Paillier Encryption and its Application in Aggregation and Billing with Smart Meters

Le Chen

Nanyang Technological University

lechen0213@gmail.com

September 21, 2013
Overview

- Introduction
- Paillier Encryption
- Aggregation and Billing
- Analysis
- Conclusion & Discussion
Reference


Importance

- The Paillier encryption scheme, like the RSA, Goldwasser-Micali, and Rabin encryption schemes, is based on the hardness of factoring a composite number \( N \) that is the product of two large primes.
- The Paillier encryption scheme is *more efficient* than the Goldwasser-Micali cryptosystem, is *as well as* the provably-secure RSA and Rabin schemes.
- The Paillier encryption scheme possesses some nice *homomorphic* properties.
Proposition

Let $N = pq$, where $p, q$ are distinct odd primes of the same length. Then:

- $(N, \Phi(N)) = 1$
- For any integer $a \geq 0$, we have $(1 + N)^a = (1 + aN) \mod N^2$.
- The order of $(1 + N)$ in $\mathbb{Z}^*_{N^2}$ is $N$. 
Proof

Since \( \Phi(N) = (p - 1)(q - 1) \), assume \( p > q \), \( (p, \Phi(N)) = 1 \).
If \( (N, \Phi(N)) \neq 1 \), the only possibility is that \( (N, \Phi(N)) = q \),
then \( q | p - 1 \). But \( (p - 1)/q \geq 2 \) contradicts the assumption
that \( p \) and \( q \) have the same length.

Using the binomial expansion theorem. It is obvious that
\( (1 + N)^a = (1 + aN) \mod N^2 \).

According to the above result, \( (1 + N)^N = 1 \mod N^2 \). And
for any \( 1 \leq a < N \), \( 1 < (1 + aN) < N^2 \). Thus the smallest
non-zero \( a \) such that \( (1 + N)^a = 1 \mod N^2 \) is therefore
\( a = N \).
Paillier Encryption Scheme

Encryption

- Public key: $N$.
- Private key: $\Phi(N)$
- Plaintext: $m \in \mathbb{Z}_N$
- Encryption. The sender generates a ciphertext $c \in \mathbb{Z}_N^{*2}$ by choosing a random $r \in \mathbb{Z}_N^{*}$ and then computing

$$c := [(1 + N)^m \cdot r^N \mod N^2].$$
For ciphertext $c$ constructed as above, given the factorization of $N$, or equivalently given $\Phi(N)$, $m$ is recovered by the following steps:

- Set $\hat{c} := [c^{\Phi(N)} \mod N^2]$.
- Set $\hat{m} := (\hat{c} - 1)/N$ (No mod here).
- Set $m := [\hat{m} \cdot \Phi(N)^{-1} \mod N]$.
Correctness

Check the correctness:

\[ \hat{c} = \left[ (1 + N)^{m \cdot \Phi(N)} \cdot r^{N \Phi(N)} \mod N^2 \right] \Phi(N^2) = N \Phi(N) \]
\[ = \left[ (1 + N)^{m \cdot \Phi(N)} \mod N^2 \right] \]
\[ = \left[ (1 + m \cdot \Phi(N) \cdot N) \mod N^2 \right] \quad (1 + N)^a = 1 + aN \mod N^2 \]
\[ = 1 + \left[ m \cdot \Phi(N) \mod N \right] \cdot N, \]

\[ \hat{m} = (\hat{c} - 1)/N \]
\[ = \left[ m \cdot \Phi(N) \mod N \right], \]

\[ m = \left[ \hat{m} \cdot \Phi(N)^{-1} \mod N \right]. \quad (N, \Phi(N)) = 1 \]
Homomorphic Encryption

If we let $Enc_N(m)$ denote the Paillier encryption of a message $m \in \mathbb{Z}_N$ with respect to the public key $N$, we have

$$Enc_N(m_1) \cdot Enc_N(m_2) = Enc_N([m_1 + m_2 \mod N])$$

for all $m_1, m_2 \in \mathbb{Z}_N$. To see this, one can verify that

$$(1+N)^{m_1} \cdot r_1^N \cdot (1+N)^{m_2} \cdot r_2^N = (1+N)^{[m_1+m_2 \mod N]} \cdot (r_1 r_2)^N \mod N^2,$$

and the latter is a valid encryption of the message $[m_1 + m_2 \mod N]$. 

Le Chen
Nanyang Technological University
lechen0213@gmail.com
Paillier Encryption and its Application in Aggregation and Billing with Smart Meters
Motivation

For example, we consider the aggregation and billing of 4 users’ daily electricity usage within 7 days:

\[
\begin{pmatrix}
U_1 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\
U_2 & m(1,1) & m(1,2) & m(1,3) & m(1,4) & m(1,5) & m(1,6) & m(1,7) \\
U_3 & m(2,1) & m(2,2) & m(2,3) & m(2,4) & m(2,5) & m(2,6) & m(2,7) \\
U_4 & m(3,1) & m(3,2) & m(3,3) & m(3,4) & m(3,5) & m(3,6) & m(3,7) \\
S_1 & m(4,1) & m(4,2) & m(4,3) & m(4,4) & m(4,5) & m(4,6) & m(4,7) \\
S_2 & T_1 & T_2 & T_3 & T_4 & S_3 & S_4 & S_5 & S_6 & S_7
\end{pmatrix}
\]

Say, we want know $S_j$(aggregation) and $T_i$(billing) without reveal $m_{ij}$, for $i = 1, ..., 4, j = 1, ..., 7$. 
Weak Privacy-preserving Aggregation and Billing

It can be solved by simply applying Paillier Encryption:

- Users encrypt their electricity usage with Paillier Encryption.
- The CG (Community Gateway) computes the sum of ciphertext.
- The Utility decrypts the sum.

