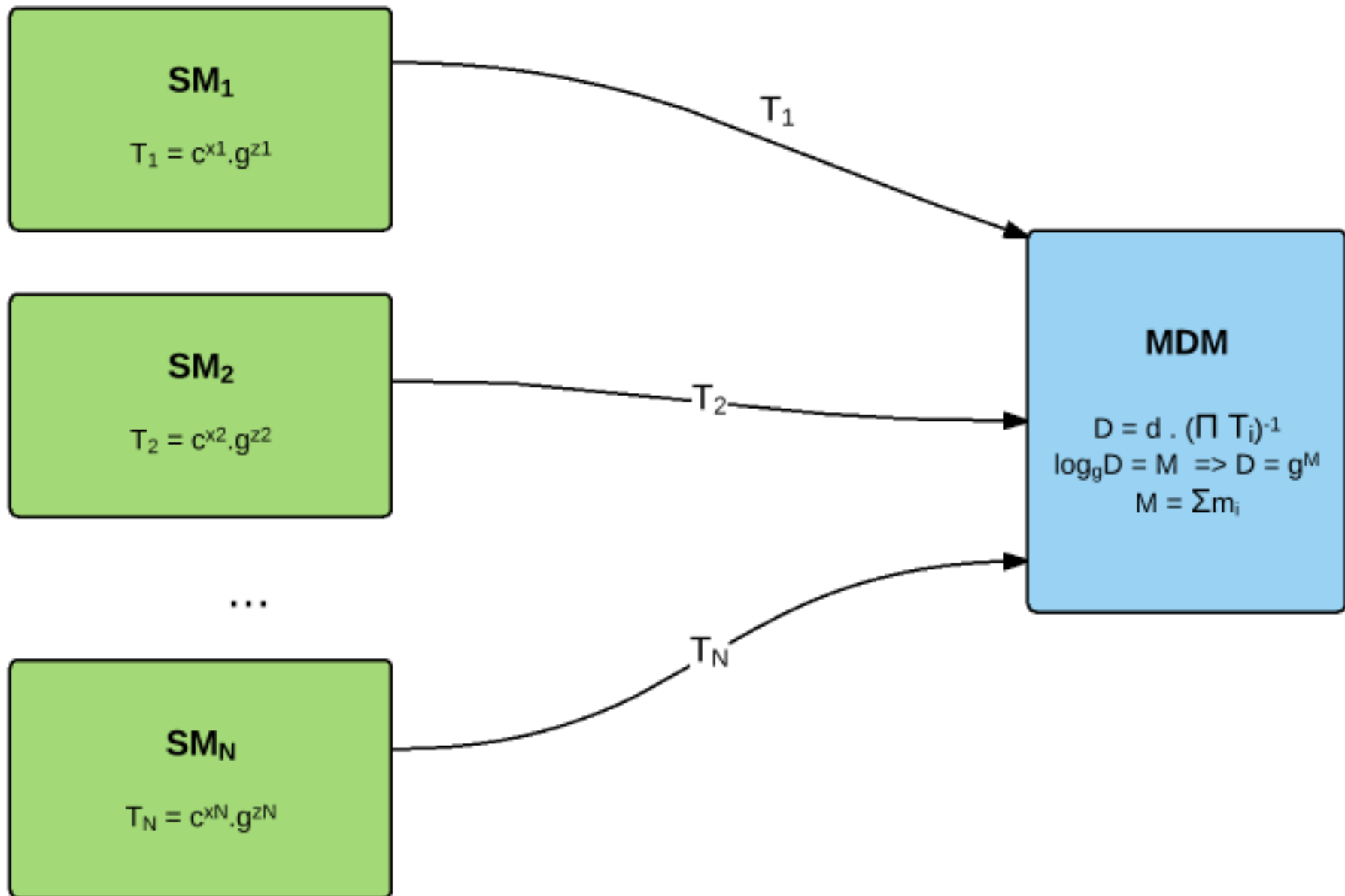
**6-** A concessionária então computa

$$D = d \cdot \left( \prod_{i=1}^N T_i \right)^{-1}$$

e finalmente computa  $\log_g D$ , obtendo

$$D = g^M \qquad M = \sum_{i=1}^N m_i$$

# Privacidade em Smart Metering por Busom et al.





# Privacidade em Smart Metering por Busom et al.

- Por que funciona?

