Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

Bachelorarbeit

Natalia Orlova

Schutz der Privatsphäre elektrischer Energieverbraucher in intelligenten Stromnetzen mittels homomorpher Verschlüsselung

Fakultät Technik und Informatik
Department Informatik

Faculty of Engineering and Computer Science
Department of Computer Science

Natalia Orlova

Schutz der Privatsphäre elektrischer Energieverbraucher in intelligenten Stromnetzen mittels homomorpher Verschlüsselung

Bachelorarbeit eingereicht im Rahmen der Bachelorprüfung

im Studiengang Technische Informatik
am Department Informatik
der Fakultät Technik und Informatik
der Hochschule für Angewandte Wissenschaften Hamburg

Betreuender Prüfer: Prof. Dr. Dirk Westhoff
Zweitgutachter: Prof. Dr. Thomas Schmidt

Abgegeben am 21. August 2011

Natalia Orlova

Thema der Bachelorarbeit

Schutz der Privatsphäre elektrischer Energieverbraucher in intelligenten Stromnetzen mittels homomorpher Verschlüsselung

Stichworte

Datenaggregation, homomorphe Verschlüsselung, intelligente Stromnetze, intelligente Stromzähler, Netzwerktopologie, regenerative Energie, Schlüsselverwaltung, Schutz der Privatsphäre, Sensorknoten, Sensornetzwerke

Kurzzusammenfassung

In dieser Arbeit werden theoretische Lösungen für den Schutz der Privatsphäre in einem intelligentem Messsystem analysiert. Weiterhin werden die Lösungen mit den konkreten Anforderungen für intelligente Netze verglichen. Dabei werden die Schwäche und Stärke einzelner Ansätze ermittelt und einen geeigneten Algorithmus für die prototypische Implementierung ausgewählt. Außerdem wird hier eine Prototypische Implementierung des intelligenten Netzes, die einen Algorithmus mit homomorpher Verschlüsselung für den Schutz der Privatsphäre nutzt, dargestellt. Dabei werden eine Simulation des intelligenten Netzes sowie eine Applikation in NesC und Java erstellt, die auf 16-Bit Mikroprozessoren läuft.

Title of the paper

Using homomorphic encoding for private sphere protection of electrical consumers in a smart grid

Keywords

data aggregation, homomorphic encoding, key distribution, network topology, private sphere protection, regenerative energy, smart grid, smart meters, sensor networks, sensor nodes

Abstract

In this paper theoretical solutions for private sphere protection in a smart meter system are analyzed. Furthermore the solutions are mapped onto practical requirements for smart grids. In the course of the comparison the most appropriate algorithm for prototype implementation is chosen. The thesis also contains a description of a smart grid prototype with a private sphere protection, implemented with the help of a homomorphic encoding algorithm. The implementation consists of a simulation and an application in NesC and Java, that runs on 16-Bit microprocessors.

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

1 Einleitung

1.1 Motivation

Ein intelligentes Stromnetz, auch ”*Smart grid*” genannt, stellt ein komplexes System dar, das verbesserte Möglichkeiten für effiziente Stromverbrauch bietet. Nach heutigem Stand der Technik sind solche Systeme, die besonderen Wert auf kontinuierliche Rückkopplung mit Endverbrauchern und integrierte regenerierbare Stromquellen legen, schon lange eine Realität.

Ein intelligentes Stromnetz bringt mehrere Vorteile mit sich. Die Energieversorgungsunternehmen bekommen ein Werkzeug zur Verfügung, das schnelle Reaktion auf Änderung im Netz erlaubt. Die aktuellen Zählerstände werden automatisch über das Internet übermittelt. Die kontinuierlich gelesene Daten können effizienter für statistische Auswertung genutzt und letztendlich für optimale Stromeinkaufs- und Verkaufsstrategien eingesetzt werden. Die Stromproduktion aus grünen Quellen (z.B. Wind oder Sonnenenergie), die vom Wetter abhängig ist, kann anhand der Daten, die eine intelligenten Stromzähler liefert auch effizienter kontrolliert werden. Die Endverbrauchern profitieren auch von Einsatz von intelligenten Stromnetzen und Stromzähler, die laufende Tarifinformation liefern. So können die Endverbraucher den Strom effizienter einkaufen (Petrlic, Februar 2011).

Energiewirtschaftsgesetz (EnWG) schreibt schon ab dem Jahr 2010 Einführung von intelligenten Stromzähler bei den Neubauten oder größeren Renovierungen vor, ”soweit dies technisch machbar und wirtschaftlich zumutbar ist”(EnWG, Stand 2008). Trotz aller Vorteile ist die tatsächliche Einführung von intelligenten Stromnetzen mit mehreren Sicherheitsproblemen verbunden. In unterschiedlichen Informationstechnischen Kontexten werden folgende Aspekte der Sicherheit hervorgehoben: Verfügbarkeit, Vertraulichkeit, Integrität und Authentizität. Verfügbarkeit von intelligenten Stromnetzen beinhaltet alle Maßnahmen gegen Systemausfall und Schutzmechanismen, die sichere Systemwiederherstellung garantieren. Ein vertrauenswürdiges intelligentes Stromnetz soll verhindern, dass die Daten über die menschlichen Profile gesammelt und nach außen gegeben werden. Integrität garantiert, dass die Daten vollständig und geschützt von Änderung durch Angreifer sind. Authentizität soll sicher stellen, dass beide kommunizierende Parteien, Energieversorger und Energieverbraucher, sind tatsächlich diejenigen zwischen denen die Kommunikation stattfindet (Müller, Februar 2011). Jeder von diesen einzelnen Aspekten braucht eigene standardisierte Lösung um ein sicheres intelligentes Stromnetz zu konstruieren.

Der Schutz der Privatsphäre ist eine der wichtigsten und spannenden Forschungs- und Anwendungsfragen, die unbedingt berücksichtigt werden soll, bevor ein intelligentes Stromnetz in Betrieb genommen wird. Zuerst sollen hier mehrere rechtliche Fragen beantwortet

werden. Welche Daten sind überhaupt für Sammlung bei einem Energieversorgungsunternehmen erlaubt? Gibt es ein Gefahr, dass diese Daten potenziell missbraucht werden können? Wo liegt die Grenze der Privatsphäre? Ist es überhaupt akzeptabel, dass aus den gesammelten Daten ein Profil über die menschlichen Aktivitäten gebildet werden kann? Daraus können z.B. Rückschlüsse gezogen werden, ob die Kunden zu Hause sind oder nicht, wann sie schlafen gehen oder aufstehen, wie oft und wann sie bestimmte Haushaltsgeräte benutzen, etc. Das Problem, dass in diesem Zusammenhang zu lösen lautet: wie wäre es möglich alle Vorteile von intelligenten Netzen und intelligenten Stromzähler zu nutzen und trotzdem die Privatsphäre einzelner Menschen nicht zu verletzen. Diese herausfordernde Forschungsfrage mit offensichtlicher praktischer Relevanz fordert tiefe Analyse und Kenntnisse aus mehreren Bereichen der Informatik: Kryptographie, Mathematik, Sensornetzen, Netzwerktopologie und Verteilten Systemen.

1.2 Zielsetzung

In dieser Arbeit soll eine Lösung für das Problem des Schutzes der Privatsphäre mittels homomorpher Verschlüsselung entwickelt werden. Als wissenschaftlicher Basis für diese Arbeit werden die aktuellsten Lösungsansätze für intelligente Stromnetze der letzten 5 Jahre genommen und analysiert. Mehrseitige Vergleich der schon existierenden oder gerade geforschten Hauptlösungen soll helfen, die aktuelle Stand der Entwicklung in intelligenten Stromnetzen zu definieren. Das Vergleich hilft auch zusammenzufassen, was schon erreicht wurde und welche Nachteile und Vorteile die bis jetzt vorgeschlagene Lösungen und Einsätze hätten. Weiterhin wird Versucht die theoretische Lösungen auf praktische Anforderungen für intelligente Messsysteme zu übertragen. Dabei soll festgestellt werden, welche Lösungen in der Praxis eingesetzt werden können und welche Verbesserungen dafür notwendig sind.

Der Hauptschwerpunkt wird in dieser Arbeit auf die Entwicklung eigener Lösung des Problems des Schutzes der Privatsphäre in einem intelligenten Stromnetz gelegt. Die Lösung wird anhand der theoretischen Vergleich der existierenden Algorithmen entstehen. Dabei wird das gewählten Verfahren so modifiziert, dass es besser für den praktische Einsatz geeignet ist. Das Konzept wird auch angesichts der möglichen Probleme vorgestellt.

Anhand des entwickelten Konzeptes soll ein Prototyp implementiert werden, dass das entwickelte System simuliert und dieses Funktionsfähigkeit beweist. Die Applikation soll auf echten Mikroprozessoren für kleine Anzahl der Sensoren installiert und getestet werden. Das Prototyp soll für 1000 Sensorknoten als Simulation getestet werden. Der Testverlauf soll Lauffähigkeit des Programms und des Algorithmus für den Schutz der Privatsphäre zeigen. Alle Ergebnisse sollen gesammelt und beschrieben werden. Diese Analyse wird als Basis für zukünftige Verbesserung genutzt.

1.3 Aufbau der Arbeit

Kapitel 1 enthält eine Einleitung sowie die Vorstellung des Problems des Schutz der Privatsphäre in intelligenten Stromnetzen.

In Kapitel 2 werden Anforderungen an Systeme mit intelligenten Stromzähler analysiert. Zuerst werden hier alle wichtigste Begriffe definiert und eingeführt. Danach werden die Anforderungen anhand des Schutzprofils des Bundesministeriums für Sicherheit in der Informationstechnik (BSI) formuliert und zusammengefasst.

In Kapitel 3 werden theoretische, in der Literatur vorgeschlagene, Ansätze für den Schutz der Privatsphäre in intelligenten Stromnetzen dargestellt. Hier werden die Algorithmen und Verschlüsselungsverfahren beschrieben.

In Kapitel 4 wird eine Analyse dargestellt, um feststellen zu können, in wie fern die theoretische Ansätze die Kriterien des BSI Schutzprofils erfüllen. Nachteile und Vorteile der einzelnen Lösungen werden für solche Aspekte, wie Software und Hardwarekomponente, Sicherheitsfunktionen, kryptographischer Support, Schutz gegen Angriffe und Schutz der Privatsphäre zusammengefasst.

In Kapitel 5 wird ein Konzept für die Implementierung, basierend auf die Lösungsansätze, dargestellt. Für das Konzept wird einen Algorithmus mit homomorpher Verschlüsselung gewählt. Deshalb werden hier auch Vorteile von homomorpher Verschlüsselung beschrieben.

Kapitel 6 präsentiert die Implementierung des Algorithmus mit homomorpher Verschlüsselung für den Schutz der Privatsphäre. Hier wird detailliert auf die Applikationsaufbau und Zusammenspiel der Komponenten eingegangen. Weiterhin werden hier die Testfälle von Applikationen, die auf Mikroprozessoren und im Simulator laufen, beschrieben.

In Abschlusskapitel 7 werden alle Ergebnisse noch mal zusammengefasst, Verbesserungsvorschläge und einen Ausblick in die zukünftige Forschung und Entwicklung gegeben.

2 Anforderungen an intelligente Messsysteme

2.1 Begriffsdefinition

Intelligentes Stromnetz und intelligenter Stromzähler sind relativ neuen Begriffe, die über keine eindeutige Definition verfügen. Verschiedene Autoren geben unterschiedliche Definitionen an. Die sind unter anderem:

- V.J. Forte (Forte, 2010) bezeichnet ein intelligentes Stromnetz als ein System, das aus mehreren Elementen besteht. Dazu gehören intelligente elektrische Verteiler, Stromzähler die in beide Richtungen kommunizieren und spezialisierte Rechnersysteme. Der Einsatz von diesen Elementen soll die Effizienz und Stabilität des Netzes steigern. Darüber hinaus sollen die Kunden zu einem kostenbewussten Verbrauch bewegt werden.

- *European Technology Platform* definiert ein intelligentes Stromnetz wie folgt:
  ein intelligentes Stromnetz ist ein Stromnetz, dass alle Aktivitäten der verbundenen Benutzern (Energieversorger, Kunden und diejenige, die beiden Rollen spielen) intelligenter weise integriert. Dabei wird Strom effizient, nachhaltig und sicher geliefert (ETP, 2006).

- Laut US Energieministerium benutzt ein intelligentes Stromnetz digitale Technologien zu Verbesserung der Zuverlässigkeit, der Sicherheit und der Effizienz (sowie Wirtschaftseffizienz als auch Energieeffizienz) in großen Elektroenergiesystemen. Die beinhalten Lieferungssysteme zu Verbrauchern und die wachsenden Anzahl von dezentralisierten Energieerzeuger und Speicherressourcen (SGSR, 2009).

	traditionelles Stromnetz	**intelligentes Stromnetz**
Kommunikation	keine oder einweg	zweiweg
Interaktion mit Benutzer	selten	viel
Instrumenttyp	elektrisch	numerisch
Operation und Steuerung	künstliche Gerätesteuerung	Fernkontrolle
Flusskontrolle	beschränkt	universal
Zuverlässigkeit	Störungstendenz und Stromunterbrechung	adaptiver Schutz
Stromwiederherstellung	künstlich	selbst-heilend
Topologie	radial	netzförmig

Tabelle 1: Vergleich zw. traditionellem und intelligentem Stromnetz (Xue-Song u. a., 2010)

In der Tabelle 1 wird gezeigt, wie sich ein intelligentes Stromnetz von einem traditionellen Stromnetz unterscheidet. Wie man aus dem Vergleich sieht, liegt der Unterschied hauptsächlich in der Rückkopplung zwischen dem Energieversorgungsunternehmen und dem Kunde. Die allgemeine Computersteuerung des Netzes und digitale Technologien spielen natürlich bei einem intelligenten Netz eine große Rolle. Diese Ressourcen als solche sind allerdings schon lange vorhanden in einem modernen, aber sogenannten traditionellen Netz.

Abbildung 1: Intelligentes Stromzähler von Firma Meterus. Quelle: www.energiesparclub.de

Die Rückkopplung wird durch Einführung von intelligenten Geräten erreicht, insbesondere intelligenten Stromzähler. Der Begriff “Intelligenter Stromzähler” hat auch keine eindeutige Definition. Alle Systeme mit intelligentem Stromzähler bestehen aus einem elektronischen Gerät und einer Kommunikationsverbindung. Ein intelligenter Stromzähler misst, der Stromverbrauch, und überträgt diese Information an andere Geräte. Die Kunden sehen wie viel Strom sie verbrauchen und wie viel das kostet. Abbildung 1 zeigt einen typischen intelligenten Stromzähler.

Für Visualisierung des Verbrauches der einzelnen Kunden kommt spezielle Software in Einsatz. Die Kunden sehen, wie viel Energie und zu welcher Uhrzeit sie verbrauchen. Sie können damit bessere Strategie für sich bestimmen. Abbildung 2 zeigt ein typisches Verbrauchsgraph.

Die Energieversorgungsunternehmen müssen monatlich die Rechnungen für die Kunden ausstellen, mit oder ohne intelligenten Stromzähler. Dafür brauchen sie Verbrauchsdaten

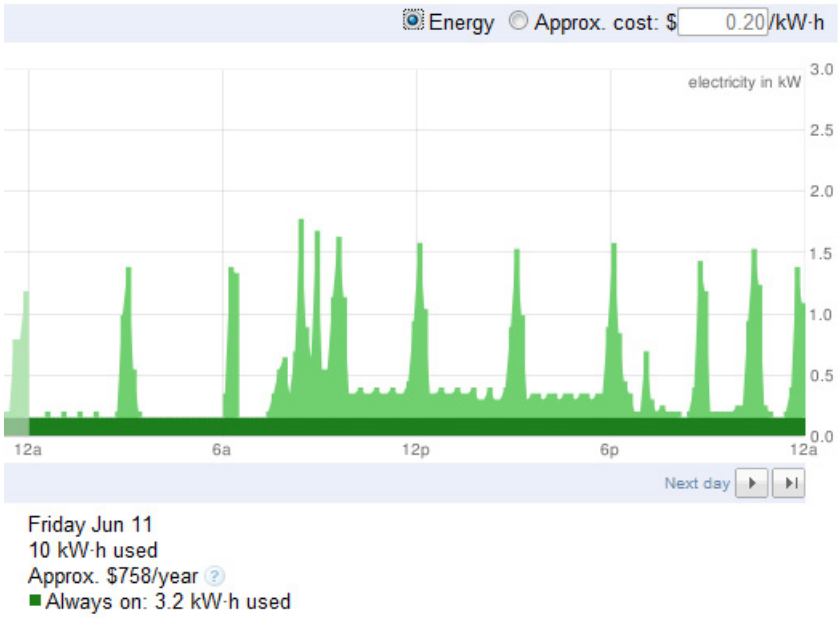

Abbildung 2: Stundenmessung mit *Google powermeter*. Quelle: http://steplight.com.au

der Kunden. Aus diesen Daten lässt sich auch Strategie für Stromeinkauf und Verkauf entwickeln. Die Frage ist nur wie viel Information die Energieversorgungsunternehmen wirklich brauchen (Garcia und Jacobs, 2010). Für Rechnungen ist gesammelte nicht-kontinuierliche Information völlig ausreichend. Für Statistik ist Information über Nutzermustern wichtig. Der Schutz der Privatsphäre erlaubt aber nicht zu detaillierte Wissen über einzelne Kunden und ihre Gewohnheiten. Diese Wissen allerdings lassen sich durch stündige Übertragung der Energieverbrauchsdaten gut zusammentragen. So die häufige Datenübertragung kann einen Verstoß gegen Privatsphäre in vielen Hinsichten sein:

- Tägliche Messungen machen es klar, wann die Bewohner zu Hause oder verreist sind. Diese Information wäre bestimmt interessant für Kriminelle.
- Stundenmessungen können sogar viel mehr über die Bewohner aussagen. Die Messungen (in Kombination mit Namen) können zeigen, ob jemand z.B. ein gläubiger Moslem ist, weil er gegen 5 Uhr jeden Tag zum Betten aufsteht. Abbildung 3 zeigt Stundenmessungen für Gas, Wasser und Elektrizität in einem bestimmten Haushalt. Es sieht so aus, dass die Bewohner gegen 17 Uhr nach Hause kommen und duschen gegen 1 Uhr in der Nacht.
- Langfristige Beobachtungen helfen herauszustellen, welcher Geräte am häufigsten im

Haushalt genutzt werden, welche sind nicht mehr effizient und müssen bald ausgetauscht werden. Diese Informationen sind natürlich sehr interessant für Händler (Garcia und Jacobs, 2010).

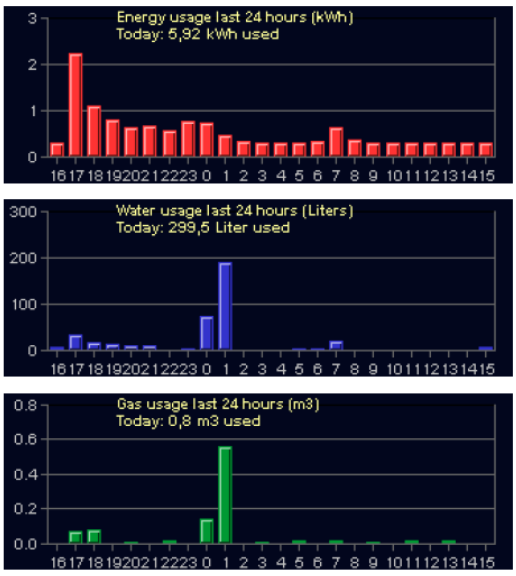

Abbildung 3: Beispiel: Wasser-, Gas- und Strommessung (Garcia und Jacobs, 2010)

Erfolgreiche Einführung der intelligenten Messsysteme braucht offensichtlich nicht nur Vorschriften für Hardware- und Softwarekomponenten, sondern auch klare Formulierung der kryptographischen Zielen, damit der Schutz der Privatsphäre in so einem System garantiert wird. Um so einen Standard zu etablieren wurde in September 2010 das Bundesamt für Sicherheit in der Informationstechnik (BSI) vom Bundesministerium für Wirtschaft und Technologie (BMWi) mit der Entwicklung eines Schutzprofils für intelligente Stromzähler beauftragt. In diesem Profil werden alle Anforderungen an Hardware, Software und kryptographische Ziele für ein intelligentes Messsystem formuliert. Das Schutzprofil beschreibt die generalisierte Sicherheitseigenschaften der Komponenten eines intelligenten Messsystems (PP, 2011).

