CSCI 5573 – Advanced Operating Systems

Smart Meter Privacy

Project Proposal

Aditya, Murali, ChitraLekha, Prashant
11/1/2011

Contents

Problem Statement 3
Background 3
Architecture 4
1. Secure Microcontrollers 5
2. GlobalPlatform 5
3. Java Card 5
Use-Cases 6
Billing 6
Settlement 6
Forecasting and Profiling 6
Other use-cases 7
Protocols and Data-Sets 7
DLMS/COSEM 8
Smart Meter Language (SLM) 8
Project Scope 8
References 8

Problem Statement

Smart meters, a novel approach, is targeted towards improving business efficiencies, enhancing electricity provisioning, improving billing and settlement processes, and providing added-value services to customers. However, they are plagued with Privacy concerns because it mandates these smart meters to take fine-grained fifteen-minutely electricity usage measurements.

Study has shown that these frequent meter readings can reveal behavioral patterns, for instance about whether the inhabitants are at home, or at what time they get up or go to bed. Refined data analysis/mining over longer periods may reveal further information, for instance about the kind of devices that are being used, at which time, etc. This clearly breaches consumer privacy and turns smart meter into a surveillance device ("spy-meter").

As a result, concerns over privacy are hampering the deployment of smart meters and makes it an interesting problem to solve. So, the goal of the project is to explore possible *technical* solutions to the Privacy concerns such that it satisfies the end goal of Utility companies without revealing too much information about the consumer.

Background

Smart Meter Privacy is an active area of research which is still under exploration with no real/complete solution to the problem yet. It is being tackled at various levels which range from putting legal laws and regulations in place to improving the technology my enhancing security and limiting data by use of aggregation.

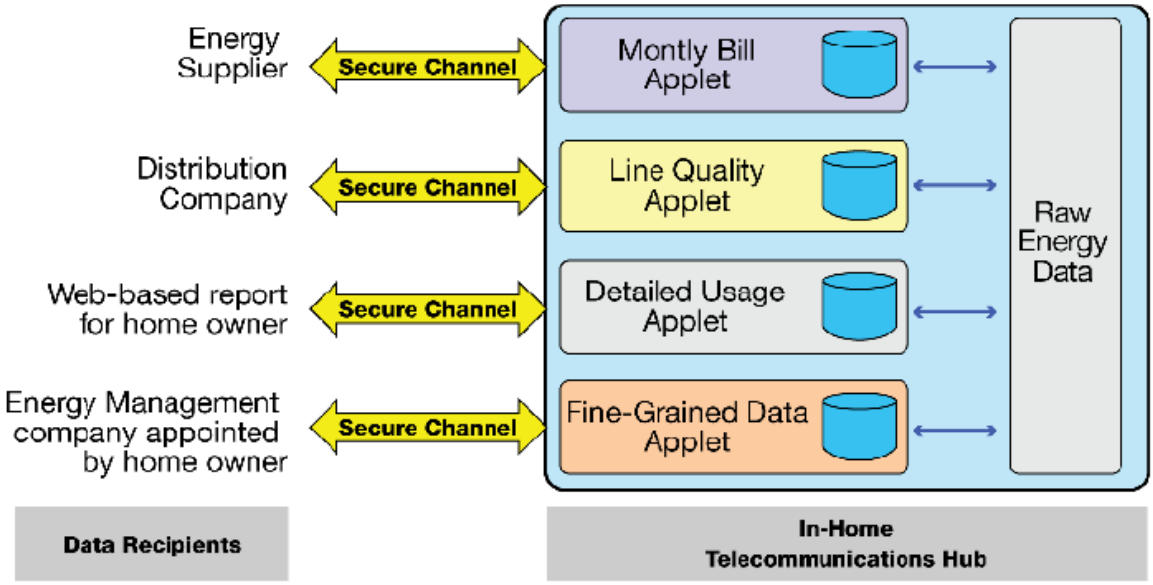
One prominent approach is "aggregation" using the concept of homomorphic cryptography. The main premise is that Utility companies are not interested in fine grained data and can satisfy most of the use-cases by an aggregate consumption data with reasonable granularity (say 1 hour). In this approach, there is an aggregator module within a network which is responsible for gathering data from a small neighborhood comprising of about 20 smart meter group. In this model, each smart meter sends its *encrypted* meter reading to aggregator (so aggregator doesn't know individual meter readings) which then computes the total consumption of the meter group using algebraic qualities of homomorphic encryption.

However, in our opinion this approach is limiting and solves particular set of use-cases which are aggregate consumption based like settlement and billing. We strongly believe that there is a case for fine-grained data to be available as well which can be used to provide value-added services like "appliance inference" and other energy management services.