- Primeiro cada medidor calcula

$$C_i = E_y(g^{mi+zi}) = (g^{ri}, g^{mi+zi} \cdot y^{ri})$$

- Em seguida a concessionária calcula

$$C = \left( \prod_{i=1}^N g^{ri}, \prod_{i=1}^N g^{mi+zi} \cdot y^{ri} \right) = (g^r, g^{m+z} \cdot y^r) = (c, d)$$

- Depois cada medidor calcula

$$T_i = c^{xi} \cdot g^{zi} = g^{r \cdot xi} \cdot g^{zi} = g^{xi \cdot r} \cdot g^{zi} = y_i^r \cdot g^{zi}$$

$$D = d \cdot \left( \prod_{i=1}^n T_i \right)^{-1} = \frac{g^{m+z} \cdot y^r}{\prod_{i=1}^n (y_i^r \cdot g^{zi})} = \frac{g^{m+z} \cdot y^r}{(\prod_{i=1}^n y_i^r) \cdot g^z} = \frac{g^{m+z} \cdot y^r}{g^z \cdot y^r} = g^m$$

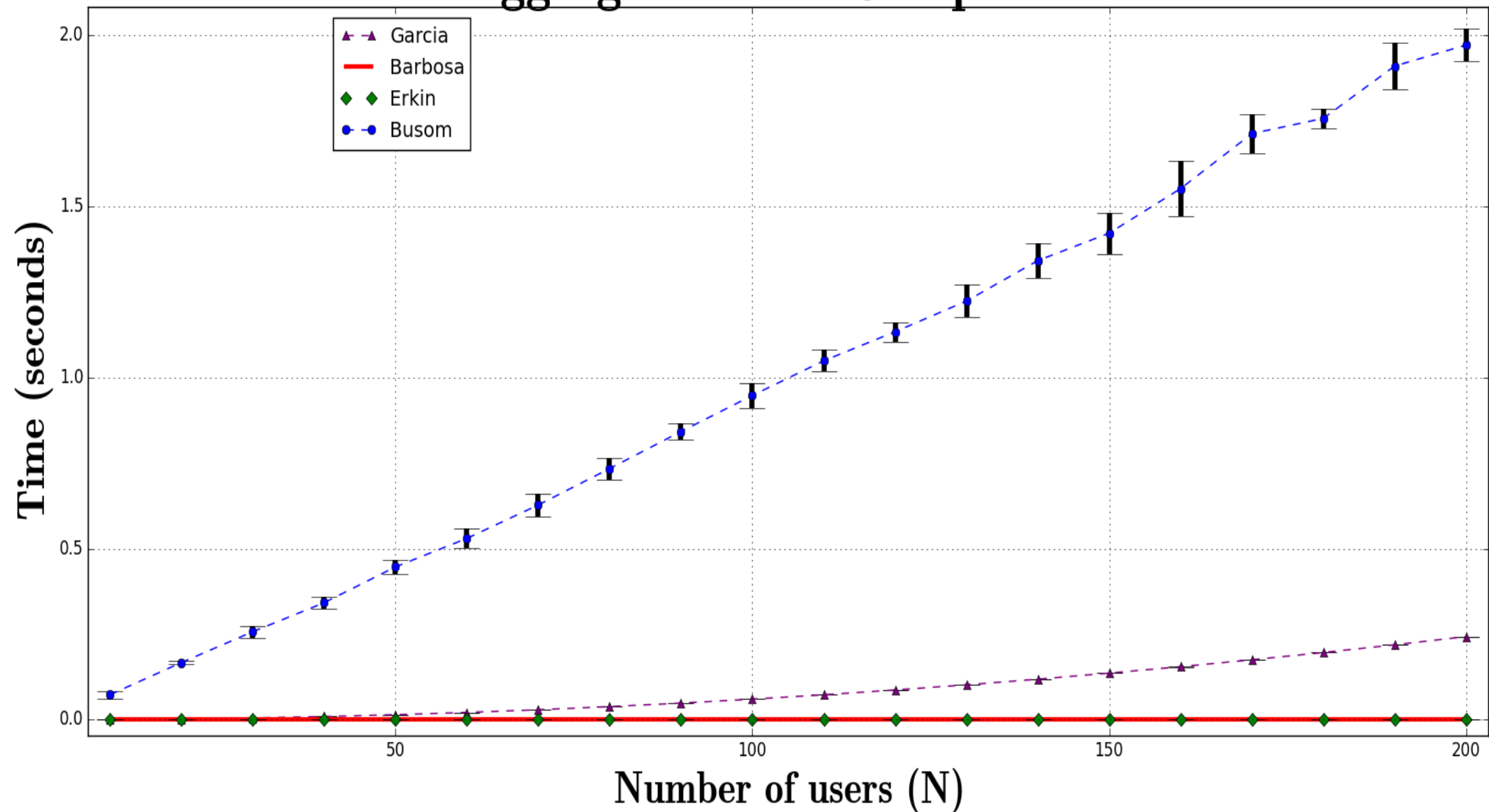
# Privacidade em Smart Metering por Busom et al.