2.2 Hardwarekomponenten und Schnittstellen

BSI gibt in dem Schutzprofil für die Kommunikationseinheit eine Übersicht über ein Messsystem mit den Hauptkomponenten und deren Schnittstellen (PP, 2011). Wie die Abbildung 4 zeigt, ist die Kommunikationseinheit eine zentrale Komponente des intelligenten Messsystems. Die Einheit ist für Aggregieren und Bearbeitung der Zählerdaten, Bereitstellung

der Kommunikationsmöglichkeiten für Geräte in Messgerätenetz, Geräteschutz in einem Lokalen Netz und Bereitstellung der kryptographischen Primitiven (in Kooperation mit einem Sicherheitsmodul) verantwortlich. Zu Hardwarekomponenten der Kommunikationseinheit gehören ein TSF-Modul, ein Sicherheitsmodul und ein Kommunikationsgerät (PP, 2011).

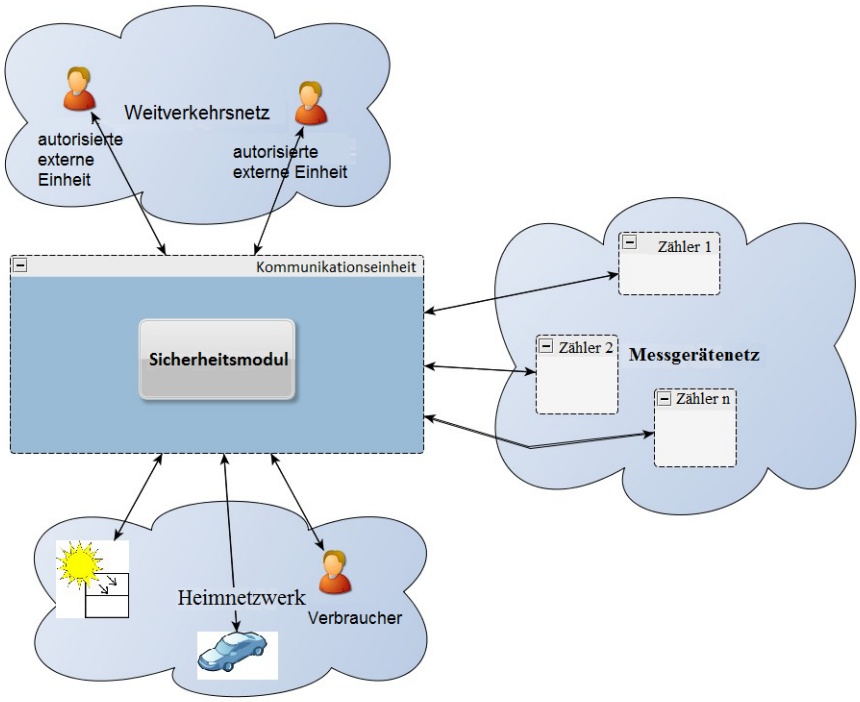

Abbildung 4: Ein intelligentes Messsystem. Übersicht (PP, 2011).

TSF ist ein Modul in Kommunikationseinheit, das für Sicherheitsfunktionen verantwortlich ist. In dem Modul werden unter anderem Verschlüsselungsmechanismen implementiert.

Ein Sicherheitsmodul wird von Kommunikationseinheit für kryptographische Unterstützung benutzt. Das Modul wird meistens in Form einer Chipkarte realisiert. Das Sicherheitsmodul ist kein Teil der Kommunikationseinheit. Die Kommunikation zwischen dem Sicherheitsmodul und der Kommunikationseinheit erfolgt über eine dafür extra vorgesehene Schnittstelle.

Das Bereich Kommunikation ist eine reine Kommunikationsdienst ohne Verschlüsselungsfunktion. Das Bereich liegt entweder in der Kommunikationseinheit oder außerhalb der Kommunikationseinheit (vgl. Abbildungen 5 und 6). Über die Kommunikationsschnittstellen werden die Daten zwischen der Kommunikationseinheit und Weitverkehrsnetz, Heimnetz-

werk und Messgerätenetz übermittelt.

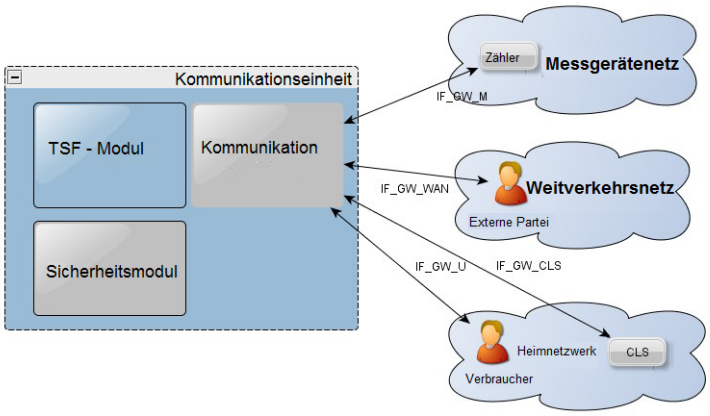

Abbildung 5: Interne Kommunikation in einem intelligenten Messsystem (PP, 2011)

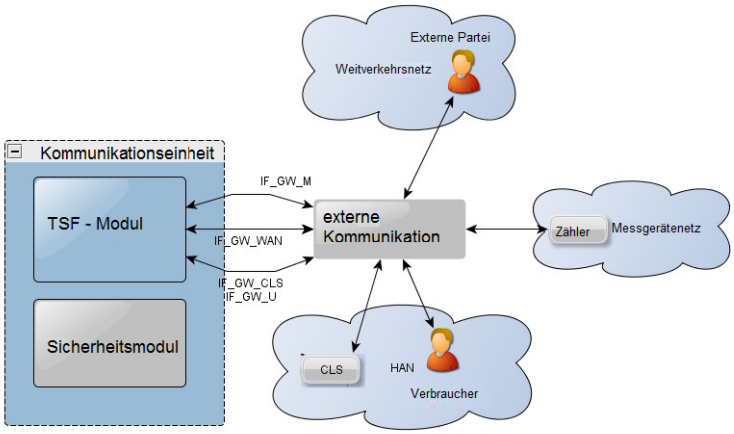

Abbildung 6: Externe Kommunikation in einem intelligenten Messsystem (PP, 2011)

Im Messgerätenetz befinden sich intelligente Zähler. Ein intelligentes Zähler verfügt über zusätzliche Funktionalitäten. Er sammelt die Messdaten, verschlüsselt die und übermittelt sie weiter an die Kommunikationseinheit. Messgerätenetz ist ein Heimnetzwerk für die lokale Datenübertragung, das für Energieverwaltung benutzt werden kann.

Weitverkehrsnetz ist ein verbreitetes Netzwerk der Datenkommunikation, das mehrere Kommunikationsgeräte im großen geographischen Gebiet verbindet. Die Kommunikationseinheit übermittelt Information an autorisierte externe Einheiten über eine Schnittstelle für

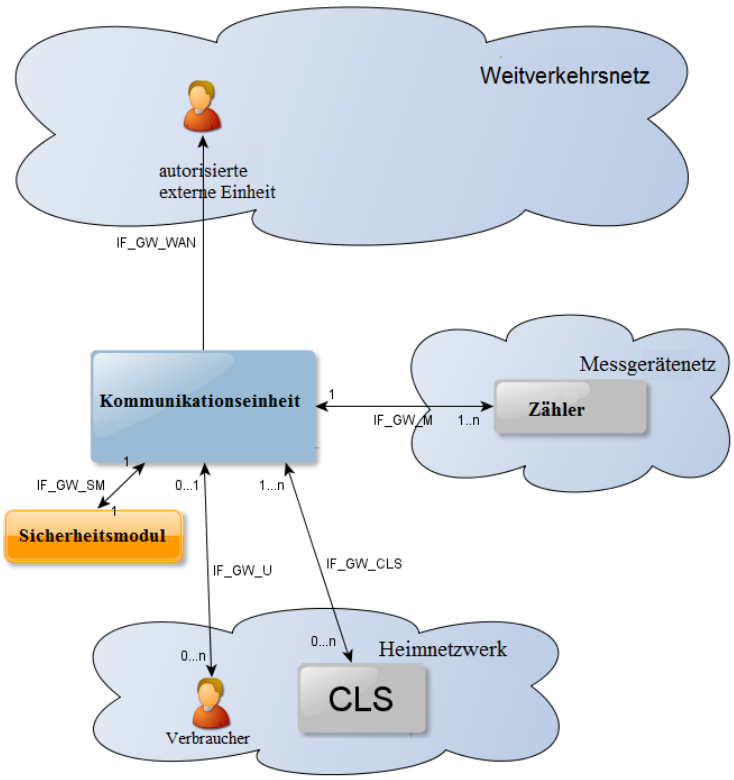

Abbildung 7: Schnittstellen des intelligenten Messsystems (PP, 2011)

die Kommunikation mit dem Weitverkehrsnetz. Zu autorisierten externen Einheiten gehören hauptsächlich Dienstanbieter und Netzbetreiber.

Heimnetzwerk ist ein lokales Netzwerk, dass Haushaltsgeräte verbindet und für Energieverwaltung benutzt werden kann. Über eine Schnittstelle zu dem Heimnetzwerk bekommen die Verbraucher einen Zugriff auf abrechnungsrelevenaten Daten. Im Heimnetzwerk befindet sich außerdem dezentral steuerbare Verbraucher- oder Erzeugersysteme (CLS). CLS sind IT Systeme im Heimnetzwerk von Verbraucher. Sie sind kein Teil des intelligenten Messsystems, können aber die Kommunikationseinheit für bestimmte Kommunikationszwecken benutzen. Zu CLS gehören unter anderem lokale Elektrizitätswerke, Klimaanlagen, intelligente Haushaltsgeräte.

Alle Schnittstellen eines intelligenten Messsystem mit entsprechenden Kardinalitäten werden in der Abbildung 7 gezeigt. Alle notwendigen Hardware Komponenten eines intelligenten Messsystem werden noch mal in der Tabelle 2 dargestellt.

Hardwarekomponente	davon notwendig
Kommunikationseinheit	1
TSF Modul	1
Kommunikationsschnittstelle	1 für WAN, MAN und HAN
Sicherheitsmodul	1
intelligenter Zähler	1 bis n
CLS	0 bis 1

Tabelle 2: Notwendige Hardwarekomponenten in einem intelligenten Messsystem

2.3 Sicherheitsfunktionen

Die logische Abgrenzung der Kommunikationseinheit wird nach BSI Schutzprofil durch folgende Sicherheitsfunktionen definiert (PP, 2011):

- Verwaltung von Messdaten
- Vertraulichkeit
- Integrität und Authentizität
- Flusskontrolle der Information
- Verwaltung von Sicherheitsfunktionen

Verwaltung von Messdaten umfasst beiderseitige Autorisierung von Kommunikationseinheiten, Geräten und Benutzern in einem intelligenten Messsystem. Diese Funktion bestimmt außerdem, welche Daten an welche Komponente oder eine externe Einheit gesendet werden müssen. Die Funktionalität soll sicher stellen, dass keine Daten an nicht-autorisierte Parteien gesendet werden oder keine Daten von nicht-autorisierten Parteien empfangen werden.

Vertraulichkeit bedeutet, dass die Kommunikationseinheit Schutz gegen Datenauskunft gewährleistet. Die Vertraulichkeit beinhaltet Datenerhalt, Datenübermittlung, Datenspeicherung und Datenlöschung. Die Kommunikationseinheit gewährleistet Verschlüsselung für Kommunikation mit den Zähler, übermittelt die gesammelte Daten an externe Einheiten und speichert die Zählerdaten, wenn die externe Einheit nicht erreichbar ist. Die Kommunikationseinheit pseudonymisiert die Daten für die Parteien, die keine Verhältnisse zwischen

den Zählerdaten und dem Verbraucher brauchen. Dann soll jede Information, die nicht mehr gebraucht wird, überschrieben werden, um sicherzustellen, dass sie nicht mehr über externe Schnittstellen der Kommunikationseinheit verfügbar ist.

Verifikation von Integrität und Authentizität bietet Schutz gegen Datenänderung von nicht-autorisierten Benutzer. Die Kommunikationseinheit verifiziert die Integrität und die Authentizität der empfangenen Daten und verifiziert Signaturen der Daten, die von dem Zähler empfangenen werden.

Flusskontrolle der Information garantiert, dass Verbindungsetablierung nur von Kommunikationseinheit und Geräten im lokalen Netz erlaubt ist. Verbindungsetablierung von externen Einheiten in Weitverkehrsnetz ist nicht erlaubt. Nur kryptographisch geschützte Verbindungen zu der konfigurierten Adressen sind erlaubt. Die Kommunikationseinheit garantiert, dass die Kommunikation mit dem Zähler innerhalb der administrativ-definierten Intervall oder innerhalb von einem Zähler-definierten Intervall etabliert wird. Die Kommunikationseinheit wiederholt Datensendung bis eine konfigurierbare Anzahl von erfolglosen Versuchen erreicht wird. Die Kommunikationseinheit stellt einen zuverlässigen Zeitstempel zur Verfügung und aktualisiert internen Zähler anhand der zuverlässigen Zeitinformation, die eine zuverlässige Quelle in Weitverkehrsnetz liefert.

Nur autorisierte Administratoren haben Rechte zu Verwaltung der Sicherheitsfunktionen. Die Kommunikationseinheit soll zusätzlich ein sicheres Mechanismus für Softwareaktualisierung implementieren, damit nur autorisierte Parteien eine Softwareaktualisierung machen dürfen.

2.4 Kryptographischer Support

Die kryptographische Funktionen gehören zu vorgeschriebenen Sicherheitsfunktionen des intelligenten Messsystems. Sie werden zwischen der Kommunikationseinheit und dem Sicherheitsmodul verteilt. Die kryptographische Funktionalität umfasst mehrere Aspekte (PP, 2011):

- Kommunikation mit den externen Einheiten
- Kommunikation mit den Verbraucher
- Kommunikation mit den Zähler
- Verifikation der empfangenen Zählerdaten
- Signieren von Daten vor dem Senden an externen Einheit

Die Kommunikationseinheit ist für Verschlüsselung und Hash-Code-Anwendung zuständig. Sicherheitsmodul übernimmt folgende Funktionen:

- Schlüsselverhandlung
- Authentifikation
- sichere Speicherung von privaten und öffentlichen Schlüsseln
- Zufallszahlengenerierung
- Erstellung und Verifikation von Signaturen

Die Funktionsverteilung zwischen der Kommunikationseinheit und dem Sicherheitsmodul ist in der Tabelle 3 dargestellt.

Aspekte	**Komm.-Einheit**	**Sicherheitsmodul**
Kommunikation mit den externen Einheit	Ver-/Entschlüsselung	Schlüsselverhandlung: - Authentifikation der externen Einheiten - Sichere Speicherung des privaten Schlüssels - Zuverlässige Speicherung des öffentlichen Schlüssels - Zufallszahlengenerierung
Kommunikation mit den Verbraucher	Ver-/Entschlüsselung	Schlüsselverhandlung: - Authentifikation des Benutzers - Sichere Speicherung des privaten Schlüssels - Zuverlässige Speicherung des öffentlichen Schlüssels - Zufallszahlengenerierung
Kommunikation mit den Zähler	Verschlüsselung	Schlüsselverhandlung: - Sichere Speicherung des

		privaten Schlüssels - Zuverlässige Speicherung des öffentlichen Schlüssels - Zufallszahlengenerierung
Verifikation der empfangenen Zählerdaten	Hash-Code-Anwendung	- Verifikation von Signaturen - Sichere Speicherung des öffentlichen Schlüssels
Signieren von Daten vor dem Senden an externe Einheit	Hash-Code-Anwendung	- Erstellung von Signaturen - Sichere Speicherung des privaten Schlüssels
Verschlüsselung der Inhaltsdaten	Ver-/Entschlüsselung	- Schlüsselgenerierung - Verschlüsselung des Schlüssels

Tabelle 3: Verteilung von kryptographischen Funktionen zwischen der Kommunikationseinheit und dem Sicherheitsmodul (PP, 2011)

2.5 Schutz gegen Angriffe

Ein intelligentes Messsystem, das entsprechend der Sicherheitsanforderungen implementiert wird, soll sicher gegen mögliche Angriffe sein. Dabei wird ein Unterschied zwischen zwei verschiedenen Angreifertypen gemacht: lokaler Angreifer und Fernangreifer (PP, 2011).

- Lokalen Angreifer haben physikalischen Zugriff auf einen intelligenten Zähler, auf eine Kommunikationseinheit und auf die Kommunikation zwischen den beiden.
- Fernangreifer versuchen die Daten über das Weitverkehrsnetz verfälschen oder abzuhören.

In einem intelligenten Stromnetz wird der Unterschied zwischen folgenden Angriffstypen gemacht:

- *Modifizierung der abrechnungsreleventen Daten*; Angreifer können versuchen die Daten, die von einem Zähler an die Kommunikationseinheit gesendet sind, zu ändern.

Das Ziel des Angriffs ist die abrechnungsreleventen Daten zu manipulieren. Um dieser Angriff zu realisieren werden wahrscheinlich auch die Änderungen im Firmware oder in den Konfigurationsparametern notwendig.

- *Modifizierung von Zeitstempel*; Ein Verbraucher oder ein lokaler Angreifer oder ein Angreifer in dem Weitverkehrsnetz können versuchen die Zeiteinstellung in der Kommunikationseinheit oder den Zeitstempel von dem Zähler in gemessenen Daten zu ändern.

- *Verletzung der Privatsphäre durch abhören von Daten, die vom Zähler an die Kommunikationseinheit übermittelt werden*; Schutz gegen diesen Angriff spielt eine wichtige Rolle, wenn mehrere Zähler für mehrere Verbraucher mit einer Kommunikationseinheit verbunden sind.

- *Verletzung der Privatsphäre durch Abhören und bekanntgeben der Daten, die an eine externe Einheit übermittelt werden*;

- *Zugriff auf gespeicherte Daten in der Kommunikationseinheit*; Physikalisch oder mit logischen Mitteln kann ein Angreifer die Daten, die nicht länger von der Kommunikationseinheit gebraucht werden, abzuhören, z. B. Zählerdaten, Zählerkonfiguration oder CLS-Konfiguration. Er kann versuchen die Information zu lesen, zu manipulieren oder zu löschen. Ein Angreifer im Weitverkehrsnetz benutzt dafür nur logische Schnittstelle, wobei ein lokaler Angreifer auch physikalisch auf die Kommunikationseinheit zugreifen kann.

- *Kontrolle über Sicherheitsmechanismen und das Beschädigen vom Netz und Netz Komponente*; Angreifer versuchen die Kontrolle über die Kommunikationseinheit, Zähler oder CLS zu bekommen. Sie können falsche Daten an externe Einheit schicken und dadurch Verbraucher, Netz oder eine externe Einheit beschädigen.

- *Verletzung von Privatsphäre durch Information über Benutzer*; Das beinhaltet verschiedenen Szenarios. Zuerst soll das System sicher stellen, das autorisierte externe Einheit keine zusätzliche Information, die den Schutz der Privatsphäre verletzt, bekommen kann. Für Angreifer muss es unmöglich sein, jene Information über Verbraucher zu bekommen. (PP, 2011)

Schutz gegen Angriffe ist ein sehr wichtiger Aspekt, der unbedingt bei einer Modellierung und dem Einführen des Systems, überprüft werden muss. Zumindest alle bekannte Angriff-Szenarios sollen analysiert werden. Das stellt eine große separate Forschungsherausforderung dar.

2.6 Schutz der Privatsphäre

In BSI Schutzprofil ist der Schutz der Privatsphäre als eine wesentliche Anforderung an die intelligente Messsysteme formuliert (PP, 2011).

Zuerst werden zwei Hauptaspekte betrachtet:

- Zugriffskontrolle
- Transparenz für die Kunden

Zugriffskontrolle bedeutet, dass nur ein Minimum an den Informationen über den Verbraucher an autorisierte externe Einheiten übermittelt wird. Um das sicher zu stellen, werden Mechanismen, wie Verschlüsselung und Pseudonymität benutzt. Verschlüsselung garantiert, dass nur autorisierte Parteien die Informationen lesen können. Pseudonymität gewährleistet, dass nicht-abrechnungsrelevante Information ohne direkte Verbindung zu Benutzeridentität gesendet wird. Vollständige Anonymisierung ist allerdings nach dem Schutzprofil trotz aller Vorteilen für Verbraucher kaum möglich. Mit vollständiger Anonymisierung können die externe Einheiten nicht überprüfen, ob die Daten von einem vertrauenswürdigen Quelle kommen (PP, 2011).