A much better solution in our opinion is "Local Processing" where-in by default the raw consumption data would stay in the meter. Software running in the meter can process the raw data and export digested data to those entitled to it, such that the most sensitive personal information is no longer available within the digested form. Additionally, the fine-grained consumption data may be exported from the meter to entities that can process it to provide valuable services for the consumer, but only with the informed consent of the customer.

In fact we can make this solution even more general purpose. The data generated from various devices (appliances, electricity, gas meters etc) can be aggregated into an in-house hub device (similar to a DVR for cable TV). The device will have various software programs installed (for various use-cases like billing, settlement, forecasting etc) which will consume/process the data and then release it to various service providers (including utilities) over a secure channel.

Running the imagination horses further, one can also envision GoogleTV, AppleTV, PS3 et al coming up with such services as they already have all the capabilities (networking, processing, storage, security etc) in place.



Architecture

So, the question is then: what architecture will allow us to deploy these various software programs into the smart meter system, securely and reliably?

The answer comes from the banking industry, which has already solved this problem for smart cards. A set of standards known as GlobalPlatform address precisely the problem of securely managing the life-cycle of software running on smart cards. The security requirements of the banking, mobile phone and pay-TV industries are very similar to those of the smart meter industry: to protect revenues in a hostile environment, and to allow new software applications to be dynamically and securely deployed at the remote device (be that the smart card, mobile phone, set-top box or smart meter).

The key technology components of this solution are:

1. Secure Microcontrollers

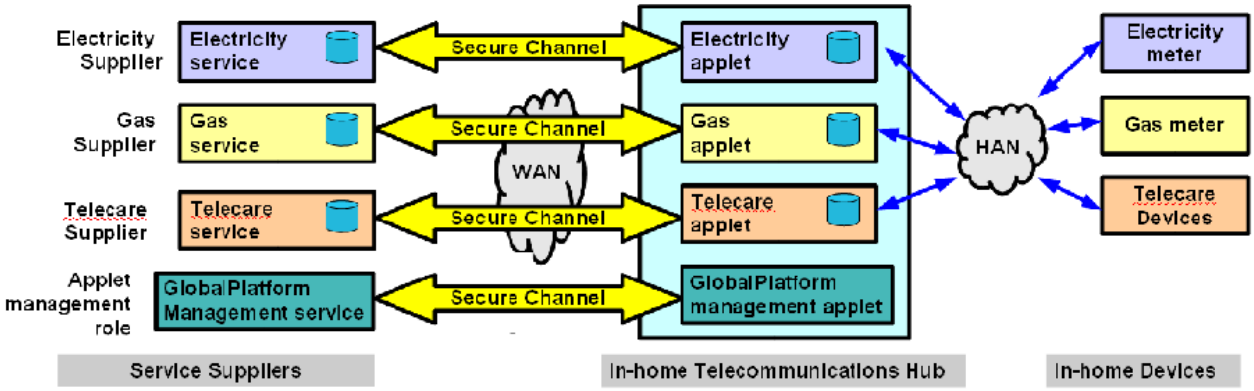
Secure microcontrollers are used in SIM cards, smart cards and pay-TV access cards. They include tamper-resistance features aimed at stopping hackers from copying or changing code and data. Importantly, the secure microcontroller protects the cryptographic keys.

2. GlobalPlatform

There must be a means of securely installing, personalizing and managing the life cycle of software that runs on the secure microcontrollers. GlobalPlatform is an industry standard that allows multiple application programs (applets), from multiple vendors, to be deployed on secure microcontrollers. So, GlobalPlatform can be used to dynamically and securely deploy multiple software programs onto smart meters.

3. Java Card

Java Card is a sub-set of the Java programming language, commonly used with secure micros and GlobalPlatform. Its design provides a secure environment for applications that run on devices with very limited memory and processing capabilities. Multiple programs (or "applets") can be deployed on a single secure micro, and new ones can be added to it even after it has been deployed in the field. Because of the standardized platform, applets written in the Java programming language can be executed securely on smart cards (or smart meter systems!) from different vendors.



The figure above shows these elements in action. The hub is implemented in a secure microcontroller running GlobalPlatform and Java Card software. Several applets run together, with their data separated by the intrinsic firewall features of the software architecture. The applets are managed remotely by a GlobalPlatform server, and GlobalPlatform establishes multiple "secure channels" each with their own cryptographic keys. The data processed by each applet is sent to the corresponding data recipient, encrypted by a key shared only with the data recipient.

Use-Cases

We intend to target following most prominent use case for Utility companies for this project:

Billing

Time-of-use tariffs are the most complex form of billing proposed relying on smart metering. It consists of different tariffs being applied depending on the time of day, day of week or day of year to the half-hourly customer consumption to determine its contribution to the final bill. All contributions are summed up to calculate a monthly or quarterly bill. Tariff structures – the time periods within which different tariffs apply – can be arbitrarily complex, and different between customers.