- Problemas:

- Muita troca de mensagens entre um medidor e a concessionária (3 mensagens)
- Não é escalável:
  - Se um único medidor falhar durante dos passos, não se obtém nem mesmo uma aproximação
- Considerando geração de chaves, criptografia, descriptografia e geração de segredos como  $O(1)$ , temos:
  - **$O(1)$**  para cada medidor (mas no key establishment é  **$O(N)$** )
  - **$O(N)$**  para a concessionária (passos 4 e 6)

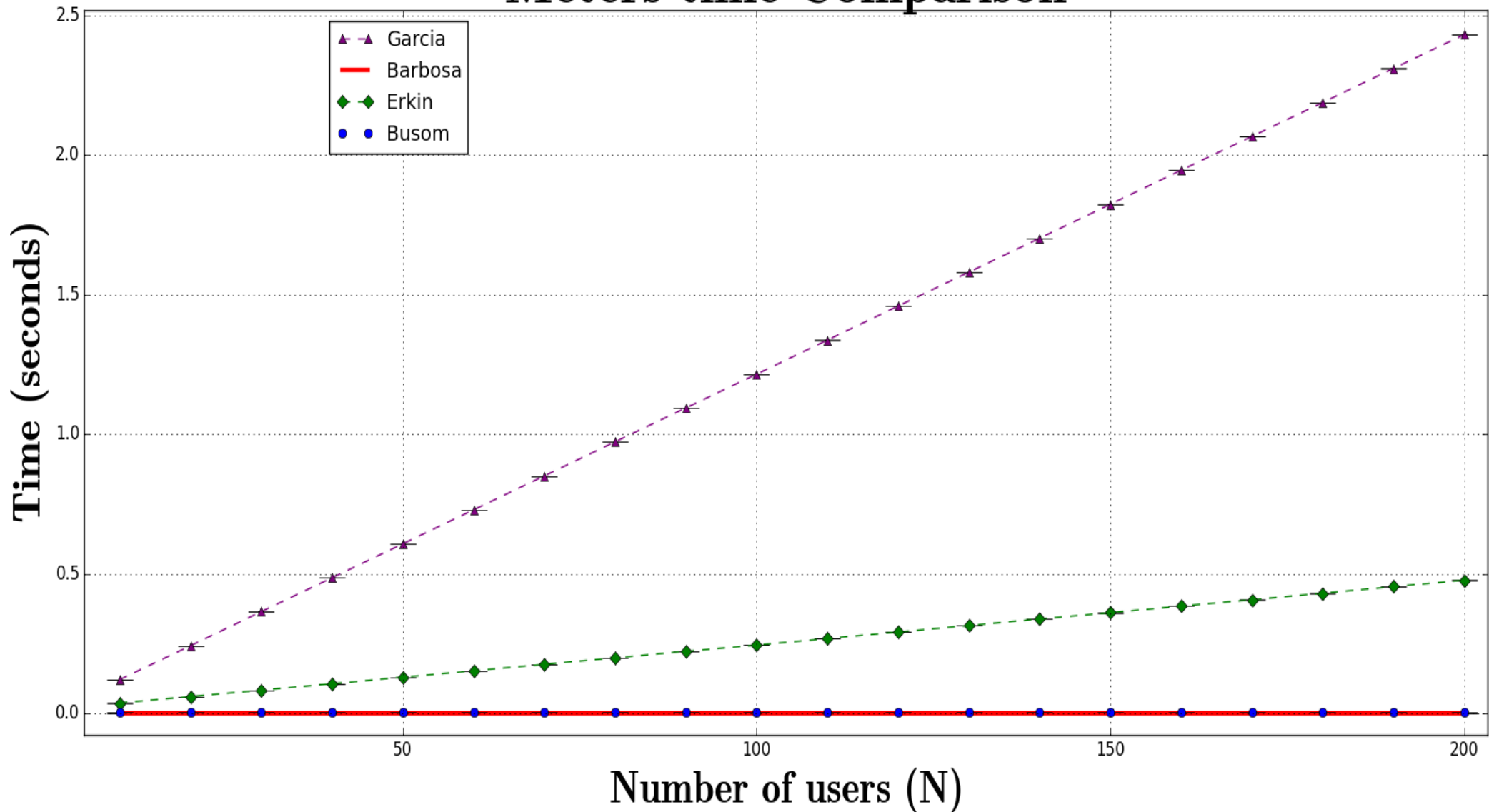
# Privacidade em Smart Metering

## Aggregator time Comparison



# Privacidade em Smart Metering

## Meters time Comparison



# Criptografia de Curva Elíptica

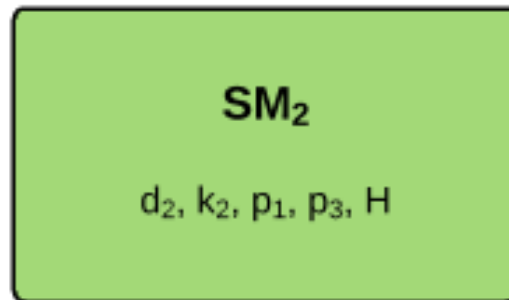
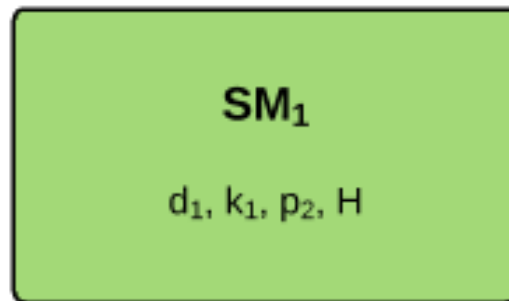
- Criptografia assimétrica