Transparenz für die Kunden bedeutet, dass sie jederzeit in der Lage sind seine eigene Daten abzurufen. Das wird durch eine *Log*-Datei gewährleistet. Zugriff auf die Datei wird nur über einen Heimnetzwerk für autorisierte Benutzer möglich sein. Der Zugriff auf die Daten der anderen Kunden ist nicht erlaubt.

Weiterhin sollen externe Einheiten selbst vertrauenswürdig sein. Sie dürfen keine unautorisierte Analyse der Information durchführen, die die Privatsphäre der Kunden in irgendwelcher Form beschädigt.

Kommunikationseinheit soll letztendlich auch garantieren, dass die Kommunikation in Weitverkehrsnetz für die externe unautorisierte Parteien verborgen bleibt. Erhalten der Information, die für den Schutz der Privatsphäre wichtig ist, durch die Analyse der Frequenz, der Last oder des Kommunikationsausfalls muss ebenfalls nicht möglich sein (PP, 2011).

3 Lösungsansätze für den Schutz der Privatsphäre in einem intelligenten Messsystem

Forschung im Bereich der intelligenten Messsysteme umfasst verschiedene Themen: Technologie, Wirtschaftsaspekte, Energieversorgung, Wartung, Kommunikation usw. Kaum überraschend, dass der Schutz der Privatsphäre in mehreren wissenschaftlichen Arbeiten der letzten Jahren im Mittelpunkt gestellt wird. Ohne Lösung dieses Problems kann das intelligentes Messsystem gar nicht rechtlich im Betrieb genommen werden, egal wie gut die andere Anforderungen implementiert sind. In diesem Kapitel werden die Hauptlösungsansätze unter die Lupe genommen und die Reihe nach vorgestellt. In dem Kapitel werden folgende Lösungen dargestellt:

- eine Lösung mit dem Einsatz von vertrauenswürdigen dritten Partei (Bohli u. a., 2010)
- eine Lösung, die sich auf statistischen Methoden basiert (Bohli u. a., 2010)
- eine Lösung mit Berechnung von Teilwerten und Aggregation von Teilwerten (Finster und Conrad, 2010)
- eine Lösung mit einer Zertifizierungsstelle, Teilwerten und homomorpher Verschlüsselung (Garcia und Jacobs, 2010)
- eine Lösung mit dem aufspannenden Baum und homomorpher Verschlüsselung (Li u. a., 2010)
- eine Lösung mit dem Schlüssel-Aggregator und Ring der Zählern (Mármol u. a., 2011)

3.1 Die Lösungsansätze: mit und ohne vertrauenswürdigen dritten Partei

Die ersten zwei Lösungsansätze stellen folgende Sicherheitsanforderungen an einen intelligenten Messsystems (Bohli u. a., 2010):

- Der Energieversorger soll die Summe der aktuellen Werte für alle Verbraucher wissen.
- Der Energieversorger soll die Verbrauchswerte der individuellen Haushaltsbesitzer für eine bestimmte Periode (z.B. ein Monat oder ein Jahr) wissen.
- Der Energieversorger soll aber keine aktuellen Werte der individuellen Haushaltsbesitzer wissen.
- Der Energieversorger soll in der Lage sein einen aktuellen Wert für alle Verbraucher zu berechnen, wenn einige Werte verloren gehen.

Diese Annahmen entsprechen vollständig der Anforderungen, die in dem Schutzprofil formuliert sind (PP, 2011). Weiterhin wird bei der Lösungen angenommen, dass die intelligenten Stromzähler vertraulich sind und alle Berechnungen entsprechend dem Protokoll ausführen können.

3.1.1 Eine Lösung mit einer vertrauenswürdigen dritten Partei

Eine vertrauenswürdige dritte Partei ist eine *Aggregation-Proxy* (Bohli u. a., 2010). Intelligente Stromzähler senden seine verschlüsselte Daten an einen *Proxy*. Das *Proxy* summiert die Daten und übermittelt sie weiter an Energieversorger. Um die Daten für einzelne Verbraucher über eine Abrechnungsperiode zu bestimmen, werden die Daten einzelner Zähler für diese Periode auch summiert und weitergeleitet.

Die Schwäche liegt darin, das eine vertrauenswürdige dritte Partei alle Verbraucherdaten hat. Das kann offensichtlich, unter Umständen, eine Bedrohung für den Schutz der Privatsphäre darstellen (Bohli u. a., 2010). Die Lösung ist sicher unter der Annahme, dass die dritte Partei vertrauenswürdig ist und die Energieversorger die Kommunikation zwischen den Zähler und der vertrauenswürdigen dritten Partei nicht abhören kann. Wenn die Energieversorger die Kommunikation zwischen dem *Proxy* und den Zähler abhören kann, ist die Sicherheit der Lösung von dem Verschlüsselungsstärke abhängig. Perfekter Schutz ist in diesem Fall nur mit *one-time pad* möglich. Relative Sicherheit wird auch mit solchen modernen Verschlüsselungschema, wie z. B. *AES* erreicht.

Bei einer Lösung mit einer vertrauenswürdigen dritten Partei kommt ein weiterer Angriff in Frage. Angenommen eine vertrauenswürdige dritte Partei liefert an Energieversorger Verbraucherdaten von Gruppen, die aus 50 Zähler bestehen. Ein Energieversorger kann versuchen eine Gruppe zu bilden, die aus 49 gefälschten und einem echten Zähler besteht. Der Energieversorger kann dann den aktuellen Verbrauch der echten Zähler berechnen. Dafür werden von der Summe aller Gruppenwerten die Werte von allen 49 gefälschten Zähler subtrahiert. Die Einschränkung besteht darin, dass man n gefälschte Gruppen braucht, um die laufende Daten von n Zähler zu bekommen. Die dritte Partei kann die gefälschten Gruppen relativ einfach erkennen, wenn ungefähre Anzahl der Verbraucher für die vertrauenswürdige dritte Partei bekannt ist (Bohli u. a., 2010).

3.1.2 Eine Lösung ohne vertrauenswürdigen dritten Partei

Die Lösung ohne vertrauenswürdigen dritten Partei basiert auf Wahrscheinlichkeitstheorie (Bohli u. a., 2010). Beim diesem Ansatz kommunizieren intelligente Zähler direkt mit dem Energieversorger. Die Zähler addieren aktuelle Zählerstände $e_{i,j}$ mit einem Zufallswert $r_{i,j}$. Der Wert $r_{i,j}$ ergibt sich aus der bekannten Verteilung mit bekannter Varianz σ^2 und einem

Erwartungswert μ. Die Summe $e_{i,j} + r_{i,j}$ wird an dem Energieversorger übermittelt.

Der Wert $r_{i,j}$ ist ein kontinuierlicher Zufallswert, der alle mögliche Realwerte, positiv als auch negativ, annehmen kann. Die Summe aller diesen Zufallswerten $r_i + r_{...} + r_j$ bildet eine Zufallsverteilung mit einem Erwartungswert 0. Daraus folgt, dass die Summe von aller Werten bei dem Energieversorger unverändert bleibt:

$$\sum_{i=1}^{c} e_{i,j} + r_{i,j} = e_{i,j} + 0 = e_{i,j} \tag{1}$$

Durch statistische Distribution ist es aber nicht mehr möglich aus der gesamten Summe $e_{i,j} + r_{i,j}$ oder aus der Summe einzelner Werten und Zufallswerten $e_i + r_i, \ldots, e_j + r_j$ die einzelne Verbrauchswerte abzuleiten. Damit wird in diesem Ansatz der geforderte Schutz der Privatsphäre erreicht. Die einzelne Schwäche dieses Ansatzes liegt darin, dass man relativ große Gruppen mit Millionen einzelner Verbraucher braucht, damit eine Zufallsverteilung mit einem Erwartungswert 0 gebildet werden kann (Bohli u. a., 2010). Das ist die Haupteinschränkung für den praktischen Ansatz dieser Theorie. Weitere Einschränkungen müssen in Vergleich mit der Anforderungen von BSI Schutzprofil analysiert werden.

3.2 Eine Lösung mit Aufteilung und Aggregation

Der Algorithmus mit Aufteilung und Aggregation besteht aus drei Phasen: Initialisierung, Anonymisierung und Aggreration. Die Kommunikationstopologie wird in der Abbildung 8 dargestellt (Finster und Conrad, 2010).

In der Initialisierungsphase sendet der Datenkonzentratoran alle Zähler sowohl eine Liste mit allen vorhandenen Zählern, als auch die Werte n und t. Der Wert n ist die Anzahl der Teilwerten und der Wert t ist die Länge der Anonymisierungsphase in Sekunden. Jeder Zähler erstellt eine temporäre Liste mit $n-1$ beliebigen ausgewählten Zählern. Zusätzlich wird das Register R_i mit dem zuletzt gemessenen Wert initialisiert. Außerdem braucht jeder Zähler einen Singnaturschlüssel $k_{i,j}$, der im Zähler während dem ganzen Messzyklus gespeichert wird.

In der Anonymisierungsphase berechnet jeder Zähler $n-1$ zufällige Teilwerte $PV_{i,j,l}$ und addiert sie zu R_i:

$$R_i = R_i + PV_{i,j,l} \tag{2}$$

Zu jedem Teilewert wird eine HMAC - Signatur wie folgt berechnet:

$$S_{k_{i,j}}(PV_{i,j,l}) = H((k_{i,j} \oplus opad) \parallel H((k_{i,j} \oplus ipad) \parallel PV_{i,j,l})) \tag{3}$$

Jeder HMAC-signierte Teilwert wird an einen Zähler aus der Liste übermittelt. Am Ende des Übermittlungszyklus bekommt jeder Zähler durchschnittlich $n-1$ Teilwerte von den anderen

Zählern.

In der Aggregationsphase wird der aktuellen Abrechnungswert $BV_{i,j}$ bestimmt. Um den eigenen Teilwert $PV_{i,j,0}$ zu berechnen, wird von dem Abrechnungswert der Wert des eigenen Registers R_i subtrahiert. Der eigenen Teilwert wird auch mit HMAC-Signatur signiert:

$$PV_{i,j,0} = BV_{i,j} - R_i \tag{4}$$

$$S_{k_{i,j}}(PV_{i,j,0}) = H((k_{i,j} \oplus opad) \parallel H((k_{i,j} \oplus ipad) \parallel PV_{i,j,0})) \tag{5}$$

Jeder Zähler berechnet außerdem ein Chiffrat $C_{i,j}$ aus dem aktuellen Abrechnungswert und geheimen Schlüssel $k_{j,i}$, das mit einem öffentlichen Schlüssel $k_{i,pub}$ verschlüsselt wird:

$$C_{i,j} = E_{k_{i,pub}}(BV_{j,i}, k_{j,i}) \tag{6}$$

Dabei wird sichergestellt, das nur der Zähler Z_i das Chiffrat entschlüsseln kann , weil es mit einem öffentlichen Schlüssel verschlüsselt wird (Finster und Conrad, 2010).

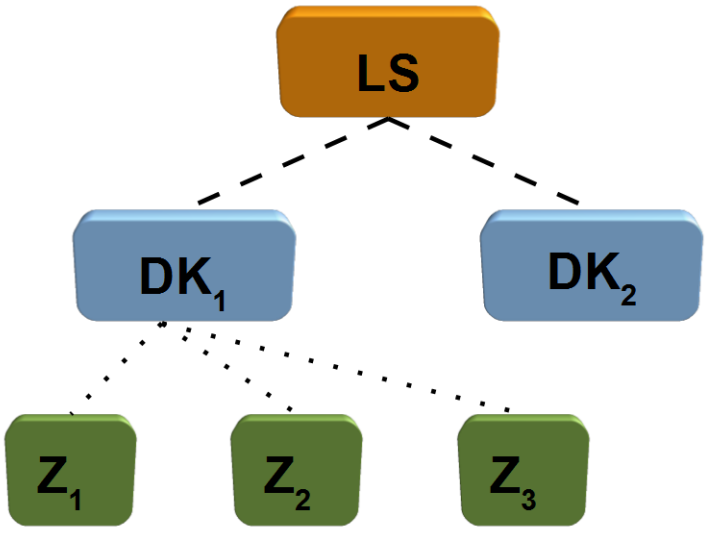

Abbildung 8: Aufbau der Kommunikationstopologie (Finster und Conrad, 2010)

Jeder Zähler Z_i übermittelt an Datenkonzentrator folgenden Daten:

- Chiffrat $C_{i,j}$
- n Teilwerte $PV_{j,x,y}$ mit HMAC-Signatur

Der Datenkonzentrator berechnet aus Teilwerten, die er von den Zählern bekommt, eine Summe von Abrechnungswerten T_j, wobei n die Anzahl der Teilwerte pro Zähler ist und m die Anzahl der Zähler des Datenkonzentrator ergibt (Finster und Conrad, 2010):

$$T_j = \sum_{i=0,l=0}^{i<n,l<m} PV_{j,i,l} \text{ mit } 0 \leq i < n,\ 0 \leq l < m \tag{7}$$

Der Datenkonzentrator DK übermittelt die berechnete Summe weiter an Leitstelle LS. Dabei ist der Kommunikationsaufwand bei der Lösung sehr hoch. Zusätzlich wird eine Liste mit N Einträge, $n-1$ Werte und MAC Werte für $n-1$ Zähler und n Werte und Verschlüsselung für den Datenkonzentrator benötigt.

3.3 Asymmetrische homomorphe Verschlüsselung mit einer Zertifizierungsstelle

In dem Ansatz von Garcia und Jakobs wird eine auf so-genanntem “no-leakage” Protokoll basierende Lösung formuliert. (Garcia und Jacobs, 2010). Nach sogenanntem “no-leakage” Protokoll sind mehrere Zähler $M_1 \ldots M_N$ mit einem Datenkonzentrator SSt verbunden. Der Datenkonzentrator verfügt über einen Gesamtwert des Stroms, der geliefert wurde. Er bekommt ebenfalls von den Zähler die verbrauchte Werte und summiert sie. Am Ende des Protokolls vergleicht der Datenkonzentrator die gelieferten und verbrauchten Mengen. Die Verbindung von Zählern und dem Datenkonzentrator wird in der Abbildung 9 angezeigt.

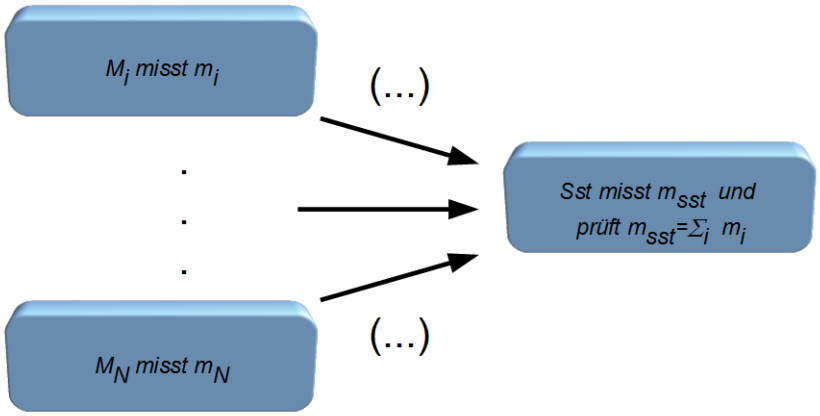

Abbildung 9: Zähler und Datenkonzentrator in “no-leakage” Protokoll (Garcia und Jacobs, 2010)

Der Datenkonzentrator initialisiert das Protokoll. Er sendet Publik-Key-Zertifikate an jeden Zähler M_1 bis M_N, wobei N immer großer 1 sein muss (Garcia und Jacobs, 2010):

$$SSt \rightarrow M_i : no-leakage; \quad cert_{M_i}, \ldots, M_N \tag{8}$$

Jeder Zähler misst sein aktuelles Verbrauch m_i und zerlegt ihm in N Teilewerte. Die Teilwerte sind Zufallszahlen $a_{i1}, \ldots a_{1N}$. Die werden so gewählt, dass $m_i = \sum_j a_{ij}$ mod n. Der Zähler M_i verschlüsselt jeden Teilwert, außer a_{ii} mit dem öffentlichen Schlüssel pk_j und sendet die Werte zurück an den Datenkonzentrator SSt. Der Teilwert a_{ii} bleibt bei dem Zähler M_i intern gespeichert.

Der Datenkonzentrator multipliziert alle verschlüsselte Teilwerte N-1, die er von dem Zähler M_i bekommt. Das Ergebnis ist wegen Homomorphismus gleich der Summe der Teilwerten N-1. Dann schickt der Datenkonzentrator die Teilwerte zurück an die Zähler für Entschlüsselung. Der Zähler entschlüsselt die Teilwerte N-1 und addiert zu ihnen Teilwert a_{ii}. Schließlich bekommt der Datenkonzentrator die Werte von allen Zählern und addiert sie zusammen. Das Ergebnis wird am Ende mit dem gelieferten Wert verglichen:

$$M_i \rightarrow SSt : \sum_{j \neq i} a_{ji} + a_{ii} = \sum_j a_{ji} \text{ mod n; } SSt \text{ sets } m := \sum_i \sum_j a_{ji} \text{ mod n} \tag{9}$$

(Garcia und Jacobs, 2010). Die Summe von Datenkonzentrator wird dann weiter an Netzbetreiber übermittelt. Bei dieser Lösung ist keine direkte Kommunikation zwischen den intelligenten Zählern notwendig. Der Kommunikationsaufwand bleibt jedoch hoch, wegen Zertifikaten und Teilwerte.

3.4 Eine Lösung mit einem aufspannenden Baum

Bei der Lösung mit einem aufspannenden Baum bauen N intelligenten Zähler einen Graph mit $N+1$ Knoten und N Kanten. Jeder Knoten sammelt rekursiv die Werte von seinen Kinderknoten, addiert sein eigenen Wert und leitet das Ergebnis weiter an Elternknoten (Li u. a., 2010).

Bei der Konstruktion der aufspannenden Baum soll unbedingt berücksichtigt werden, dass die Höhe des Baums relativ klein bleibt. Dadurch wird die Anzahl des *Hops* reduziert. Der Baum soll auch nicht zu viel Kinder haben. Sonst führt dass zu den aufwendigen Berechnungen. Um ein optimalen Baum zu bekommen, wird er durch Breiten- und Tiefensuche gebildet. Dabei wird der Datenkollektor als Startknoten gewählt. Wenn der Knoten zu viel Kinder hat, muss der Baum entsprechend neue ausbalanciert werden.

Nach der Initialisierung bekommt jeder Knoten einen Tupel mit dem Aggregationsplan:
{ T_{id} - Nachrichten Id,
Trigger - startet Aggregation,
Data - übertragende Daten,
Collect - eine Anweisung, um die Daten für bestimmten Knoten zu sammeln,
Operation - Verschlüsselung und Aggregation,
Destination - Elternknoten,
Key - Schlüssel } (Li u. a., 2010).

Das Protokoll mit dem aufspannenden Baum besteht aus folgenden Schritten:

- Jeder Zähler bekommt ein Aggregationsplan und entscheidet, ob er für den Trigger warten muss oder gleich starten kann
- Der Zähler verschlüsselt die gemessenen Daten mit dem Schlüssel
- Der Zähler sammelt die Daten von Kinderknoten und ausführt die Verschlüsselung und Aggregation von aller gesammelten Daten, inklusive eigene Daten
- Schließlich, sendet der Zähler die verschlüsselte Daten an die Elternknoten. Die gesendete Nachricht besteht aus einem Tupel {T_{ID}, TS, Data} (Li u. a., 2010). Wobei TS ist einen Zeitstempel, der für Synchronisation benutzt wird.

Der Algorithmus mit dem aufspannendem Baum hat kleinen Kommunikationsaufwand wegen baumförmiger Aggregation. Die Hauptschwierigkeit für die Implementierung liegt auch im Baum selbst. Erstellen eines Baums und deren mögliche Ausbalancieren sind zwei große Probleme. Weiterhin soll eine Lösung gefunden werden für den Fall, wenn einer oder mehrere Knoten die Verbindung mit dem Elternknoten vermissen. Für die Knoten, die Kinderknoten haben, bedeutet das, dass die Verbindung mit den Kinderknoten auch vermisst wird. In dem Lösungsansatz wird angenommen, dass der Graph stabil für eine bestimmte Zeit bliebt (Li u. a., 2010). Das kann aber unter realen Bedienungen auch anderes sein.