Technically the computation of a certified bill involves the selection of the correct tariff, a multiplication with the reading to determine its contribution to the final bill, and a sum of all contributions to determine the bills. So, there are 2 inputs for the computation:

a. The tariff policies which can be pushed to the Billing applet on regular basis by the utility companies.
b. Consumption profile which can be as simple as hourly consumption data.

One challenge is the how does the supplier trust the billing information provided by the consumer's device? For this we want to explore a zero-knowledge proof based approach using Pedersen Commitments proposed by SAP Research.

Settlement

The settlement process determines the fraction of grid production at any time that corresponds to customers of a particular supplier, and thus the amount a particular supplier should pay for electricity consumed by its customers. Currently, this faction is estimated according to the number of customers of each supplier and their profiles. In the future it will be desirable to enhance the accuracy of this process.

We can extend the privacy friendly billing solutions described in the previous section to implement "penny-accurate" settlement mechanisms. Meters not only output the bill due using the tariffs that apply to the customer, but also output a normalized version of the total value of the electricity consumed by the customer in the same period (say per month). This is simply the detailed meter readings multiplied by an indexed value of electricity as determined on the grid. Suppliers can use those values that are aggregated by customer over a month, and sum them up to determine the fraction of electricity used by those customers in that month.

Forecasting and Profiling

Current settlement processes are based on profiling users, and using those profiles to estimate the fraction of consumption per supplier. Profiles of users are also useful for the financial forecasts of suppliers.

Given a number of profiles it is possible to estimate how well a customer matches a profile through correlation. Assessing the match of all customers to a set of profiles would allow an unprecedented increase in accuracy when it comes to forecasting. The correlation between customers and profiles can be uncovered using sampled data, or again using computations on certified readings. Two families of methods can be used to extract profiles for a customer population: sampling, and exact matching.

Matching profiles through sampling is straight forward: meters are instructed to leak a sample of readings (again 1 in 100 would be equivalent to one reading every 2 days). Given those sampled readings users can be classified approximately as belonging to different classes. This is a statistical estimate, and subject to an error that depends on the rate of sampling and the regularity of profiles.

A second approach is to define a privacy friendly computation that takes as an input certified meter readings, and profiles, and outputs the probability or likelihood a customer belongs to each of the profiles. The technical details of these correlation calculations are very similar to billing – they involve mostly multiplications and sums. Thus they can be performed very quickly on meters or any other devices.

Other use-cases

1. **Fraud Detection:**
   Aggregate consumptions from different customers on the same sub-station can be summed up and computed with the total electricity served in an area. Systematic and serious discrepancies may indicate fraud or a fault in the network.

   A further fraud detection technique can be applied in a similar fashion to profiling: specific suspect profiles can be constructed and matched with detailed consumption records using privacy preserving computations. The result indicates the likelihood a user fits a suspicious pattern, and can be used to guide further investigations.

2. **Demand Response:**
   Demand-response is a mechanism that notifies customers and their appliances of the load on the network or cost of supplying electricity. They may choose to respond by shifting their individual consumption to different times to reduce costs or as part of a contractual obligation. This is an important functionality to spread load and lower energy demand peaks.

Protocols and Data-Sets

For Smart Meter reading and reporting protocols, we have identified two relevant application layer protocol specifications:

DLMS/COSEM

DLMS specifies how one can talk about energy metering objects. Energy Metering objects are described by the COSEM specification. The standard does not dictate specific transport protocols. The Smart Meter operates as server and communication follows the pull-strategy from the view of the BS system. Read and write access is realized by transmitting respective COSEM objects in APDUs (Application Protocol Data Units).

Smart Meter Language (SML)

SML describes an application and presentation layer and Smart Meters operate either in a push (SM initiates) or pull (SM reacts) scenario. All data is encoded in either an SML request or SML response message. Encryption of SML messages on the application or presentation layer is not part of the SML specification.

Project Scope

Basically, we intend to do a proof-of-concept implementation of the solution based on the architecture specified above and target the 3 main use-cases of billing, settlement and forecasting.

The main approach would be:

1. Use the Java-Card platform and its emulator to simulate the hub device so that we don't need a real hardware.
2. Implement 3 applets for the respective use-cases.
    a. The tariff and profiles data will be hard-coded in applets to avoid the GlobalPlatform complexity.
    b. The applets will implement the Pedersen commitment concept for establishing trust relationships.
3. Simulate the meter reading using DLMS or SLM protocol and output the results in same format.

References

1. Architecture + Use-Cases: Microsoft Research and Project Hydra
2. Java Card and Global Platform
3. Billing using Pedersen Commitment
4. Protocols: DLMS or SML
5. Energy Forecasting: Pricing and NREL