A Zero Knowledge Building Temperature Controller

Cory Thoma, Washington & Jefferson College
Tao Cui and Franz Franchetti, Carnegie Mellon University

Problem

Smart Grid technology is the future of power distribution, but has some minor problems:

- Users reluctance to share raw energy usage data with power companies.
- Companies reluctance to reveal price levels compared to other power companies.

For example; power companies can see what an individual house's usage of air conditioning is, but users may not want the power companies to know how much they use it.

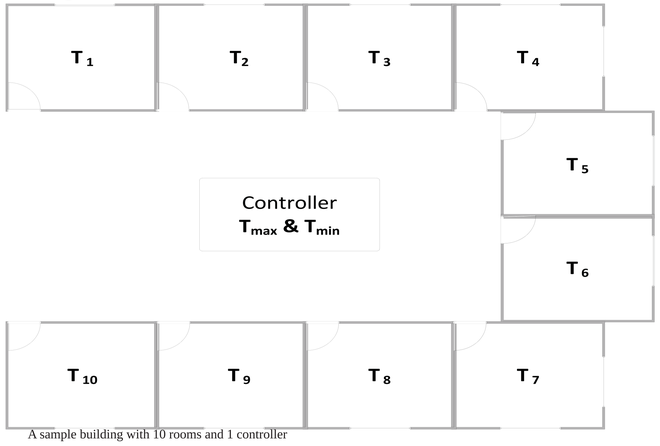
Solution

The solution to the energy usage data sharing problem of the Smart Grid technology is in implementing a zero-knowledge data sharing environment.

In a zero-knowledge environment, all participants can compare and compute any usage data among themselves without giving individual data information away.

Example Problem

To illustrate the idea how to share data in the Smart Grid, a simpler problem was proposed, controlling the temperature of a building.



A sample building with 10 rooms and 1 controller

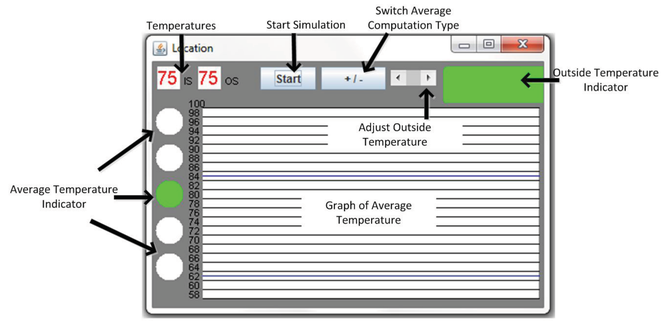
Consider that:

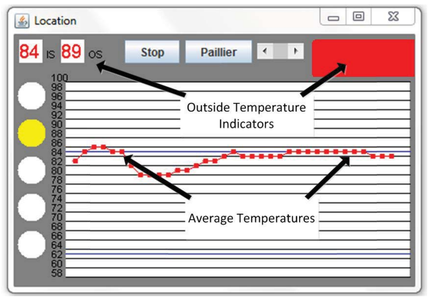
- Each room i has temperature T_i that it would like to keep private
- The controller has an acceptable temperature range (T_{max} to T_{min}) that it would like to keep private

The controller needs to compute the average temperature without knowing any T_i. If the average is above T_{max} or below T_{min}, then the controller checks each room to see if it needs to be cooled or heated, but does so without knowing T_i.

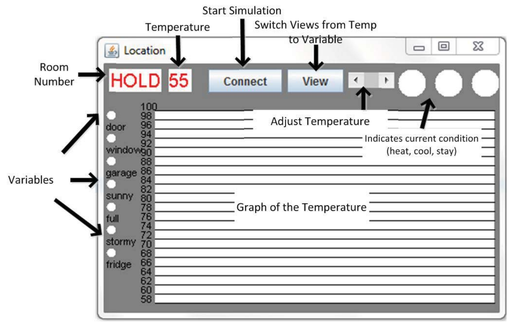
Implementation

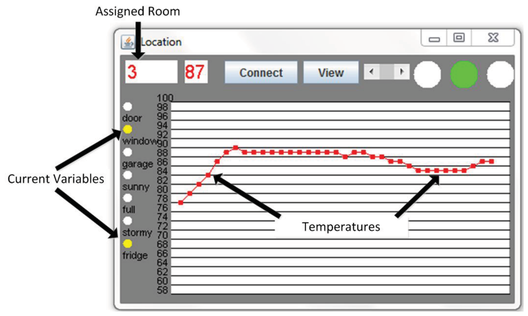
Controller





Room





Implementation Theory

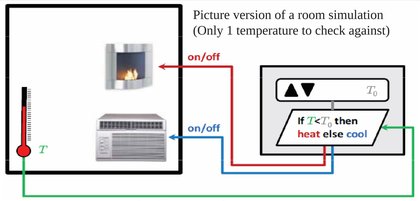
There are two parts to the implementation:

- The Controller
- The Rooms

The implementation is focused on keeping the buildings average temperature within reasonable limits. Each room is given a set of variables to change the temperature based on probabilities of an event happening. These variables are what influence the average temperature, and the controller. The controller computes the average and does single comparisons based on these temperatures, but does so without knowing the temperatures individually.

The average temperature can be computed in two ways:

- A random number is selected by the room and added to each rooms temperature. Then these numbers are passed between rooms and added together. The random values are subtracted after another pass and the number left is divided by the number of rooms to give an average. This is the +/- implementation.
- A homomorphic encryption scheme is used, the Paillier encryption scheme. Each room encrypts their temperature. The encryptions are added, and the controller decrypts and divides by the number of rooms to get the average.



Although each room gets information on the other rooms' temperature, there is no worry of eavesdropping because the temperatures are either encrypted or essentially random

If the average temperature exceeds T_{max} or does not reach T_{min}, then the controller uses the algorithm below to compare each room's temperature individually.

- **Step 1.** Bob (room) picks a random n-bit integer called x (Alice (controller) will later use a $n/2$ bit prime, so the length of the integer is important). Bob first calculates c as the RSA encipherment of x using Alice's public key.
- **Step 2**. Bob transmits $c - b + 1$ to Alice.
- **Step 3.** Alice generates a series of numbers $y_1, y_2, \ldots, y_k$ such that y_i is the RSA decipherment (using her private key) of $c - b + i$.
- **Step 4.** Alice now generates a random $n/2$ bit length prime p. Alice then generates $z_1, z_2, \ldots, z_k$ by calculating $z_i = y_i \bmod p$. Note that p must be chosen so that all the z_i differ by at least 2.
- **Step 5.** Alice now transmits the prime p to Bob, and then sends k numbers u_i. The first few u_i are $u_1 = z_1$, $u_2 = z_2, \ldots, u_a = z_a$ with a being Alice's worth in millions. Then Alice adds 1 to all the remaining $k - a$ values u_i to be sent and sends $u_a + 1 = z_a + 1 + 1, \ldots, u_k = z_k + 1$.
- **Step 6.** Bob receives p and $u_1, \ldots, u_k$. He computes $g = x \bmod p$. If the $u_b = g$ then Alice is equal or greater in wealth ($a \geq b$). If the $u_b \neq g$ then Bob is wealthier than Alice ($a < b$).

Zero-Knowledge number comparison algorithm,
A. C. Yao, *Protocols for Secure Computation*

Planned Extensions

- Extend the system to simulate a Smart Grid environment.
- Allow the consumer to bid for power, and allow the power companies to offer prices to the consumer. This will be done through comparing price functions in a zero-knowledge environment to find the best price both agree on.