3.5 Eine Lösung mit dem Schlüssel-Aggregator und dem Ring der Zählern

In dieser Lösung werden zwei Fälle unterschieden: Normalfall und ein Fall mit einem bösartigen Zähler. Die zweite Schema kommt in Frage, wenn einen bösartigen Zähler erkannt wird. Sie funktioniert mit dem Umschaltungsmechanismus. Die Gruppierung und Ring-Architektur der Zählern bleibt für beide Teilansätze gleich.

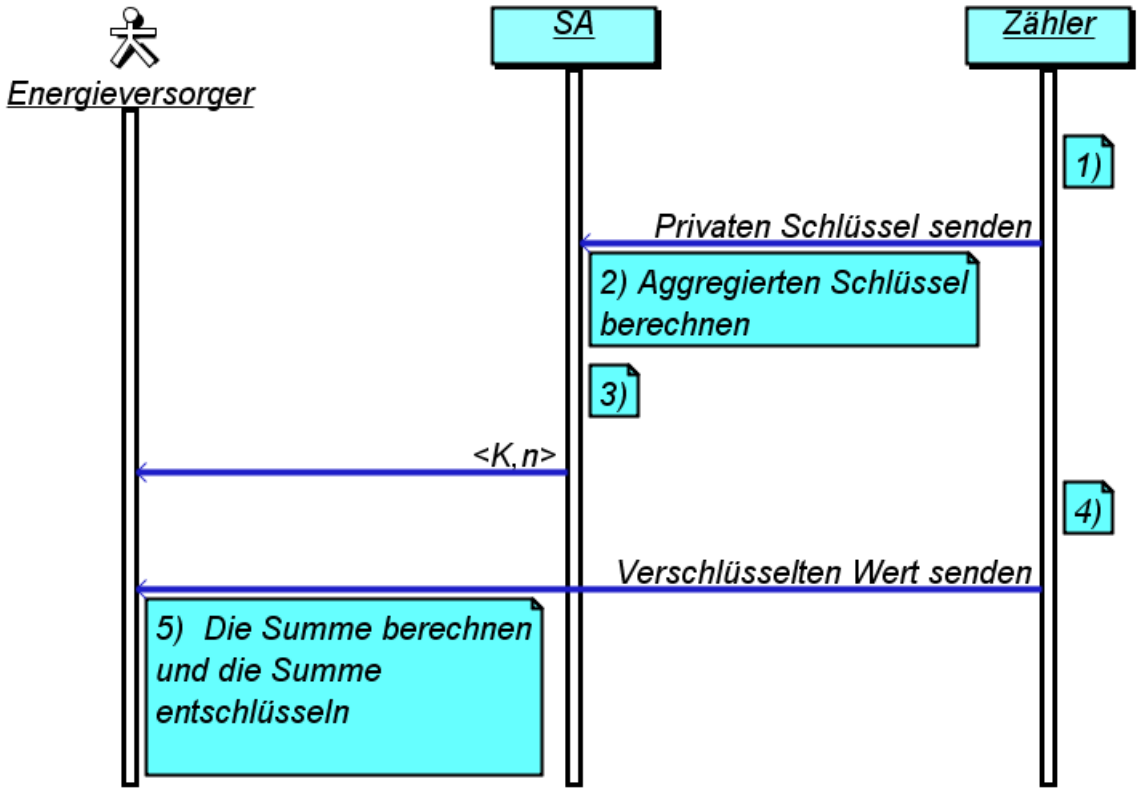

Abbildung 10: Ablauf des Algorithmus im Normalfall (Mármol u. a., 2011)

3.5.1 Normalfall

Die Schritte des Algorithmus werden in der Abbildung 10 dargestellt:.

1. Jeder Zähler sendet sein Schlüssel k_{ij} über einen sicheren Kanal an einen Schlüssel-Aggregator. Das passiert jeder Berichtsperiode j.

2. Der Schlüssel-Aggregator berechnet den Schlüssel $K = \sum_{i=1}^{n} k_{ij}$

3. Der Schlüssel-Aggregator schickt den Schlüssel K an den Energieversorger zusammen mit der Anzahl der angekommenen Schlüsseln n.

4. Jeder Zähler sendet seine gemessene Daten verschlüsselt mit dem Schlüssel k_{ij}. Die verschlüsselte Daten $E_{K_{ij}}(e_{ij})$. werden an Energieversorger nach einer bestimmten Periode j über einem sicheren Kanal übermittelt. Dabei wird die Identität der einzelnen Zählen verborgen.

5. Schließlich, überprüft der Energieversorger, ob die Anzahl der angekommenen verschlüsselten Nachrichten $E_{K_{ij}(e_{ij})}$ mit der Anzahl der Schlüsseln n übereinstimmt. Danach berechnet der Energieversorger die Summe aus der verschlüsselten Nachrichten und entschlüsselt diese Summe. Das Ergebnis ist die Summe einzelner Werte, die von den Zählern gemessen wurde:

$$D_k\Big(\sum_{i=1}^{n} E_{k_{ij}}(e_{ij})\Big) = \sum_{i=1}^{n} e_{ij} \tag{10}$$

(Mármol u. a., 2011)

3.5.2 Ein Fall mit bösartigen Zählern

In diesem Fall wird ein Szenario betrachtet, bei dem Zähler bösartig agieren können. Der Ablauf besteht aus folgenden Schritten (Mármol u. a., 2011):

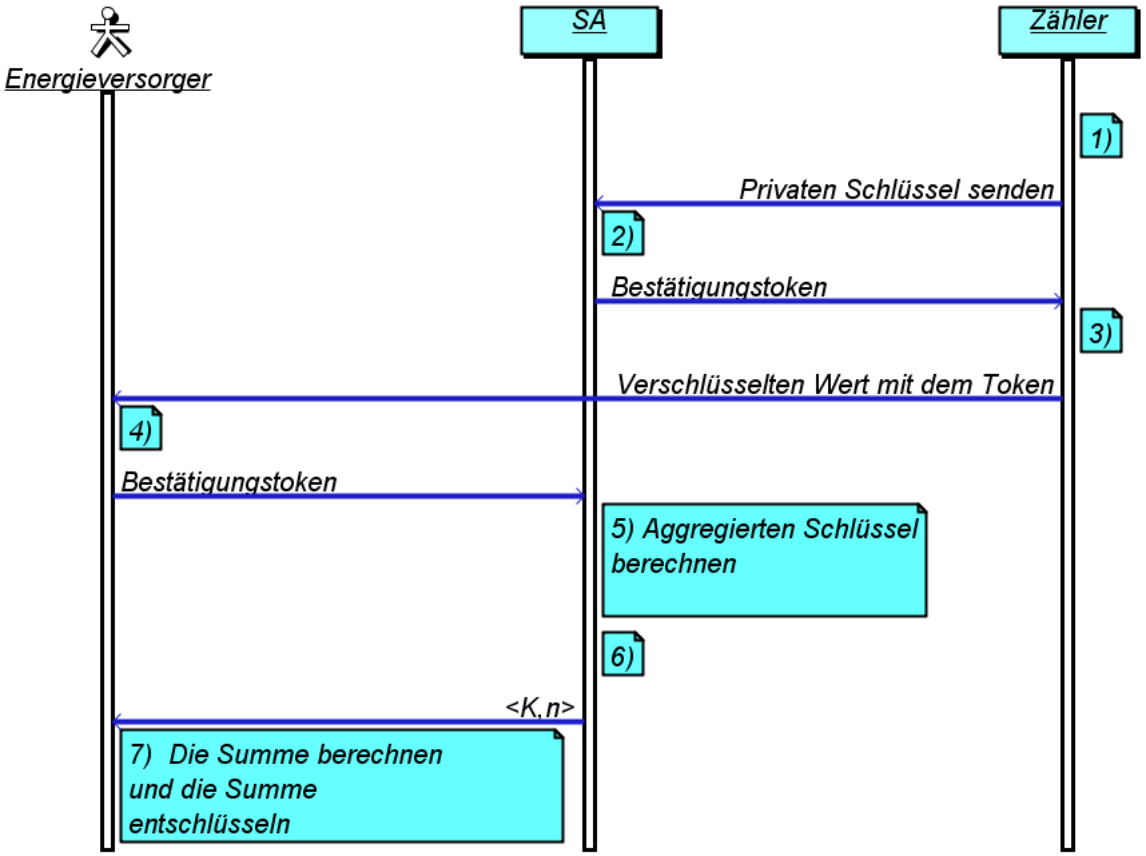

Abbildung 11: Ablauf des Algorithmus im Fall mit bösartigen Zählern (Mármol u. a., 2011)

1. Jeder Zähler sendet sein Schlüssel k_{ij} über einen sicheren Kanal an einen Schlüssel-Aggregator KA, wie in dem normalen Fall.

2. Der Schlüssel-Aggregator antwortet mit einem ACK Token $T_{KA,i}$ zu jedem Zähler.

3. Jeder Zähler sendet die verschlüsselte Daten $E_{K_{ij}}(e_{ij})$ zusammen mit dem ACK Token $T_{KA,i}$ an den Energieversorger.

4. Der Energieversorger akzeptiert verschlüsselte Nachrichten von den Zählern, nur dann, wenn sie zusammen mit einem Token kommen. Der Energieversorger antwortet mit einem ACK Token $T_{ES,i}$ an den Schlüssel-Aggregator.

5. Wenn der Schlüssel-Aggregator den Token $T_{ES,i}$ bekommt, berechnet er den aggregierten Schlüssel $K = \sum_{i=1}^{n} k_{ij}$. Wenn der Aggregator kein Token $T_{ES,i}$ bekommt, der Schlüssel k_{ij} wird weggeworfen und nicht aggregiert.

6. Der Schlüssel-Aggregator schickt den Schlüssel K zusammen mit der Anzahl der angekommen Schlüsseln n an den Energieversorger .

7. Der Energieversorger berechnet die Summe aus der verschlüsselten Nachrichten und entschlüsselt diese Summe. Das Ergebnis ist die Summe einzelner Werte, die von den Zählern gemessen worden sind, wie in dem Normalfall:

$$D_k\Big(\sum_{i=1}^{n} E_{k_{ij}}(e_{ij})\Big) = \sum_{i=1}^{n} e_{ij} \tag{11}$$

   (Mármol u. a., 2011)

Die Schritte werden in der Abbildung 11 dargestellt.

Bei beiden letzten Ansätzen gehören die Zähler zu einer Gruppe und bilden einen Ring. Das Problem von eintreten des Ringes sowie dynamische Gruppenbildung steht außerhalb dieser Analyse und muss zusätzlich untersucht werden. Das Umschaltungsmechanismus zwischen zwei Fällen ist zwar eingedeutet, aber nicht vollständig beschrieben. Für die mögliche Implementierung soll der Aufwand ebenfalls untersucht werden.

4 Theoretische Lösungsansätze und BSI Anforderungen

In vorigen Kapitel sind die Algorithmen, die den Schutz der Privatsphäre in einem intelligenten Netz ermöglichen, schematisch dargestellt. Nun wird untersucht, ob und wie weit die Algorithmen den BSI Anforderungen entsprechen. Daraus lassen sich die Lösungen herausfinden, die für die Implementierung am besten geeignet sind.

Verglichen werden:

- Hardwarekomponenten und Schnittstellen
- Sicherheitsfunktionen
- Kryptographischer Support
- Schutz gegen Angriffe
- Schutz der Privatsphäre

4.1 Hardwarekomponenten und Schnittstellen

Die Zuordnung der Teile der Algorithmen zu den Komponenten die von BSI gefordert sind, ist in der Tabelle 4 dargestellt. Wie man sieht, die Kommunikationseinheit lässt sich fast in allen Lösungsansätzen definieren. Die Sicherheitsmodule und CLS sind in den Algorithmen nicht explizit beschrieben. Es wird angenommen, dass sie als Standartkomponenten Integrierbar sind. Genaue Aufwand für jede einzelne Lösung fordert allerdings tiefere Analyse sowie Modellierung. Die Sicherheitsmodule müssen verschiedene Funktionalitäten beinhalten, die von dem Verschlüsselungsverfahren abhängig sind.

4.2 Sicherheitsfunktionen

Wie man sieht, ist die Art der Datenverarbeitung von dem verwendeten Algorithmus abhängig. Die Entschlüsselung, Verschlüsselung und die Aggregation sind in allen Lösungsansätzen nachvollziehbar. Die Datenverteilung und Signieren beinhalten nur einige Lösungsansätze. Bei einigen Lösungen wird symmetrische bei den anderen asymmetrische Verfahren vorgeschlagen (vgl.Tabelle 5).

Folgenden Verschlüsselungsalgorithmen werden genannt:

BSI Anforderung	Bohli u.a. (2010) Lösung 1	Bohli u.a (2010) Lösung 2	Finster und Conrad (2010)	Garcia und Jakobs (2010)	Li u.a. (2010)	Marmol u.a. (2011)
Hardwarekomponenten						
Kommunikation-einheit	vertrauens-würdige dritte Partei	-	Daten-Konzentrator	Daten-Konzentrator	Eltern-Knoten	Schlüssel - Aggregator
Sicherheits - modul (SM)	SM für symmet. Verschlüssel. notwendig	SM für symmet. Verschlüssel. notwendig	SM für asymmet. Verschlüssel. notwendig	SM für asymmet. Verschlüssel. notwendig	SM für asymmet. Verschlüssel. notwendig	SM für symmet. Verschlüssel. notwendig
Heimnetzwerk	+	+	+	+	+	+
Messgerätenetz	Zähler	Zähler	Zähler	Zähler	Zähler (Baum)	Zähler (Ring)
CLS	integrierbar	integrierbar	integrierbar	integrierbar	integrierbar	integrierbar
Weitverkehrsnetz	Energie-versorger	Energie-versorger	Energie-versorger	Energie-versorger	Energie-versorger	Energie-versorger
Schnittstellen						
Kommunikation-Einheit - Zähler	+	-	+	+	+	+
Kommunikation-Einheit - Benutzer	+	-	+	+	+	+
Kommunikation-Einheit - CLS	noch nicht definiert	-	noch nicht definiert	noch nicht definiert	noch nicht definiert	noch nicht definiert
Kommunikation-Einheit - SM	noch nicht definiert	-	noch nicht definiert	noch nicht definiert	noch nicht definiert	noch nicht definiert
Kommunikation-Einheit - externe Einheit	+	-	+	+	+	+

Tabelle 4: Vergleich der Hardwarekomponenten und Schnittstellen mit BSI Anforderungen

- homomorphe symmetrische Verschlüsselung nach Castelluccia (Mármol u. a., 2011)
- asymmetrische Verschlüsselung mit elliptischen Kurven (Finster und Conrad, 2010)
- asymmetrische homomorphe Verschlüsselung nach Paillier (Li u. a., 2010; Garcia und Jacobs, 2010)

In drei der bewerteten Ansätzen wird die homomorphe Verschlüsselung vorgeschlagen. Da nicht bei allen betrachteten Lösungsansätzen das Verschlüsselungsverfahren ausreichend beschrieben ist, ist die Implementierung derzeit sehr erschwert. Die Verschlüsselungsalgorithmen müssen an der Stelle konkretisiert werden.

Obwohl die Anwendung von Signaturen in den meisten Lösungen nicht erwähnt wird, lässt sie sich bei einigen Ansätzen indirekt vermuten. Bei der Lösung mit der Aufteilung und Aggregation werden HMAC-Signaturen verwendet und ihre Anwendung dementsprechend sehr detailliert im Algorithmus beschrieben. (Finster und Conrad, 2010). In der Lösung mit homomorpher Verschlüsselung nach Castelluccia (Mármol u. a., 2011) werden Gruppensignaturen nach Ateniese vorgeschlagen (Ateniese u. a., 2000). Genaue Erweiterung des Algorithmus soll in den weiteren Forschungsarbeiten gemacht werden. Alle Autoren definieren die Zeitintervalle für Messdaten. Der exakte Zeitwert (Dauer eines Abrechnungszyklus) ist auch in den BSI Anforderungen noch nicht festgelegt. Die Pseudonymität ist ein Teil der Datenverwaltung in dem BSI Schutzprofil (PP, 2011). Das ist die wichtigste Bestandteil eines anderen Aspektes - des Schutzes der Privatsphäre. Dieses Aspekt wird separat beschrieben. Wie der Vergleich zeigt, wird die Pseudonymität durch vorgeschlagenen Algorithmen für den Schutz der Privatsphäre gewährleistet.

Vertraulichkeit wird nach BSI Schutzprofil als Schutz gegen Datenauskunft sowohl in Messgerätenetz, als auch in Weitverkehrsnetz verstanden (PP, 2011). Dazu gehören auch sichere Datenspeicherung und Löschen von Daten. Bei allen Lösungen wird angenommen, dass die Vertraulichkeit bei den vorgeschlagen Algorithmen vorhanden ist. Dazu muss zusätzlich untersucht werden, ob das den BSI Anforderungen für den Schutz gegen Angriffe entspricht. Die Grundsätze für die Datenhaltung sind kaum beschrieben.

Der Schutz der Integrität und Authentizität ist einen weiteren sehr wichtiger Aspekt. Die genaue Beschreibung, wie die Autorisierung realisiert werden soll, steht allerdings zur Zeit bei meisten Autoren außerhalb der Lösungsbeschreibung. In der Lösung mit dem Ring der Zähler wird das Algorithmus von Camenisch und Losaynskaya oder Ateniese vorgeschlagen (Mármol u. a., 2011). Die andere Lösungen bieten gar kein Autorisierungsverfahren an. Für prototypische Realisierung kann man erst davon ausgehen, das die Parteien vertrauenswürdig sind. Bei der eigentlichen Implementierung soll das Autorisierungsmechanismus genau definiert werden.

BSI Anforderung	Bohli u.a. (2010) Lösung 1	Bohli u.a (2010) Lösung 2	Finster und Conrad (2010)	Garcia und Jakobs (2010)	Li u.a. (2010)	Marmol u.a. (2011)
Sicherheitsfunktionen						
Art der Daten-verarbeitung	Verschlüssel., Aggregation, Entschlüssel.	Verteilung, Verschlüssel. Aggregation (EV), Entschlüssel.	Aufteilung, Verschlüssel. Signieren Aggregation Entschlüssel.	Verschlüssel., Aggregation, Entschlüssel.	Verschlüssel., stufigartige Aggregation, Entschlüssel.	Verschlüssel., Aggregation, Entschlüssel.
Verschlüssel.	symmetrische	symmetrische	asymmetrische ellip. Kurven	asymmetr. homomorphe	asymmetr. homomorphe	symmetrische homomorphe
Signaturen	TTP (vertrauens-) würdige dritte Partei	nicht beschrieben	HMAC - Signaturen	PKI	PKI	Gruppen-signaturen
Zietintervall für Messdaten	Abrechnungs-periode	Abrechnungs-periode	Abrechnungs-zyklus BP	regelmäßige Intervalle	definiert durch Trigger	Berichts-periode j
Pseudonymität	mittels TTP	statistische Destribution	Anomymisier. vor Daten-übermittlung	no-leakage Protokolle	stufigartige Aggregation	kein Verbind. zw. Schlüssel und dem Zähler
Vertraulichkeit	WAN (große Gruppen), MAN (TTP abhäng.)	MAN, WAN (bei vertraunswürd. Zähler)	MAN, WAN	MAN, WAN “no-leakage” Spiel	MAN, WAN	MAN,WAN
Integrität und Authentizität	TTP	sicherer Kanal	nicht beschrieben	Zertifizierungs-stelle	zukunftige Forschung	I. Camenisch/ Losyanskaya oder Ateniese
Flusskontrolle der Information	TTP	sicherer Kanal	nicht beschrieben	Zertifizieruns-stelle iniz.	Operations-plan	sicherer Kanal
Verwaltung der Sicherheitsfunkt.	nicht beschrieben	nicht beschrieben	nicht beschrieben	nicht beschrieben	nicht beschrieben	nicht beschrieben

Tabelle 5: Vergleich der Sicherheitsfunktionen mit BSI Anforderungen

Für die Flusskontrolle ist wichtig, wer die Kommunikation initialisieren darf und wer nicht. Die Verbindung wird von der Kommunikationseinheit initialisiert. Gemäß der aktuellen Version der BSI Schutzprofil ist die Kommunikation zwischen den Geräten in einem Messgerätenetz nicht vorgesehen (PP, 2011). Das bedeutet einige Einschränkungen für drei Lösungen (Mármol u. a., 2011), (Finster und Conrad, 2010) und (Li u. a., 2010) Zwei Autoren sprechen auch über Verwendung von einem sicheren Kanal in Zusammenhang mit der Flusskontrolle der Information. Die Definition und Aufbau eines sicheren Kanals, ist aber nicht beschrieben. Die Verwaltung der Sicherheitsfunktionen steht auch außerhalb dem Forschungsumfang. Wobei dafür gibt es mehrere standardisierte Lösungen, die in normaler Netzadministration sehr breite Verwendung finden.