- Mais rápida e usa chaves mais curtas do que os métodos antigos como RSA, proporcionando ao mesmo tempo um nível de segurança equivalente
- Baseada na suposição de que é difícil encontrar um inteiro  $k$ , tal que  $Q = k \cdot P$ , onde  $P$  e  $Q$  são dois pontos em uma curva elíptica
- Existem algumas implementações, como o Suite B da NSA. Entretanto, a maioria delas possuem patentes

# Privacidade em Smart Metering por Borges et al. (iKUP)

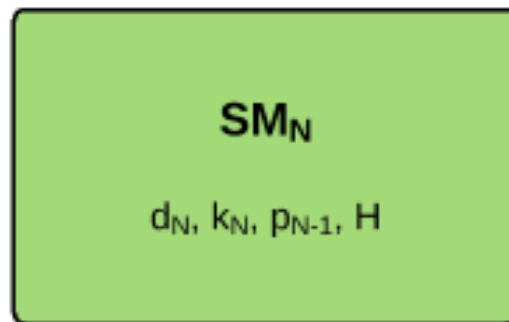
## 1- Cada medidor possui:

- Uma chave simétrica  $d_i$
- Uma chave privada  $k_i$
- As chaves públicas dos vizinhos (ex.,  $p_{i-1}$  e  $p_{i+1}$ )
- Uma função hash  $H$
- Assume-se que a concessionária possui a lista de chaves simétricas  $\{d_1, d_2, \dots, d_n\}$

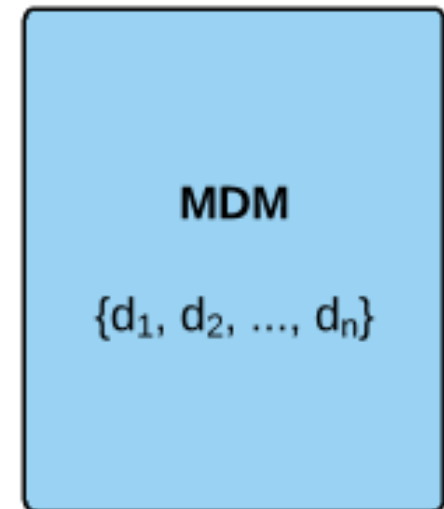
# Privacidade em Smart Metering por Borges et al. (iKUP)



...