Advantages:

- The user electricity usage privacy is protected.
- Aggregation saves communication and computing overhead of the Utility.
Weak Privacy-preserving Aggregation and Billing

Disadvantages:

- If the Utility and CG collude, each individual user’s electricity usage can be revealed.
- A centralized CG is needed.
Step 1 - Exchanging Random Numbers

- Each smart meter $sm_i$ generates a random number $r(i\rightarrow j, t)$ and sends it to a peer $sm_j$.
- Next, each $sm_i$ computes $R(i, t)$ based on all collected randomness:

\[
R(i, t) = N + \sum_{j=1, i\neq j}^{k} r(i\rightarrow j, t) - \sum_{j=1, i\neq j}^{k} r(j\rightarrow i, t).
\]

- Notice that

\[
\sum_{i=1}^{k} R(i, t) = kN.
\]
Step 2 - Encrypting Measurements

- For each time interval $t$, each smart meter $sm_i$ computes a hash $h_t = H(t)$, where $H(\cdot)$ is a secure hash function, such that $(h_t, N) = 1$.

- Next, $sm_i$ encrypts its measurement $m(i,t)$ as follows:

$$\text{Enc}_N(m(i,t)) = (1 + N)^{m(i,t)} \cdot h_t^{R(i,t)} \mod N^2.$$  

Note that no one in the smart grid can decrypt the individual encryption because $h_t^{R(i,t)}$ is not a valid Paillier encryption, even if everyone has the decryption key.
Step 3 - Aggregation

- To obtain total usage within time $t$, any $sm_i$ multiplies all encrypted measurements, including its own:

$$\prod_{i=1}^{k} \text{Enc}_N(m(i,t)) = \prod_{i=1}^{k} (1 + N)^{m(i,t)} \cdot h_t^{R(i,t)} \mod N^2$$

$$= (1 + N)\sum_{i=1}^{k} m(i,t) \cdot h_t^{\sum_{i=1}^{k} R(i,t)} \mod N^2$$

where

$$\sum_{i=1}^{k} R(i,t) = kN$$

thus

$$\begin{align*}
(1 + N)\sum_{i=1}^{k} m(i,t) \cdot h_t^{\sum_{i=1}^{k} R(i,t)} &= (1 + N)\sum_{i=1}^{k} m(i,t) \cdot h_t^{kN} \\
&= (1 + N)\sum_{i=1}^{k} m(i,t) \cdot (h_t^{k})^N \\
&= \text{Enc}_N(S_t) \mod N^2
\end{align*}$$
Step 4 - Billing

- To obtain total usage within $M$ time intervals, one may multiply all $M$ ciphertexts from the same $sm_i$:

$$
\prod_{t=1}^{M} \text{Enc}_N(m_{i,t}) = (1 + N) \sum_{t=1}^{M} m_{i,t} \cdot \prod_{t=1}^{M} h_t^{R(i,t)} \mod N^2,
$$

but it is impossible to decrypt the resulting ciphertext, since it does not represent a valid encryption.

- To decrypt it, an additional random number $R(i, M+1)$, must be provided by $sm_i$ such that the following condition is satisfied:

$$
R(i, M+1) = \frac{r^n}{\prod_{t=1}^{M} h_t^{R(i,t)}} \mod N^2
$$

where $r$ is a random number in $\mathbb{Z}_N^*$. 
Step 4 - Billing

Thus, we have
\[
\prod_{t=1}^{M} \text{Enc}_N(m(i,t)) \cdot R(i,M+1) = (1 + N) \sum_{t=1}^{M} m(i,t) \cdot r^n \mod N^2
\]
\[
= (1 + N)^{T_i} \cdot r^n \mod N^2.
\]

which can be decrypted properly.
Security and Privacy

- Collusion resistance. For smart meter $sm_i$, unless all other $k-1$ users collude with the utility, its electricity usage will not be revealed.

- Detailed usage. Only $S_t$ or $T_i$ can be computed, detailed electricity usage is protected.
Efficiency

- Shared random numbers. Smart meters can exchange the seeds of their pseudo-random number generators when they initially become active.

- Complexity. The proposed cryptographic protocol is only based on performing encryption, hash function and random number generation, which are all highly efficient.
Flexibility

- Decryption key. The decryption key can be privately protected by the utility or community gateway, or disseminated to all users, based on the specific application.
- Processing without CG. Aggregation and billing can be processed by any user (smart meter).
- User addition. Each old user exchanges random numbers with the newly added user.
- User deletion. Each smart meter ignores the deleted user’s random number when computing $R(i,t)$. 
Additional Properties

- Malfunction in billing. When malfunction of some smart meter occurs, the utility can ask the smart meter manufacturer to provide an additional random number to support decryption.

- Multiple measurement. In practice, a number of measurements can be packed into one plaintext:

\[ \hat{m}(i,t) = m(i_1,t) | m(i_2,t) | \cdots | m(i_L,t) \]
Disadvantage

- Random numbers exchange. Exchanging random numbers, even exchanging the seeds of their pseudo-random number generators, may cause heavy communication overhead, especially in large community.

- Malfunction in aggregation. When malfunction of some smart meter occurs, the aggregated usage data can not be decrypted.
The proposed scheme uses a modified Pallier encryption to achieve strong privacy-preserving aggregation and billing with smart meters. It protects the individual smart meter’s electricity usage privacy with efficient and flexible distributed system.

Discussion?