4.3 Kryptographischer Support

Laut BSI werden die Funktionen, die für den kryptographischen Support relevant sind, zwischen der Kommunikationseinheit und dem Sicherheitsmodul verteilt (PP, 2011). Die Angaben über die Aufteilung fehlen zwar bei den untersuchten Lösungsansätzen, die Funktionen selbst sind aber vorhanden. Einige von denen sind aber nicht vollständig und müssen erweitert werden. Die Ergebnisse des Vergleiches der Funktionen aus dem kryptographischen Support findet man in der Tabelle 6.

Verschlüsselung/Entschlüsselung, die Schlüsselgenerierung und die Zufallzalgenerierung sind wichtige Bestandteile aller Lösungen. Der Unterschied hängt von dem verwendeten Algorithmus ab. Für die Implementierung müssen diese Funktionen detailliert werden, da die meisten Lösungen nur allgemeinen Überblick bieten.

Hash-Code-Anwendung wird kaum beschreiben. Nur in der Lösung mit dem Aufteilung und Aggregation (Finster und Conrad, 2010) wird als Hash-Funktion SHA-256 für mögliche Verwendung vorgeschlagen. Einige Autoren erwähnen die Hash-Code-Anwendung überhaupt nicht. An der Stelle müssen deren Vorschläge über die entsprechende Anwendung erweitert werden

BSI fordert klare Unterteilung in privaten und öffentlichen Schlüssel. Das gesamte Architektur von BSI ist einer Publik-Key-Infrastruktur mit asymmetrischer Kryptographie sehr ähnlich. Diese Anforderung wird bei drei der Lösungen voll erfüllt: die Lösung mit der Aufteilung und Aggregation (Finster und Conrad, 2010),“no-leakage” Protokoll (Garcia und Jacobs, 2010) und die Lösung mit dem aufspannendem Baum (Li u. a., 2010). Die restliche Ansätze verwenden symmetrische Kryptographie. In der der Lösung mit der homomorphen Verschlüsselung nach Castellucia (Mármol u. a., 2011) wird zwar symmetrische Verfahren verwendet, werden aber ein privaten und ein aggregierten Schlüssel definiert. Weiterhin wird in diesem Ansatz vorgeschlagen für die Implementierung die Gruppensignaturen nach

BSI Anforderung	Bohli u.a. (2010) Lösung 1	Bohli u.a (2010) Lösung 2	Finster und Conrad (2010)	Garcia und Jakobs (2010)	Li u.a. (2010)	Marmol u.a. (2011)
Kryptographischer Support						
Kommun. Einheit:						
- Verschlüsselung/ Entschlüsselung	+	+	+	+	+	+
-Hash-Code-Anwendung	-	-	SHA-256	-	-	Teil des Gruppensign.
Sicherheitsmodul (SM)						
- Authentifikation der externen Einheit - Authentifikation des Benutzers	TTP	sicherer Kanal	-	Zertifiz.-stelle	zukunftige Forschung	Camenish/ Lysynskaya oder Ateniese
- Speicherung des privaten Schlüssels	symmetrische Verschlüssel.	symmetrische Verschlüssel.	+	+	+	privater und aggregierter Schlüssel
- Speicherung des öffentlichen Schlüssels			+	+	+	
- Zufallszahl-generierung	+	+	+	+	+	+
- Schlüssels-generierung	+	+	+	+	+	+
- Erstellung und Verifikation von Signaturen	TTP	-	HMAC-Signaturen	PKI	PKI	nach Ateniese

Tabelle 6: Vergleich des kryptographischen Supports mit BSI Anforderungen

Ateniese zu integrieren (Ateniese u. a., 2000). Dadurch erreicht man gute Vergleichbarkeit mit dem asymmetrischen Verfahren.

In dem kryptographischen Support werden die Funktionen, wie Authentifikation und Erstellung und Verifizieren von Signaturen erwähnt. Die Funktionen stellen eine Erweiterung von Authentizität und Integrität sowie Signaturen als Sicherheitsfunktionen aus kryptographischen Sicht dar. Wie schon erwähnt, die genaue Beschreibung dieser Funktionen muss noch in der Lösungen integriert werden. Das betrifft alle Ansätze.

4.4 Schutz gegen Angriffe

Der Schutz der Privatsphäre in allen betrachteten Lösungen steht im Mittelpunkt. Angriffe gegen Privatsphäre sind einen der mehreren Angriffstypen, die BSI Schutzprofil explizit erwähnt. Die Anforderungen wie, Schutz gegen Verletzung der Privatsphäre in Weitverkehrnetz und Heimnetzwerk, Schutz gegen Bekanntgabe der Informationen über die Benutzen sind in allen Lösungen erfüllt. Erfüllungsgrad und mögliche Einschränkungen sind allerdings Algorithmus- und Implementationsabhängig.

Die andere Angriffstypen wie z.B. Kontrolle über die Sicherheitsmechanismen, Beschädigen von Netz oder Netzkomponenten, Modifizieren von Zeitstempel etc., die BSI Schutzprofil ebenso erwähnt, gehören zu sehr verbreiteten Angriffen in einem Netz. In den Lösungsansätzen werden sie nicht untersucht oder nur am Rande erwähnt. Für erfolgreiche Implementierung unter realen Bedingungen müssen die Angriffstypen unbedingt analysiert werden. Letztendlich soll sicher gestellt werden, dass das System sicher gegen solcher Angriffe ist. Was allerdings ein Teil anderer Forschungsarbeit im Bereich intelligenten Messsysteme ist.

4.5 Schutz der Privatsphäre

Schutz der Privatsphäre wird in BSI Schutzprofil als extra Anforderung spezifiziert. Teilweise spiegeln in dieser Anforderung einige Sicherheitsfunktionen wieder. Sie werden nun detailliert beschrieben mit dem Hinblick auf den Schutz der Privatsphäre. Dazu kommen noch andere Anforderungen, wie Datentransparenz für die Kunden und physikalische Analyse des Netzes.

Wie man aus der Tabelle 7 sieht wird Zugriffskontrolle (Verschlüsselung und Pseudonymität) bei allen Ansätzen gewährleistet. Die Umsetzung dieser Funktionen ist von Verschlüsselungsschemas und Lösungsalgorithmen abhängig. Der Grad der Pseudonymität ist ein wichtiger Punkt, der z.B. mittels homomorphe Verschlüsselung, statistische Distribution, Aufteilung oder Baumerzeugung erreicht werden kann.

BSI Anforderung	Bohli u.a. (2010) Lösung 1	Bohli u.a (2010) Lösung 2	Finster und Conrad (2010)	Garcia und Jakobs (2010)	Li u.a. (2010)	Marmol u.a. (2011)
Schutz der Privatsphäre						
Zugriffskontrolle (Veschlüsselung, (Pseudonomität)	+	+	+	+	+	+
Transparenz für Kunden	nicht beschrieben	nicht beschrieben	nicht beschrieben	nicht beschrieben	nicht beschrieben	nicht beschrieben
vertrauenswürdige externe Einheit	s. Sicherheits-anforderungen	s. Sicherheits-anforderungen	s. Sicherheits-anforderungen	s. Sicherheits-anforderungen	s. Sicherheits-anforderungen	s. Sicherheits-anforderungen
nur autorisiert. Parteien	s. Sicherheits-anforderungen	s. Sicherheits-anforderungen	s. Sicherheits-anforderungen	s. Sicherheits-anforderungen	s. Sicherheits-anforderungen	s. Sicherheits-anforderungen
keine Analyse der physicalischen Netzparametern möglich	nicht untersucht	nicht untersucht	nicht untersucht	nicht untersucht	nicht untersucht	nicht untersucht

Tabelle 7: Vergleich des Schutzes der Privatsphäre mit BSI Anforderungen

Wie die Transparenz für die Kunden realisiert wird, welche Mechanismus für Datenablesen bei den Benutzern angewendet wird, ist nicht beschrieben. Die *Log*-Datei, die in BSI Schutzprofil spezifiziert ist, ist bei allen untersuchten Lösungen nicht erwähnt. Die Registrierung aller übermittelten Daten sowie Autorisierung der Benutzer sind aber sehr wichtig, um die Transparenz für die Kunden zu gewährleisten.

Externe Einheiten sollen auch vertrauenswürdig sein. Verschlüsselung Algorithmen zusammen mit Pseudonymität bieten Schutz gegen externe Einheiten, die eventuell mehr Informationen über die Privatsphäre der einzelnen Verbraucher bekommen wollen. Bei Lösungen von Bohli gibt es einige Einschränkungen. Bei Lösung Nr. 1 die Vertrauenswürdigkeit ist nur bei großen Gruppen garantiert. Bei Lösung Nr. 2 die Zähler müssen auch unbedingt vertrauenswürdig sein (Bohli u. a., 2010). Wie schon bei der Beschreibung der Sicherheitsfunktionen erwähnt, Autorisierung spielt eine wichtige Rolle für die sichere Implementierung. Das Autorisierungsmechanismus muss allerdings in allen Ansätzen vollständig beschrieben werden.

Analyse der physikalischen Netzparameter wird bei den Lösungen nicht berücksichtigt. Lediglich bei der Lösung von Finster und Conrad wird ein Versuch gemacht Kommunikationsaufwand zu messen (Finster und Conrad, 2010). Analyse von Zusammenhang zwischen Kommunikationsaufwand und möglicher Veröffentlichung der Information über die Privatsphäre wird nicht gemacht.

Für die vollständige Analyse der Netzparameter braucht man eine Netzmodellierung und Simulation mit echter Anzahl der Knoten (z.B. 1000). Als nächstes muss dass Netz real implementiert und die physikalische Parameter, wie Last, Frequenz, gemessen werden. Aus diesen Messungen soll es unmöglich sein den Schlussfolgen über die einzelne Teilnehmer und die übertragende Daten zu ziehen.

4.6 Allgemeine Anmerkungen

In allgemeinen, lässt sich beobachten, dass der Schutz der Privatsphäre im Mittelpunkt aller Lösungsansätze steht. Die Lösungsalgorithmen, die in dieser Arbeit analysiert sind, sind in erster Linie entwickelt, um dieses Ziel zu erreichen. Kernaspekte sind Verschlüsselung und Pseudonomytät. Zusätzliche Funktionen, die BSI erwähnt, müssen noch in die Lösungen teilweise oder vollständig integriert werden.

Die Architektur der Lösung mit vertrauenswürdiger dritter Partei (Bohli u. a., 2010) passt gut zu BSI Anforderungen. In diesem Fall wird TTP als Kommunikationseinheit interpretiert. Zwischen Energieversorger und intelligenten Zähler gibt es keine direkte Verbindung. Das garantiert Anonymität der Sender. Zwischen den intelligenten Zähler entsteht auch keine

Kommunikation. TTP verfügt allerdings über alle Daten der Verbraucher. Obwohl die Daten verschlüsselt übermittelt werden, stellt ein möglicher Angriff gegen TTP der Schutz der Privatsphäre in Frage. Weiterhin wird in den Lösungsansatz ein symmetrisches Verfahren für Verschlüsselung vorgeschlagen. BSI Schutzprofil priorisiert die asymmetrische Kryptographie mit einem privaten und einem öffentlichen Schlüssel. Das Verschlüsselungsverfahren ist auch nicht genau spezifiziert, was die Implementierung erschwert.

Die Lösung mit der statistischen Distribution (Bohli u. a., 2010) ist zwar sehr interessant, passt aber nicht in die Architektur von BSI Schutzprofil. Hier fehlt die Kommunikationseinheit und die intelligenten Zähler kommunizieren miteinander direkt. Die Verschlüsselung ist wieder symmetrisch. Außerdem, mit der Gruppengröße von 1000 Zähler kann dieser Ansatz keine Anonymität garantieren (Bohli u. a., 2010). Wegen obenbeschriebener Nachteile kommt diese Lösung für die Implementierung nicht in Frage.

Die Architektur der Lösung mit der Aufteilung und Aggregation (Finster und Conrad, 2010) passt gut zu dem BSI Schutzprofil. Hier spielt der Aggregator die Rolle der Kommunikationseinheit. Allerdings ist in der Initialisierungsphase eine Kommunikation zwischen den Zählern notwendig. Außerdem ist die Kommunikationsaufwand bei dieser Lösung relativ hoch, aufgrund des Einsatzes von elliptischen Kurven und HMAC-Signaturen. Das kann die Implementierung mit großen Gruppen auf Mikroprozessoren relativ instabil machen. Wobei der Algorithmus mit den elliptischen Kurven weniger Rechnerleistung und Speicherressourcen als übrige asymmetrische Verfahren benötigt.

Die Lösung mit “no-leakage” Protokoll (Garcia und Jacobs, 2010) hat auch passende Architektur zu dem BSI Schutzprofil. Als Verschlüsselungsverfahren wird hier homomorphe Verschlüsselung nach Paillier vorgeschlagen. Das Verfahren wird in der nächsten Kapitel detailliert analysiert.

Die Lösung mit dem aufspannendem Baum (Li u. a., 2010) passt nicht in die Architektur von dem BSI Schutzprofil. Hier ist direkte Kommunikation zwischen den Zähler notwendig. Außerdem stellt das Baum selbst mehrere Probleme für Implementierung. Beim stabilen und balancierten Baum funktioniert der Algorithmus ganz gut. Ist das nicht der Fall, so ist die Lösung bedenklich.

Die Lösung mit dem Ring (Mármol u. a., 2011) passt teilweise in die Architektur von BSI Schutzprofil. Die erste Widerspruch zu BSI Anforderungen besteht darin, dass die Zähler miteinander kommunizieren. Die zweiter ist die Verbindung zwischen den Zäher und dem Energieversorger. Der Algorithmus soll aber in diesem Fall die Anonymität garantieren, weil keine direkte Zusammenhang zwischen den Schlüssel und Identität des Zählers existiert. Der vorgeschlagene Algorithmus von Castelluccia erfüllt trotz relativ unkomplizierter Aufbau

ebenso vollständig die Kriterien für den Schutz der Privatsphäre. Die Lösung ist auch gut für die Implementierung auf eingebetteten Geräten mit beschränkten Ressourcen geeignet. Mit Aufteilung der Schlüssels in einen aggregierten und einen privaten Schlüssel ist das Verfahren mit der asymmetrischen Kryptographie vergleichbar. Die Lösung lasst sich auch mit der Autorisierungsverfahren erweitern. Das wird in den Lösungsansätzen sogar erwähnt.

Wie den Vergleich zeigt sind drei Lösungen am besten für die Implementierung nach BSI Anforderungen geeignet:

- die Lösung mit der Aufteilung und Aggregation (Finster und Conrad, 2010)
- die Lösung mit “no-leakage” Protokoll (Garcia und Jacobs, 2010)
- die Lösung mit homomorpher Verschlüsselung nach Castellucia (Mármol u. a., 2011)

Die Lösung mit der Aufteilung und Aggregation wird hier nicht mehr weiter betrachtet. Die Implementierung dieses Algorithmus fordert weitere Informationen über die Benutzung der elliptischen Kurven im Protokoll. Die Beschaffung dieser Informationen ist sehr aufwändig und stellt eine weitere Forschungsaufgabe dar. Deswegen wird dieses Verfahren hier nicht mehr betrachtet. Die restliche zwei Algorithmen werden in der folgenden Kapitel detailliert analysiert.

5 Konzept für die praktische Implementierung

In diesem Kapitel wird das Konzept für die praktische Implementierung dargestellt. Hier werden Vorteile von homomorpher Verschlüsselung erläutert und die Verschlüsselungsschema nach Paillier und Castelluccia beschrieben. Am Rande werden auch einige andere homomorphe Verschlüsselungsschemas erwähnt.

5.1 Vorteile von homomorpher Verschlüsselung

Homomorphe Verschlüsselung ist eine kryptographische Transformation, die direkt mathematische Operationen mit verschlüsselten Daten erlaubt. Homomorphe Verschlüsselungsschema ist vorteilhaft, weil eine arithmetische Operation auf n Chiffrate gefolgt von einer Entschlüsselung gleiche Ergebnis liefert, wie dieselbe Operation angewendet auf n entsprechende unverschlüsselte Nachrichten. Homomorphe Verschlüsselung ist besonders wertvoll in Szenarios, wenn eine arithmetische Operation auf eine Reihe der verschlüsselten Nachrichten ohne Kenntnisse der einzelnen Schlüsseln angewendet wird (Castelluccia u. a., 2005).

Ist E eine Verschlüsselung und D eine Entschlüsselung. So wird angenommen, dass k einen Schlüssel zwischen einem Sender und einem Empfänger ist. α ist eine Funktion über Chiffrat und β ist eine Funktion über die Nachricht (Peng und Jian-ping, 2010) .

Homomorphe Verschlüsselung sieht wie folgt aus:

$$D_k(\alpha(E_k(a), \quad E_k(b)))) = \beta(a+b) \tag{12}$$

Eine homomorphe Verschlüsselung ist additiv, wenn:

$$D_k(\alpha(E_k(a), \quad E_k(b)))) = a+b \tag{13}$$

Eine homomorphe Verschlüsselung ist multiplikativ, wenn:

$$D_k(\alpha(E_k(a), \quad E_k(b)))) = a \times b \tag{14}$$

Ein guter Beispiel für homomorphe Verschlüsselung ist RSA-Kryptosystem, dass multiplikativ ist (L.Rivest u. a., 1978). Die RSA Verschlüsselungsfunktion lautet $Enc(m) = m^e = c(mod\ n)$. Die entsprechende Entschlüsselungsfunktion lautet $Dec(c) = c^d = m(mod\ n)$. Wobei m ist eine Nachricht, c ist ein Chiffrat, n ist ein Produkt von zwei großen Primzahlen (p und q), e und d sind Verschlüsselungs- und Entschlüsselüngsexponenten, so dass $e*d = 1(mod(p-1)(q-1))$.

Zwei am häufigsten verwendeten additive homomorphe Verschlüsselungsschemas sind Paillier-Kryptosystem (Paillier, 1999) und Boneh-Goh-Nissim (BGN) Kryptosystem (Boneh u. a., 2006). Pailier-Kryptosystem ist eine asymmetrische Kryptosystem. Die funktioniert wie folgt:

Schlüsselerzeugung:

- p und q sind zwei große Primzahlen. Sie werden so gewählt, dass $ggT(pq,(p-1),(q-1)) = 1$.
- $n = p * q$ und $\lambda = kgV(p-1)(q-1)$
- g ist eine Zufallszahl mit $g \in Z^*_{n_2}$
- $ggT(L(g^\lambda \bmod n^2), n) = 1$), wobei die Funktion $L(u) = \frac{u-1}{n}$
- Öffentlicher Schlüssel$= (n,q)$
- Privatschlüssel$= (p,q)$ oder gleichwertig λ

Verschlüsselung:

- Nachricht $m < n$
- Zufallszahl $r < n$
- Chiffrat $c = g^m \cdot r^n \; mod \; n^2$

Entschlüsselung:

- Chiffrat $c < n^2$
- Nachricht $m = \frac{L(c^\lambda \bmod n^2)}{L(g^\lambda \bmod n^2)} \bmod n$

Diese homomorphe Verschlüsselungsfunktion ist additiv. Daraus folgt:

$$\forall m_1, m_2 \in \mathbb{Z}_n \quad und \quad k \in \mathbb{N} \tag{15}$$

$$D(E(m_1) \cdot E(m_2) \bmod n^2) = m_1 + m_2 \bmod n \tag{16}$$

BGN-Kryptosystem ist eine Erweiterung von Paillier-Kryptosystem mit bilinearen Gruppen (Boneh u. a., 2006). Dafür werden folgende Bezeichnungen verwendet:

- $\mathbb{G}$ und $\mathbb{G}_\mathbb{1}$ sind zwei multiplikative zyklische Gruppen der endlichen Ordnung
- g ist ein Erzeuger von $\mathbb{G}$

- e ist eine bilineare Abbildung $e : \mathbb{G} \quad \times \mathbb{G} \to \mathbb{G}_1$. Anderes formuliert für alle $u, v \in \mathbb{G}$ und $a, b \in \mathbb{Z}$ gilt $e(u^a, v^b) = e(u, v)^{ab}$

$\mathbb{G}$ ist eine bilineare Gruppe, wenn eine Gruppe $\mathbb{G}_1$ und eine bilineare Abbildung, wie oben beschrieben, existieren. Homomorphe asymmetrische Verschlüsselungsschema mit bilinearen Gruppen sieht wie folgt aus:

Schlüsselerzeugung:

- Öffentlicher Schlüssel$= (n, \mathbb{G}, \mathbb{G}_1, e, g, h)$
- $Ord(g) = n$
- $Ord(h) = q_1$
- Privater Schlüssel=q_1

Verschlüsselung:

- Chiffrat $C = g^m h^r \in \mathbb{G}$

Entschlüsselung:

- $C^{q_1} = (q^m h^r)^{q_1} = (g^{q_1})^m$

Ist $\hat{g} = g^{q_1}$. So gilt: um eine ursprüngliche Nachricht zu entschlüsseln, wird einen diskreten Logarithmus von C^{q_1} zum Basis $\hat{g}$ berechnet.