Medidores formam uma  
spanning tree



# Privacidade em Smart Metering por Borges et al. (iKUP)

**2-** O medidor  $SM_1$  calcula a sua medição  $m_1$ , e para um instante de tempo  $t$  computa:

$c_1 = m_1 + H(d_1 || t)$ . Em seguida, ele envia  $c_1$  criptografado com  $p_2$  para  $SM_2$

- $SM_2$  então computa  $c_2 = c_1 + m_2 + H(d_2 || t) = m_1 + H(d_1 || t) + m_2 + H(d_2 || t)$ , criptografa com  $p_3$  e envia para  $SM_3$
- Este processo continua até  $SM_N$ , que por fim, envia  $c_n$  para a concessionária



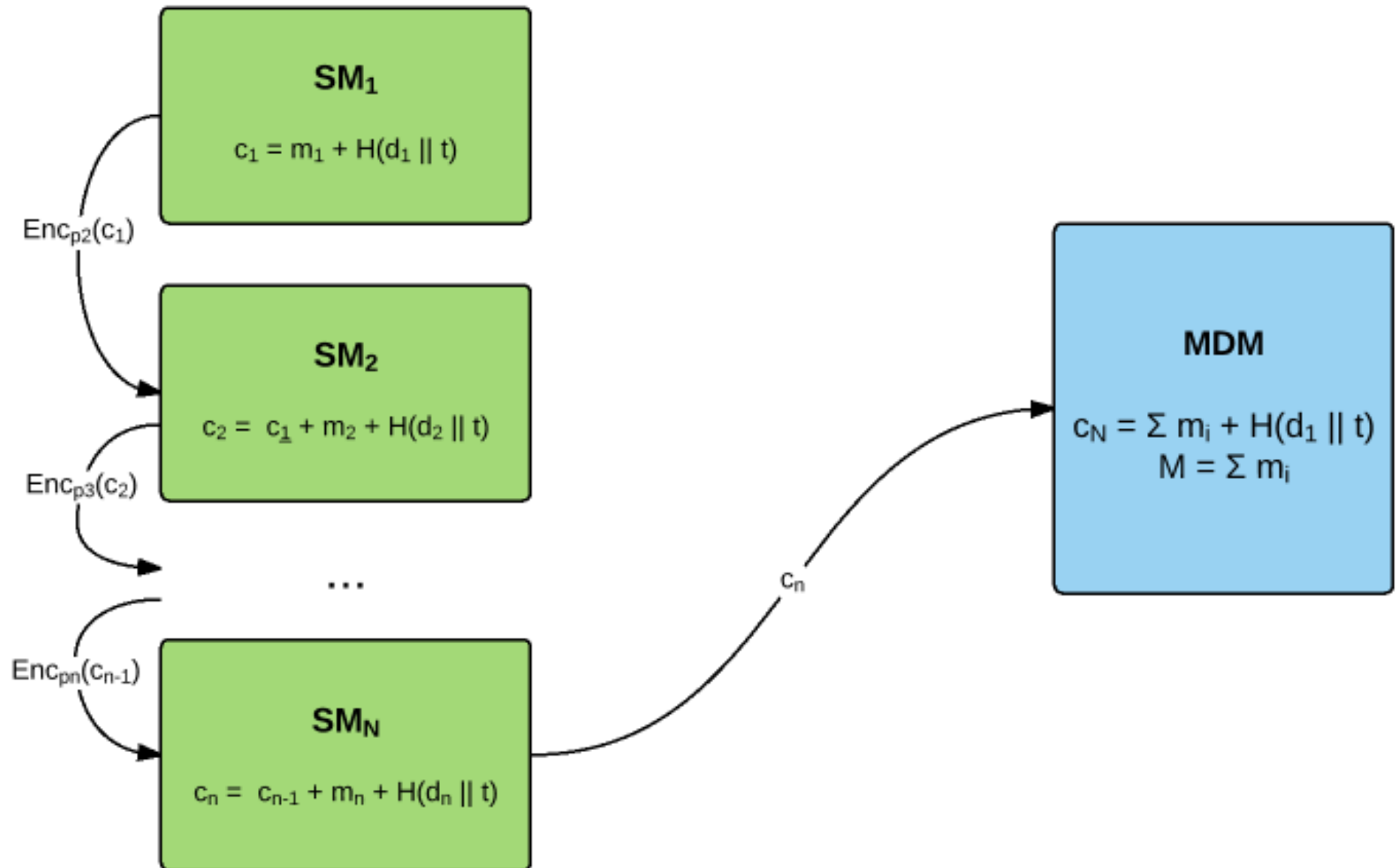
# Privacidade em Smart Metering por Borges et al. (iKUP)