In der Lösung mit “no-leakage” (Garcia und Jacobs, 2010) Protokoll und Lösung mit dem aufspannendem Baum (Li u. a., 2010) wird homomorphe Verschlüsselung nach Paillier für sichere Datenübertragung vorgeschlagen. Die Lösung mit dem Baum kommt aber für die Implementierung trotz der Benutzung von Paillier-Schema nicht in Frage. Der Grund ist, dass hier angewendete baumförmige Netztopologie den BSI Anforderungen nicht genügt. Die Lösung mit dem “no-leakage” Protokoll, trotz aller Vorteile von homomorpher Verschlüsselung, fordert hohe Kommunikationsaufwand zwischen den Zählern und aufwändige Berechnungen. Das stellt gewisse Einschränkungen für Implementierung auf eingebetteten Geräten mit beschränkten Ressourcen.

5.2 Algorithmus mit homomorpher Verschlüsselung nach Castelluccia

In dieser Arbeit wird für die Implementierung eine additive homomorphe Verschlüsselungsschema nach Castelluccia angewendet (Castelluccia u. a., 2005). Dieses Schema entspricht vollständig den Anforderungen für den Schutz der Privatsphäre. Es erlaubt notwendige

statistische Berechnungen ohne, dass man über die Kundendaten und Schlüssel verfügen muss. Der Algorithmus sieht wie folgt aus:

Verschlüsselung:

- Eine Nachricht m ist ein Integer $m \in [0,\ M-1]$, wobei M ist ein großer Integer.
- k ist ein zufallsgenerierter Schlüsselstrom, wobei $k \in [0,\ M-1]$
- Chiffrat $c = Enc(m,\ k,\ M) = m + k(mod\ M)$

Entschlüsselung:

- $Dec(c,\ k,\ M) = c - k(\ \text{mod}\ M)$

Addition von Chiffraten:

- $c_1 = enc(m_1,\ k_1,\ M)$ und $c_2 = enc(m_2,\ k_2,\ M)$
- $k = k_1 + k_2,\ Dec(c_1 + c_2,\ k,\ M) = m_1 + m_2$

Wenn n Chiffrate addiert werden, muss M großer als $\sum_{i=1}^{n} m_i$ sein. Sonst liefert der Algorithmus falsches Ergebnis. Wenn $p = max(m_i)$, dann $M = 2^{\lceil log_2(p*n)\rceil}$ (Castelluccia u. a., 2005).

Der Verschlüsselungsalgorithmus nach Castelluccia ist relativ einfach für Implementierung im Vergleich zu den anderen Algorithmen. Gleichzeitig erlaubt der Algorithmus sehr effiziente Datenaggregation mit symmetrischen Schlüsseln. Wie bereits erwähnt, ist der Verschlüsselungsalgorithmus wegen der Unterteilung in einen privaten und einen aggregierten Schlüssel mit einem asymmetrischen Verfahren vergleichbar. In der Castelluccia-Schema wird statt *xor* modulare Addition verwendet. Solche Schema ist sehr gut geeignet für Mikroprozessoren mit CPU-Einschränkungen. Intelligenten Stromzähler werden in realen System auf solchen Mikroprozessoren implementiert.

5.3 Beschreibung des Modells

Im Mittelpunkt des Konzeptes basiert die Lösung mit homomorpher Verschlüsselung nach Castelluccia (Mármol u. a., 2011). Wie der Vergleich mit BSI Anforderungen gezeigt hat, hat diese Lösung mehrere Vorteile und Übereinstimmungen mit den Anforderungen. In dem Kapitel werden auch einige Modifikationen des ursprünglichen Ansatzes dargestellt. Außerdem wird die Argumentation für die Notwendigkeit diesen Änderungen für prototypische Implementierung erläutert.

BSI Schutzprofil sieht keine Verbindung zwischen den Zählern in einem Messgerätenetz vor

(PP, 2011). Deshalb wird für die Implementierung eine Ring-Topologie mit der Kommunikation zwischen der Zähler abgelehnt. Die ausgewählte Aufbau der Kommunikationstopologie sieht, wie folgt aus (vgl. Abbildung 12):

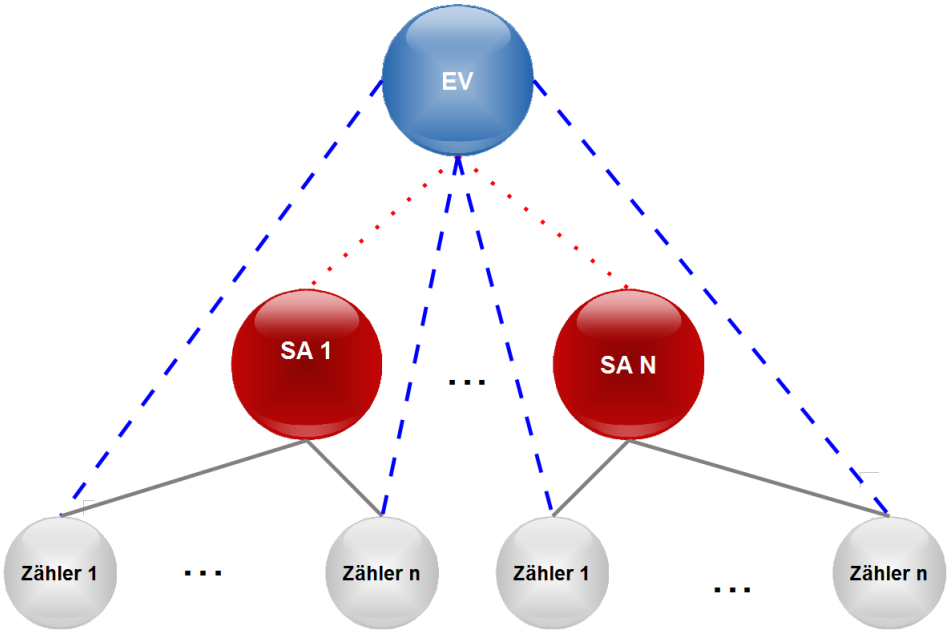

Abbildung 12: Aufbau der Kommunikationstopologie des Konzeptes

- Ein Energieversorger ist mit n Schlüssel-Aggregatoren je nach Anzahl der Gruppen verbunden. Wobei für die prototypische Implementierung ist die Anzahl der Gruppen auf 1 reduziert. Das heißt es gibt nur eine Gruppe mit einem Schlüssel-Aggregator.
- Jeder Schlüssel-Aggregator mit n Zählern je nach Große der Gruppe verbunden
- Zwischen den Zähler gibt keine direkte Verbindung
- Schlüssel-Aggregator spielt die Rolle der Kommunikationseinheit
- Zwischen einem Zähler und einem Energieversorger existiert eine Verbindung. Die Anonymität soll aber dadurch garantiert werden, das es keine Zusammenhang zwischen dem Identität des Zählers und den einzelnen Schlüsseln gibt.

Für die Implementierung wird ein Schema mit der Bestätigungstokens gewählt (vgl. Abbildung 11 in der Kapitel 3). Der Fall wird für alle Szenarios vorgeschlagen, sowohl für einen normalen Fall, als auch für einen Fall mit bösartigen Zähler. Dabei braucht man keinen Umschaltungsmechanismus zwischen beiden Fällen. Der Kommunikationsaufwand wird

dadurch steigen, aber das System wirkt stabiler. Bestätigungstokens sind nicht nur für die Erkennung von bösartigen Zähler nützlich, sondern auch für die verlorene Nachrichten. Wenn einige Nachrichten zwischen dem Zähler und dem Schllüsel-Aggregator verloren gehen, kann der Algorithmus trotzdem weiter arbeiten. Der aggregierten Schlüssel wird abhängig von den *Tokens*, die von dem Energieversorger empfangen werden, berechnet. Nur wenn die Bestätigungstokens von dem Energieverbraucher verloren gehen, kann die Entschlüsselung nicht ausgeführt werden.

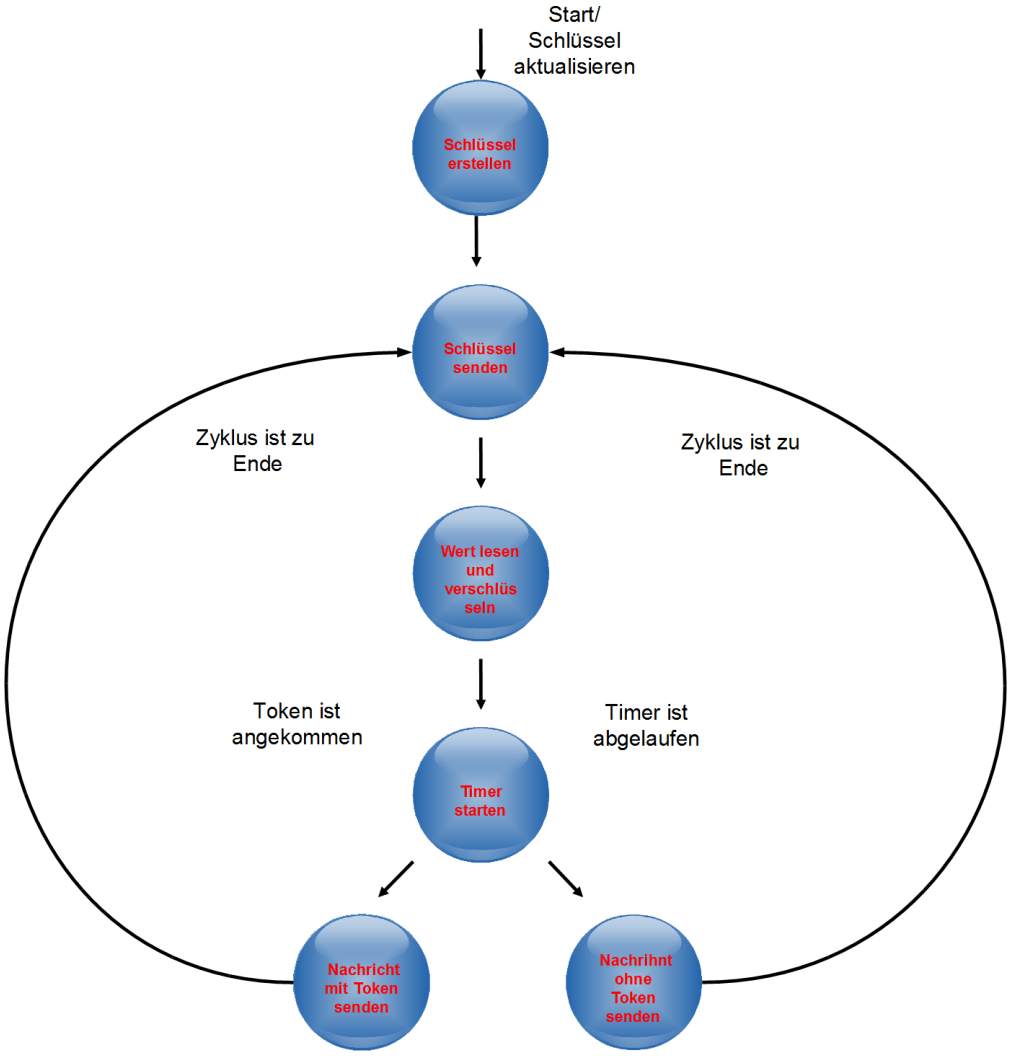

Abbildung 13: Zustandsautomat des Zählers. Hauptfunktionalität

Die drei Haupteinheiten: Energieversorger, Schlüssel-Aggregator und Zähler lassen sich am besten mit Hilfe der Zustandsautomaten modellieren.

Jeder Zähler hat die Aufgabe nach der Initialisierung der Runde einen privaten Schlüssel zu generieren. Der Schlüssel wird gültig bis eine Anfrage auf eine Aktualisierung des Schlüssels kommt. Weiterhin arbeitet jeder Zähler in einer Schleife gemäß der gewählten Zyklusperiode. Wartezeit für den Empfang des Bestätigungstokens, das von dem Schlüssel-Aggregator kommt, wird mit einem *Timer* überwacht. Kommt während dieser Zeit keine Bestätigung, sendet der Zähler eine verschlüsselte Nachricht ohne Token.

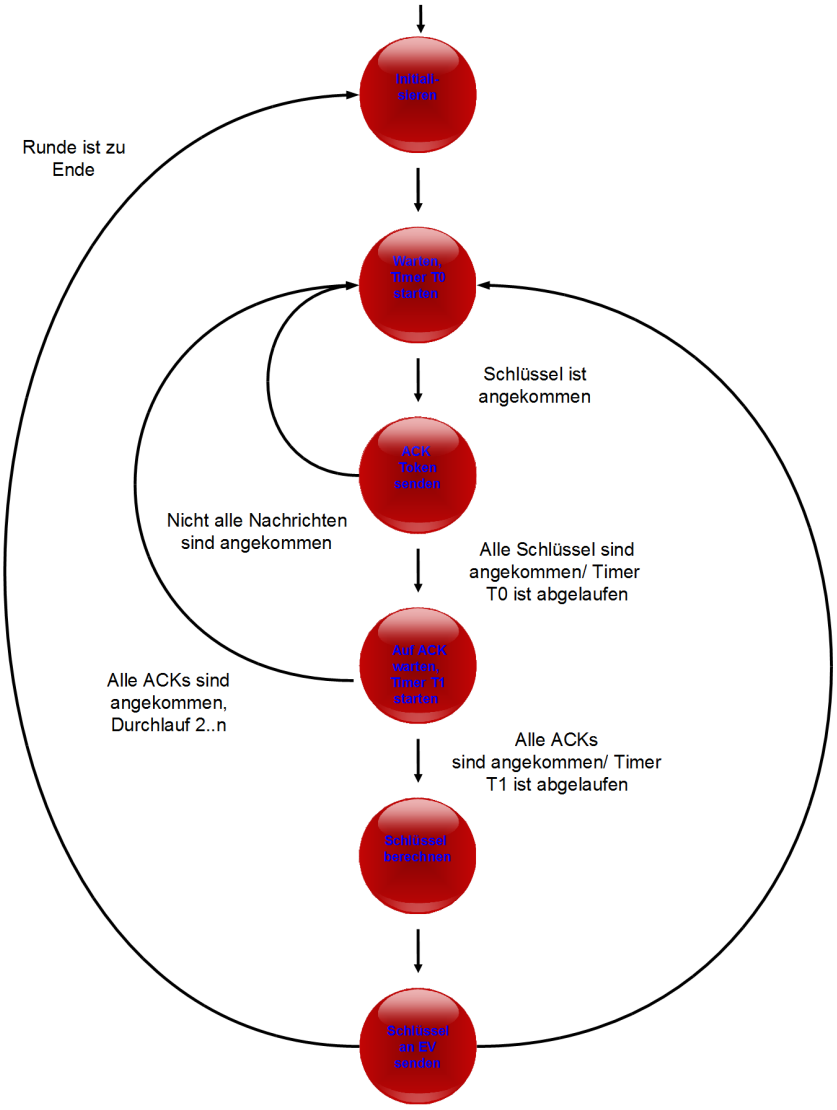

Abbildung 14: Zustandsautomat des Schlüssel-Aggregators. Hauptfunktionalität

Der Schlüssel-Aggregator aggregiert die Schlüssel entsprechend den Bestätigungstokens, die von Energieverbrauher gesendet werden. Aggregation läuft nach der Formel von Castelluccia (Castelluccia u. a., 2005). Dabei werden die Schlüssel modular n addiert:

$$K = f(k_{1j}, k_{2j}, \ldots, k_{nj}) = \bigoplus_{i=1}^{n} k_{ij} = \sum_{i=1}^{n} k_{ij} \tag{17}$$

Nicht bestätigte Schlüsseln werden weggeworfen und nicht aggregiert. Aggregierter Schlüssel wird nur einmal pro Runde Berechnet. Es sei denn die Anzahl der Schlüsseln mit der Bestätigungen von Energieversorger nicht übereinstimmt. In diesem Fall wird der Schlüssel auch innerhalb der Runde wiederholt berechnet (Mármol u. a., 2011). Wartezeit sowie die Gültigkeit des Schlüssels werden durch *Timer* reguliert.

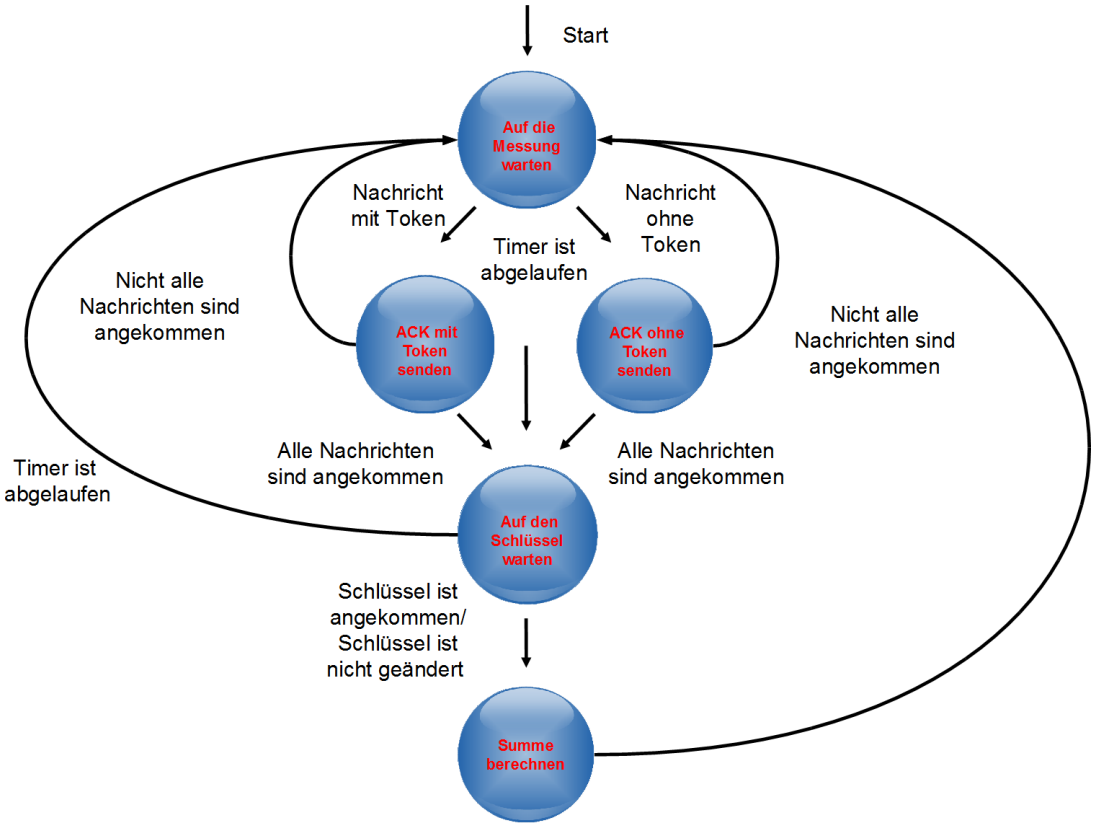

Abbildung 15: Zustandsautomat des Energieversorgers. Hauptfunktionalität

Der Energieversorger wartet auf verschlüsselte Nachrichten von den Zähler und bestätigt die mit einem Token. Wenn die Anzahl der verschlüsselten Nachrichten während der Runde gleich bleibt, wartet der Energieversorger auf einen neuen aggregierten Schlüssel nicht.

Wenn die Anzahl der Nachrichten nicht in Frage sich über die Runde nicht in Frage, wird die Entschlüsselung mit einem neuem Schlüssel durchgeführt. Wenn die Anzahl der Nachrichten mit der Anzahl der Knoten von dem Aggregator nicht übereinstimmt, wird die Entschlüsselung nicht ausgeführt. In diesem Fall fängt der Algorithmus von vorne an.

5.4 Analyse der möglichen Probleme

Eine prototypische Implementierung dient hauptsächlich dazu, eine Hauptfunktionalität des Algorithmus zu zeigen. Dabei werden einige reale Bedingungen vereinfacht oder vernachlässigt. Vor der Implementierung wurde eine Reihe der möglichen Probleme analysiert. Dabei wurden folgenden Anforderungen an die Implementierung formuliert:

- *Erstellen einer Gruppe*. Hier geht es um statische oder dynamische Gruppenbildung. Für die prototypische Implementierung reicht eine statische Gruppe mit der festen Anzahl der Mitgliedern. Die Verbindung mit einigen Zähler kann allerdings, unter Umständen, verloren gehen.
- *Größe einer Gruppe*. Die Größe der Gruppen soll so gewählt werden, das sie die Große einer realen Gruppe entspricht. Die Große soll mit einem Simulator testbar sein.
- *Gruppe von Zählern Verlassen und wieder Eintreten*. Der Algorithmus soll in beiden Fällen funktionieren.
- Aktualisierung eines privaten Schlüssels. Der privaten Schlüssel soll nach einer festgelegten Periode neue berechnet werden.
- *Schlüsselgenerierung*. Der Schlüssel soll mit einem Zufallszahlgenerator erzeugt werden.
- *Länge des Schlüssels, Länge der verschlüsselten Nachricht im Fall einer großen Gruppe*. Die Große soll so gewählt werden, das der Algorithmus funktionsfähig bleibt. Die Länge darf nicht zu klein aber auch nicht übermäßig groß sein.
- *Übertragbarkeit der Software auf die reale Geräte*. Die Applikation soll auf einem realen Gerät laufen.
- *Kommunikationsaufwand und Stabilität des System im Fall der möglichen Überlastung*. Das System soll selbst mit einer großen Gruppe stabil bleiben.

Die detaillierte Beschreibung für die Umsetzung diesen Anforderungen enthält das Kapitel “Implementierung eines Prototyps”.

6 Implementierung eines Prototyps

In dieser Arbeit sollen folgende Ziele für die prototypische Implementierung erreicht werden:

- Das ausgewählten Verfahren soll für die Implementierung auf kleinen eingebetteten Geräten geeignet sein. Solche Geräte werden in der Zukunft als intelligente Zähler hauptsächlich eingesetzt.
- Die Aggregation von 1000 Werten soll richtiges Ergebnis liefern. 1000 wird als Beispielzahl einer großen Gruppe gewählt. Natürlich muss das Algorithmus richtiges Ergebnis für alle mögliche Aggregationswerte bis einer physikalische Einschränkung liefern.
- Die Größe des Moduls M in dem Algorithmus mit 1000 aggregierten Werten muss ermittelt werden. Darüber hinaus muss festgestellt werden, ob es Einschränkungen für den Modul gibt. Auch die Größe des Chiffrats und Schlüssels muss ermittelt werden.

In allgemeinen ist es zu zeigen, dass die obengenannten Ziele erreichbar sind. Wenn die Ziele nicht erreicht werden können, so sollen die Grunde dafür aufgelistet werden. In diesem Fall ist einen Verbesserungsvorschlag zu machen.

6.1 Einsatz der Werkzeuge und Hardware

Der Algorithmus mit homomorpher Verschlüsselung nach Castelluccia wird unter TinyOs 2.0 Betriebsystem implementiert. TinyOs ist ein *Open-Source* und ereignisorientiertes Betriebssystem. TinyOs wird speziell für Sensorknoten von Universität Berkeley, Kalifornien, entwickelt. Ursprünglich wurde das Betriebssystem in Programmiersprache C geschrieben und später in NesC rekompiliert. NesC ist eine Erweiterung von C, die speziell für Sensornetze entwickelt worden ist. Die Module, die auf Hardware von Sensorknoten laufen, werden auch in NesC geschrieben. NesC hat folgende Eigenschaften (Gay u. a., 2003; Culler, 2006):

- *Komponente*. Das Programm besteht aus Komponenten, die miteinander verdrahtet sind. Die Komponenten sind entweder Module oder Konfigurationen. Ein Modul enthält Implementierung in NesC-Code. In der Konfiguration findet man die Komponente, die bestimmte Konfigurationen bilden, sowie Verdrahtung der Komponenten.
- *Bidirektionale Schnittstellen*. Schnittstellen sind Spezifikationen für die Implementierung der Komponenten. Die Spezifikation besteht aus den Befehlen und Ereignissen. Befehle werden vom Anbieter der Schnittstelle implementiert. Ereignisse werden vom Benutzer der Schnittstelle implementiert.

- *Nebenläufigkeit und ereignisorientiertes Modell*. Nebenläufigkeit wird mittels einen gemeinsamen Stack realisiert. Die konventionelle Betriebssysteme nutzen ein eigenen Stack für jeden Thread. Im Mittelpunkt des Systems steht ein *Dispatcher*, der *Interrupt-Handlers* aufruft. Die laufenden Tasks werden durch Ereignisse unterbrochen.

Eine typische Konfiguration für TinyOs hat 5 Subsysteme - Sensors/Aktoren, Kommunikation, Speicher, *Timer*, und Prozessor/Strom Management (Culler, 2006). Die Konfiguration wird in der Abbildung 16 dargestellt.

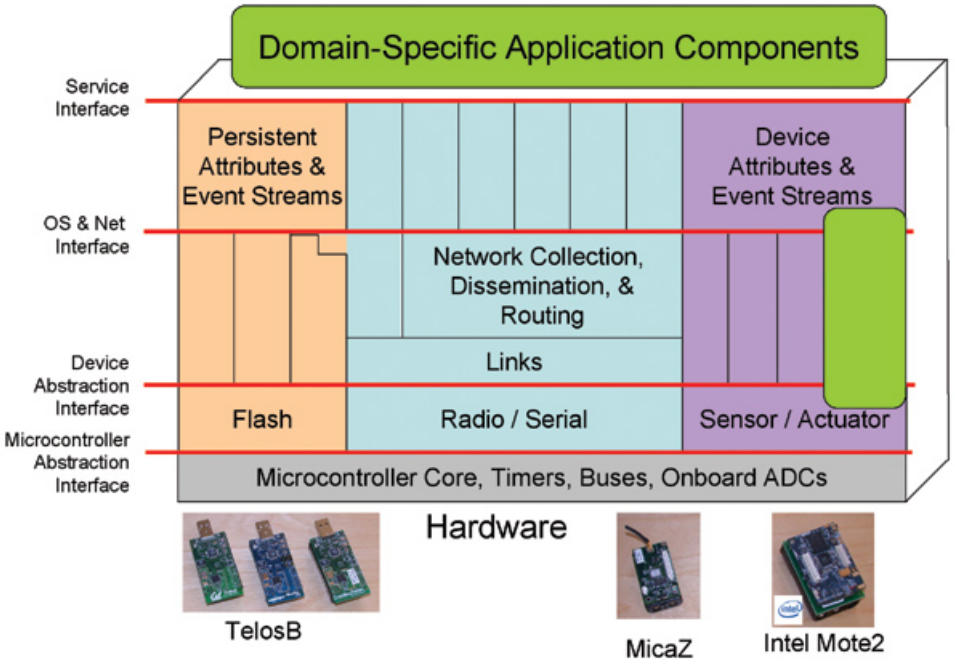

Abbildung 16: Eine typische TinyOs Konfiguration (Culler, 2006)

Für die Simulation der größeren Netzen wird der Simulator TOSSIM benutzt. TOSSIM ist speziell für TinyOs entwickelt und vereinfacht die Entwicklung der Applikationen. Die einzige Einschränkung für die Simulation ist, dass nur NesC-Code, die für die Geräte *Micaz* kompiliert wurde, läuft im Simulator. Die Code selbst ist allerdings Plattform-unabhängig.

Außerdem stellt TinyOs ein graphisches Visualisierungswerkzeug *Graphviz* für Quellcode zur Verfügung. Mit dem Tool wird Dokumentation für die Komponente und die Schnittstellen erzeugt. Die Verdrahtung zwischen einzelnen Komponenten wird graphisch dargestellt.

Die Applikationen für intelligente Zähler und den Schlüssel-Agregator werden auf *TelosB* und *Tmote Sky* Geräten implementiert (vgl. Abbildung 17). *Tmote Sky* ist in dem

Entwurf identisch mit *TelosB* Plattform. Das Gerät wird nur von einen anderen Hersteller vertrieben. *Tmote Sky* bzw. *TelosB* verfügen unter anderem über folgenden Eigenschaften (TmoteSky, 2006; TelosB, 2004):

- Entspricht dem Standard IEEE 802.15.4 für WPAN
- 8MHz Texas Instruments MSP430 Mikrocontroller (10k RAM, 48k Flash)
- 16-Bit RISC-Prozessor
- Datensammlung und Programmierung via USB Schnittstellen
- Geringer Stromverbrauch
- TinyOS-Support

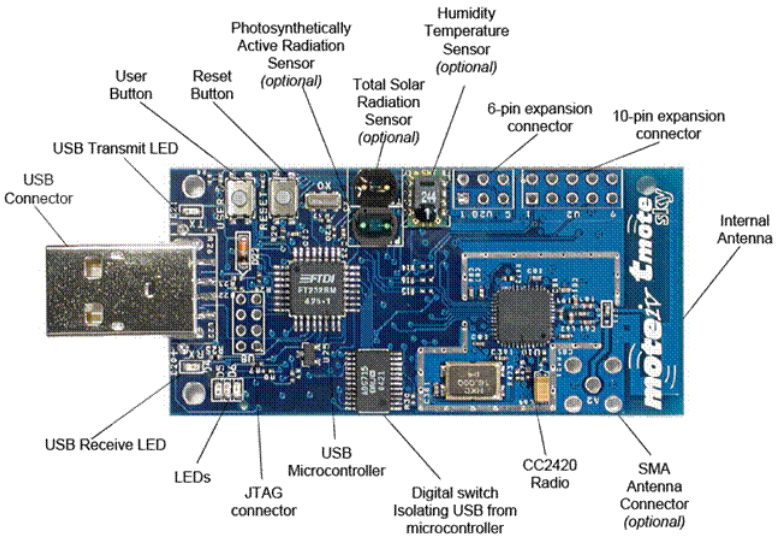

Abbildung 17: Tmote Sky (TmoteSky, 2006)

Die Applikationen, die auf Sensorknoten laufen, kommunizieren miteinander über eine sogenannte Radio Schnittstelle. Die Kommunikation zwischen den Sensorknoten und anderen PC-Applikationen erfolgt über eine serielle Schnittstelle. Die PC-Applikationen können in Java oder C programmiert werden. In dieser Arbeit wird eine PC-Applikation in Java implementiert.

6.2 Eigenschaften und Einschränkungen der Implementierung

Das Prototyp hat folgende Eigenschaften und Einschränkungen:

- *Die Gruppe*. Gruppen sind statisch definiert. Die Anzahl der Mitglieder wird fest vor der Applikationsstart gesetzt. Die maximale Anzahl der Zähler ist 1000. Die minimale sinnvolle Anzahl ist 2. Mit einem Zähler kann die Aggregation nicht erfüllt werden. Die Große entspricht auch der Große einer realen Gruppe. Ein Verlassen und ein Wiedereintreten der Gruppe wird bei der Implementierung erkannt.

- *Der Schlüssel*. Die Länge des privaten Schlüssels ist 64 Bit. Die Länge des aggregierten Schlüssels ist 64 Bit. Der Schlüssel wird mit einem Zufallszahlengenerator erzeugt. Die Aktualisierungsperiode für die neue Schlüsselerzeugung ist durch einen *Timer* einstellbar. Für die Testzwecke ist eine Periode von 240 Sekunden gewählt.

- *Die Nachricht*. Die Länge der Messung ist 4 Byte. Die Länge der verschlüsselten Nachricht ist 64 Bit. Die Länge der entschlüsselten Nachrichten ist 64 Bit. Die Berichtsperiode für die Messungsübermittlung ist durch einen *Timer* geregelt. Für Simulationszwecke ist eine Periode von 60 Sekunden gewählt.

- *Die Authentifikation*. Das Mechanismus ist nicht Implementiert. Das ist die Aufgabe einer weiteren Forschung.

- *Die Tmote bzw. TeloSB Geräte*. Die Applikation ist mit 3 Tmote und TelosB Geräten getestet.

- *Die Simulation*. Die Applikation ist in dem Simulator TOSSIM für 1000 Knoten getestet. Schlüssel-Aggregator und Energieversorger werden in Simulation auch als Sensorknoten realisiert.

6.3 Softwarearchitektur

Das Protokoll, das für die prototypische Implementierung gewählt wurde, verwendet die Lösung von Mármol u.a. (Mármol u. a., 2011). Dabei wurde nur den Fall mit bösartigen Zähler Implementiert (vgl. Abbildung 11). Das Protokoll selbst wurde allerdings teilweise geändert. Wie man in der Abbildung 18 sieht, wurde einige Bestätigungsnachrichten zum ursprünglichen Protokoll hinzugefügt. Bei der Implementierung wurde festgestellt, das mehrere Nachrichten im Radiokanal verloren gehen, wenn die Anzahl der Zählern in einer Gruppe steigt. Die Implementierung der Betsätigunsgnachrichten führte dazu, dass die verlorene Nachrichten wiederholt gesendet wurden.

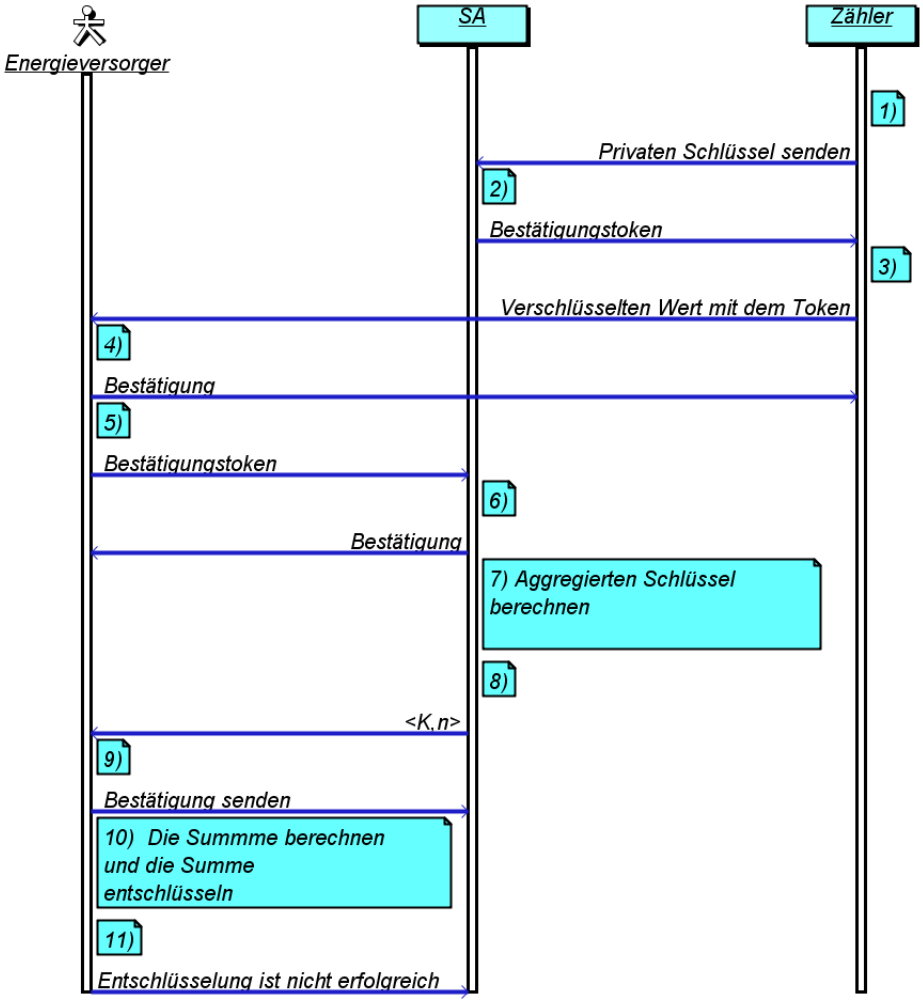

Abbildung 18: Das Protokoll für die prototypische Implementierung

Wie bei allen NesC-Applikationen, besteht die Software aus verdrahteten Komponenten. Komponenten sind in Module und Konfigurationen unterteilt. Module enthalten die Implementierung und die Konfigurationen die sich nur auf Abhängigkeiten der Komponenten beziehen. Die Hauptkomponenten sind:

- Sensor;
- Key Aggregator;
- Supplier;
- Drahtlose und serielle Schnittstellen;

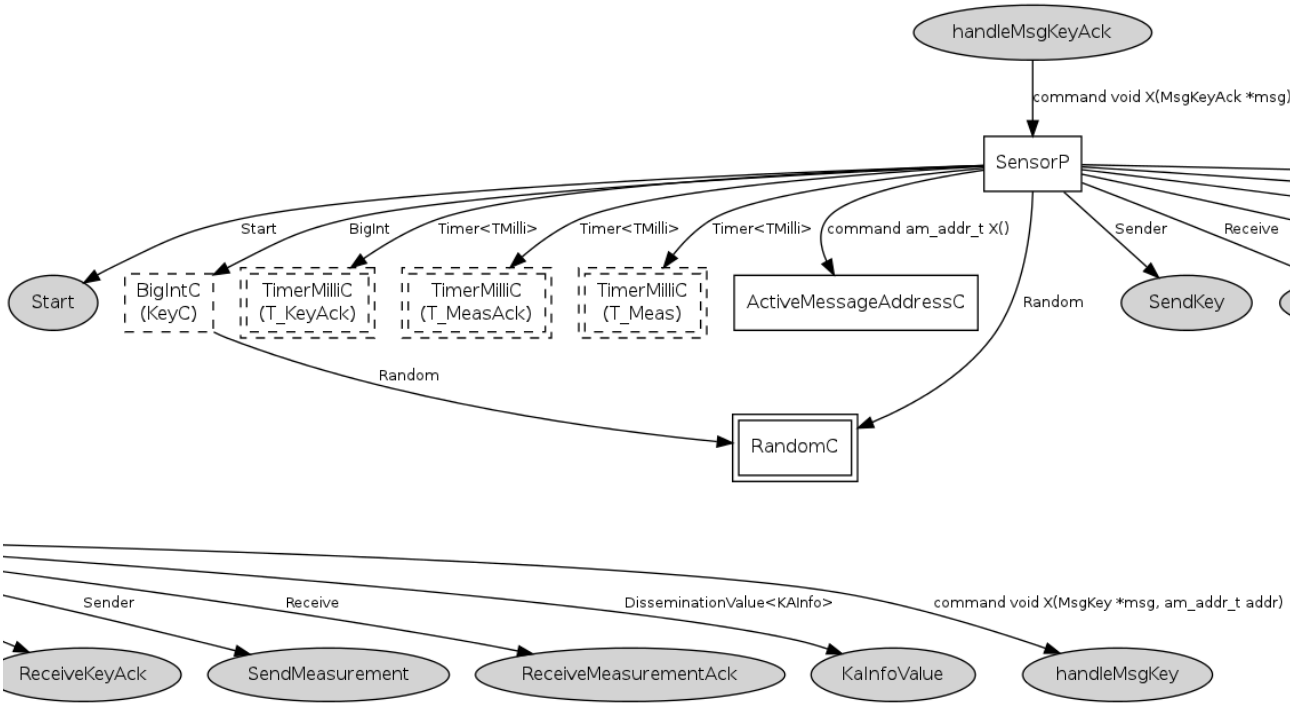

Abbildung 19: Konfiguration der Komponente “Sensor”

Die Komponente “Sensor” stellt den intelligenten Zähler mit seinen Funktionen dar. Die Konfiguration wird in der Abbildung 19 dargestellt. Die Komponente verfügt über interne Funktionen, um den Schlüssel und die verschlüsselte Nachricht zu senden. Der Schlüssel wird immer in einem Zyklus, anhängig von dem *Timer* T0, gesendet. Alle andere Funktionen, inklusive *Timer*, sind in Form der Ereignissen implementiert. Der Zähler erstellt einen Schlüssel, sendet ihn, wartet auf die Bestätigung und am Ende des Zyklus sendet die verschlüsselte Nachricht. Wenn keine Bestätigung von dem Schlüssel-Aggregator kommt und der *Timer* T1 abläuft, wird eine Nachricht ohne Bestätigungstoken gesendet.