**3-** A concessionária obtém do último medidor

$$c_N = \sum m_i + H(d_1 || t)$$

- Como ela possui  $\{d_1, d_2, \dots, d_n\}$  e sabe os timestamps, ela consegue obter  $M = \sum m_i$

# Privacidade em Smart Metering por Borges et al.

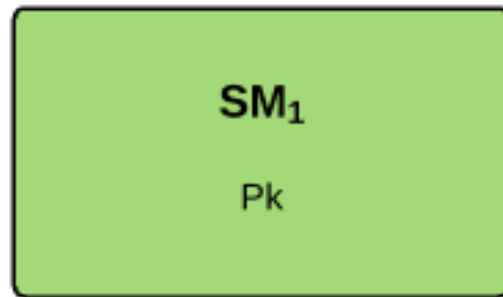


# Privacidade em Smart Metering por Borges et al. (Usando Paillier)

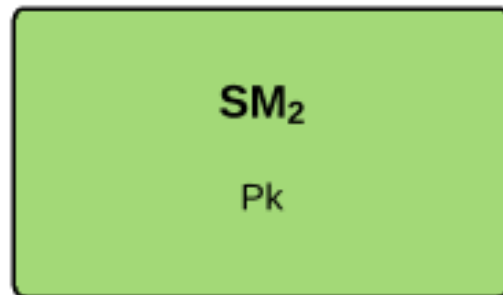
## 1- Cada medidor possui:

- As chaves públicas dos vizinhos (ex.,  $p_{i-1}$  e  $p_{i+1}$ )
- A chave pública de Paillier da concessionária ( $P_k$ )
- A concessionária possui uma chave privada de Paillier ( $S_k$ )

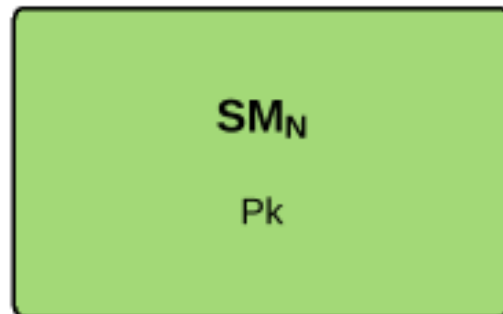
# Privacidade em Smart Metering por Borges et al. (Usando Paillier)



Medidores formam uma  
spanning tree



...



# Privacidade em Smart Metering por Borges et al. (Usando Paillier)

**2-** O medidor  $SM_1$  calcula a sua medição  $m_1$ , e para um instante de tempo  $t$  computa:

$c_1 = Enc_{PK}(m_1)$ . Em seguida, ele envia  $c_1$  para  $SM_2$

- $SM_2$  então computa  $Enc_{PK}(m_2)$  e  $c_2 = c_1 \cdot Enc_{PK}(m_2)$ , e envia para  $SM_3$
- Este processo continua até  $SM_N$ , que por fim, envia  $c_n$  para a concessionária

# Privacidade em Smart Metering por Borges et al. (Usando Paillier)

**3-** Devido a propriedade homomorfica de Paillier concessionária obtém do último medidor

$$c_N = \text{Enc}_{pk}(\sum m_i)$$

– Ao descriptografar, ela obtém  $M = \sum m_i$

# Privacidade em Smart Metering por Borges et al. (Usando Paillier)