Für die Erstellung des Schlüssels wird die Komponente *BigInt* und *Random* benutzt (vgl. Abbildung 22). Die Länge des Schlüssels ist 8 Byte (64 Bit). $k \in [0; 2^{64} - 1]$. Der Schlüssel wird mit einem Zufallsgenerator erstellt. Die Random-Funktionen sind in der Komponente *Random* schon vorhanden. Es gibt die Möglichkeit eine Zufallszahl der Länge 16 oder 32 Bit zu bekommen. In der Implementierung wird die Funktion mit 32-Bit Rückgabewert 8 mal

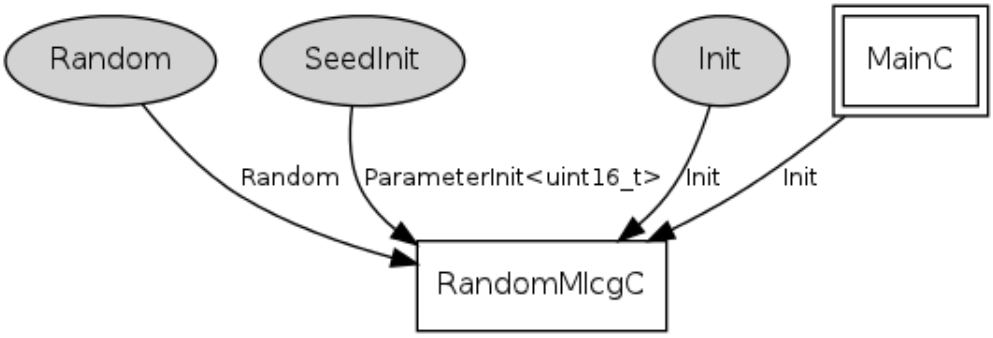

Abbildung 20: Konfiguration der Komponente “Random”

aufgerufen. Dabei bei jedem Aufruf werden die 8 letzte Bits in einem Feld gespeichert. Das Feld wird dann mit der Nachricht, die an den Schlüssel-Aggregator gesendet wird, übergeben. Der Modul M it gleich 2^{64}. Die Länge der unverschlüsselten Nachricht ist 4 Byte (32 Bit) und der verschlüsselten Nachricht 8 Byte (64 Bit). In diesem Fall darf laut Castelluccia (Castelluccia u. a., 2005) 2^{32} Zähler in der Kommunikation teilnehmen. Für 1000 Zähler reicht das vollkommen aus.

Die Komponente “Key aggregator” verfügt über die Funktionen von dem Schlüssel-Aggregator. Die Konfiguration des Moduls wird in der Abbildung 20 dargestellt. Der Schlüssel-Aggregator wartet auf die Schlüssel von einzelnen Zähler und antwortet mit einem Bestätigungstoken, wartet auf Bestätigung von dem Energieversorger, berechnet der aggregierten Schlüssel und die Anzahl der Bestätigungen und schließlich sendet die Information an den Energieversorger. Die Aufgaben des Schlüssel-Aggregators sind in Funktionen und Ereignisse unterteilt. Bestätigungsnachrichten werden in Form der Ereignissen implementiert. Außerdem initialisiert der Schlüssel-Aggregator auch eine Aktualisierung der Runde für die Zähler. Nach dem Ablauf des *Timers* sendet er die entsprechende Information an die Zähler, damit sie die neue privaten Schlüsseln erzeugen können.

Der Schlüssel-Aggregator ist für die Berechnung des aggregierten Schlüssel verantwortlich. Nach dem Algorithmus von Castelluccia wird der aggregierten Schlüssel als die Summe einzelner Schlüsseln berechnet: $K = k_1 + k_2 + \ldots k_n$, wobei der aggregierten Schlüssel $K \in [0, M-1]$ (Castelluccia u. a., 2005). Die einzelne Schlüssel, die von den Zähler empfangen werden, sind in einem 8 Byte Feld gespeichert. Der Schlüssel-Aggregator addiert nur die Schlüssel der Zähler, für die er eine Bestätigung bekommen hat. Die andere Schlüssel werden nicht addiert. Die Anzahl der Zähler wird bei fehlender Bestätigung nicht inkrementiert.

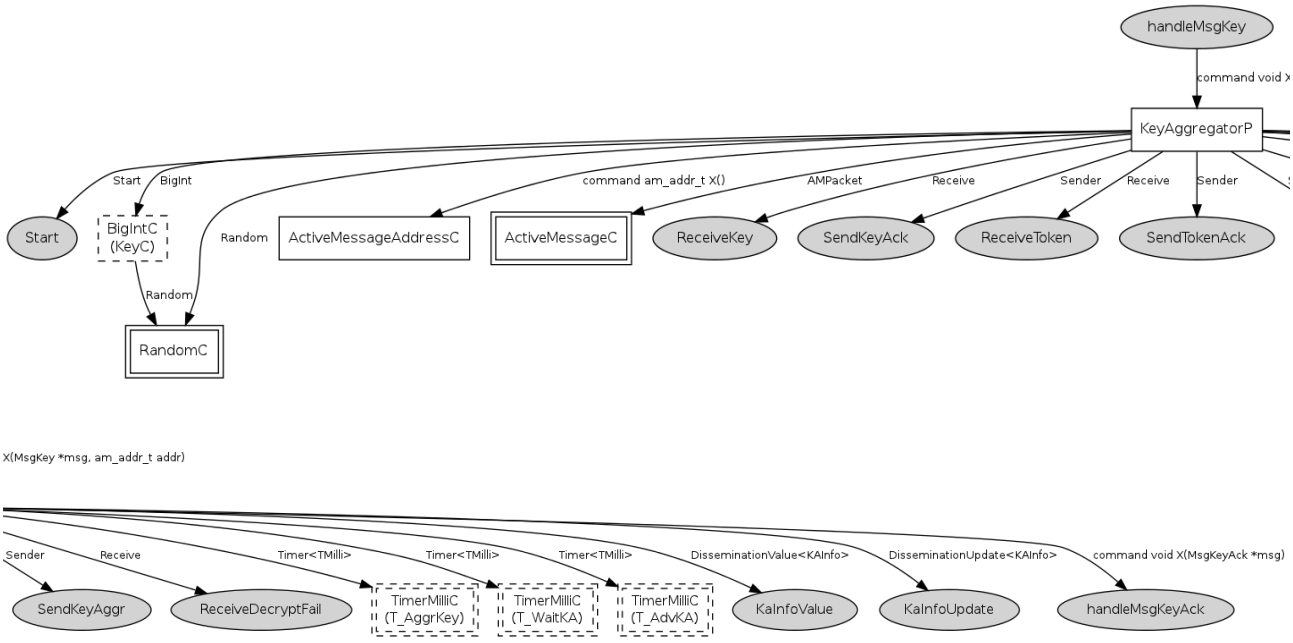

Abbildung 21: Konfiguration der Komponente “Key Aggregator”

Die Komponente “Supplier” repräsentiert einen Energieversorger. Der Energieversorger bekommt die verschlüsselte Nachrichten von den Zählern, sendet Bestätigungen an den Schlüssel-Aggregator und berechnet die Summe aller Messungen mit dem aggregierten Schlüssel, der von dem Schlüssel-Aggregator erzeugt wird.

Für die Simulationszwecke wird die Komponente “Supplier” in NesC geschrieben. Für die Implementierung auf den realen Geräten wird die Applikation zusätzlich in Java geschrieben. Die Java-Implementierung wird detailliert unten beschrieben.

Der Energieversorger summiert alle verschlüsselte Nachrichten. Dann subtrahiert er von dem Ergebnis den aggregierten Schlüssel. Die Subtraktion-Funktion ist eine Umkehrfunktion zu modularer Addition. Das Ergebnis wird danach noch mal entsprechend des Moduls M angepasst. Die Summe aller Nachrichten ist $m_1 + m_2 + \ldots m_n \in [0 \ldots 2^{64} - 1]$.

Die Kommunikation zwischen den Modulen erfolgt über eine drahtlose oder serielle Schnittstelle. Die Kommunikation zwischen den Zähler und dem Schlüssel-Aggregator benötigt eine Schnittstelle für eine drahtlose Verbindung (vgl. Abbildung 23). Die Kommunikation zwischen

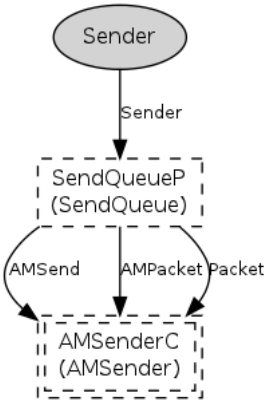

Abbildung 22: Konfiguration der Komponente “Serial Sender”

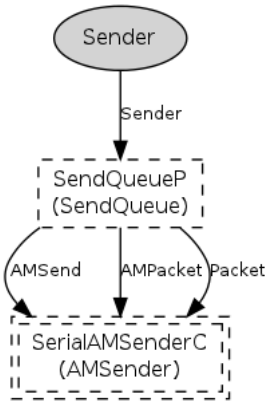

Abbildung 23: Konfiguration der Komponente “Wireless Sender”

den Zähler und dem Energieverbraucher sowie zwischen dem Schlüssel-Aggregator und dem Energieverbraucher wird mittels einer seriellen Schnittstelle aufgebaut (vgl. Abbildung 24). Die Komponente “AMsenderC” und “SerailAMsenderc” stehen in der *TinyOs* Bibliothek zur Verfügung. Das Empfangen von Nachrichten wird ebenfalls mittels einer drahtlosen und einer seriellen Schnittstelle realisiert.

6.4 Darstellung der Testfällen

Die Implementierung wurde in dem Simulator TOSSIM getestet. Dabei sind verschiedene Gruppen getestet: angefangen von kleiner übersichtlichen Gruppe bis einer Gruppe mit 1000 Knoten. Dabei wurde nicht nur die Richtigkeit der Berechnungen überprüft, sondern auch die

Stabilität des Systems. Die Berechnungen mit Zufallszahlen für kleine Gruppen sind relativ übersichtlich (vgl. Listing 1).

```
DEBUG (1): sending value = 1160519691
DEBUG (2): sending value =  448123021
DEBUG (3): sending value = 2003128752
DEBUG (999): Total value = 3611771464 (3)
```

Listing 1: Test mit 3 Sensors

Für die größere Gruppe werden Messwerte nicht nur als Zufallszahlen gewählt, sondern auch Werte von 1 bis 1000. So kann man die Richtigkeit des Ergebnisse mit Gauss-Formel ($\sum_{i=1}^{N} = (N+1)(N/2)$) überprüfen (vgl. Listing 2). Beim dem Test mit größeren Gruppen lässt sich beobachten, dass die Nachrichten in gemischter Reihenfolge empfangen und gesendet werden .

```
DEBUG (146): sending value =        146
DEBUG (81): sending value =         81
DEBUG (150): sending value =        150
DEBUG (133): sending value =        133
DEBUG (117): sending value =        117
_____________________________________________
DEBUG (999): Total value =      32896 (256)
```

Listing 2: Test mit 256 Zähler

Die Zähler werden nacheinander gestartet. Dabei in einer großen Gruppe beginnt die Aggregation, wenn die Anzahl der Zähler noch nicht ihr Maximum erreicht hat. Dadurch wird gezeigt, dass der Algorithmus funktionsfähig bleibt, wenn die Anzahl der Zähler in einer Gruppe kleiner, als gesetzter Maximum ist. Wenn neue Zähler in einer Gruppe eintreten, wird die Berechnung für neue Anzahl der Zähler durchgeführt. Dabei wird die Bedingung von Verlassen und Eintreten einer Gruppe simuliert.

```
DEBUG (999): Total value =  139128000 (528)
DEBUG (999): Time (KA) = 36.410 sec
DEBUG (999): Total value =  139128000 (528)
DEBUG (999): Time (normal) = 31.385 sec
DEBUG (999): Total value =  139128000 (528)
DEBUG (999): Time (normal) = 31.384 sec
```

Listing 3: Test mit 528 Zähler plus Zeitangabe

Bei der Simulation wurde auch die Zeit gemessen, die für empfangen allen Nachrichten sowie die Berechnung und Empfangen einem aggregierten Schlüssel benötigt wird. In den Listing 3 und 4 wurde zwei Beispiele für 528 und 999 Zähler dargestellt. Time (KA) ist die Zeit des Algorithmus, wenn den aggregierten Schlüssel von dem Schlüssel-Aggregator neue berechnet und gesendet wird. Time (normal) ist die Zeit des Algorithmus, wenn der aggregierten Schlüssel konstant bleibt. Der Unterschied zwischen beiden Zeiten liegt immer bei 5 Sekunden. Das ist genaue die Wartezeit von dem Schlüssel-Aggregator auf alle angekommene Bestätigungen.

```
DEBUG (999): Total value =  498501000 (999)
DEBUG (999): Time (KA) = 60.454 sec
DEBUG (999): Total value =  498501000 (999)
DEBUG (999): Time (normal) = 55.460 sec
DEBUG (999): Total value =  498501000 (999)
DEBUG (999): Time (normal) = 55.462 sec
DEBUG (999): Total value =  498501000 (999)
DEBUG (999): Time (KA) = 64.196 sec
DEBUG (999): Total value =  498501000 (999)
DEBUG (999): Time (normal) = 59.192 sec
DEBUG (999): Total value =  498501000 (999)
DEBUG (999): Time (normal) = 59.198 sec
DEBUG (999): Total value =  498501000 (999)
DEBUG (999): Time (KA) = 60.757 sec
DEBUG (999): Total value =  498501000 (999)
DEBUG (999): Time (normal) = 55.797 sec
```

Listing 4: Test mit 999 Zähler plus Zeitangabe

Die Tests in Simulator haben gezeigt, dass je großer die Gruppe ist desto weniger stabil das System wird. Bei der Gruppe mit 1000 Knoten wurden so viel Nachrichten in einem Radiokanal verloren gegangen sind, dass die Berechnung von dem aggregierten Werten sehr erschwert war. An der Stelle muss die Implementierung noch verbessert werden.

6.5 Visualisierung von Datenaggregation im prototypischen Netz

Die Datenaggregation im prototypischen Netz wird nicht nur im Simulator TOSSIM, sondern auch in einer Java-Applikation visualisiert. TOSSIM ist am besten für einen Test mit einer großen Gruppe geeignet. Die Java-Applikation wird für die Implementierung mit realen *TelosB* bzw. *Tmote Sky* Geräten benutzt.

Die Kommunikation zwischen den *TelosB* Geräten und der Java-Applikation erfolgt über *PC-mote* serielle Schnittstelle. Die Java-Applikation enthält GUI-Visualisierung und Funktionen eines Energieversorgers. Die Funktionalität entspricht dem Moduls “Supplier”. Der Energieversorger agiert entsprechend dem oben beschrieben Algorithmus von Catelluccia (Castelluccia u. a., 2005) und dem dargestellten Protokoll (vgl. Abbildung 18).

Die Visualisierung ist in den Abbildungen 24 und 25 dargestellt. Wobei die Diagramm zeigt den Fall, wenn die Messungen gleich über Zeit blieb. Die Messungen sing gleich Knoten-ID⋆1000. Bei der zweiten Diagramm werden die Messwerten mittels eines Zufallszahlengenerator erstellt.

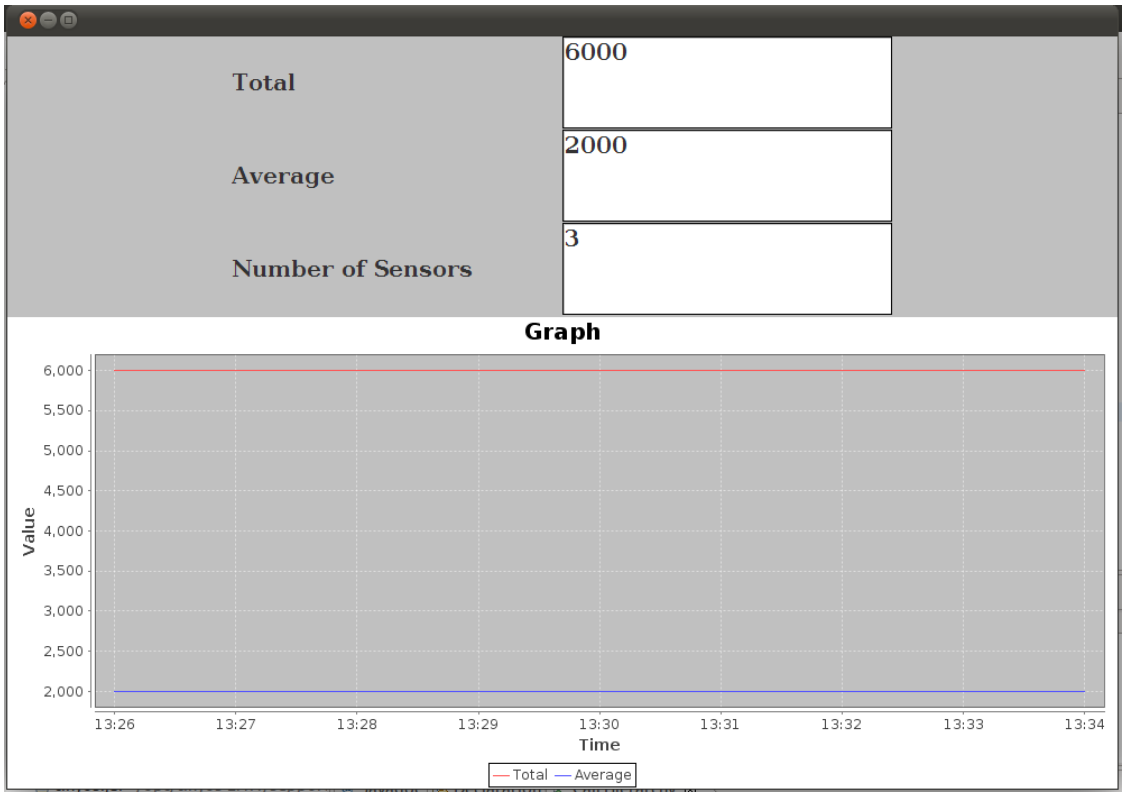

Abbildung 24: Visualisierung der Ergebnisse in Java. Konstante Messung

Die Visualisierung der Ergebnisse wurde in der Form einer Diagramm realisiert. Mit jedem neuem aggregierten Ergebnis wird die Diagramme aktualisiert und neue Diagramme gezeichnet. Die Summe aller Messungen wird in Abhängigkeit von der Zeit dargestellt.

TinyOs stellt einige Tools zu verfügen um die Kommunikation zwischen den Geräten

und PC-Applikation über serielle Schnittstellen zu vereinfachen. *Listen Tool* hilf die Kommunikation zwischen den PC und dem Gerät aufzubauen und einen seriellen Port zu öffnen. Dabei beim Starten von *Listen Tool* muss der seriellen Port für die Kommunikation genannt werden. Die empfangenen Pakete werden als Bytestrom ausgegeben. Mit *Serial Forwarder* werden die empfangene und gesendete Pakete über einer serielle Schnittstelle weiter geleitet. MIG (*Message Interface Generator*) wurde benutzt um automatisch Java-Klassen für Nachrichtentypen zu generieren. MIG liest die Strukturen, die NesC benutzt, und erstellt Java-Klassen, die für automatisches Packen und Auspacken der Nachrichten sorgen.

Java-Applikation zusammen mit NesC-Teil für den Schlüssel-Aggregator und den Zähler zeigt, dass der Algorithmus mit homomorpher Verschlüsselung nach Castelluccia nicht nur in einem Simulator, sondern auch auf reale eingebetteten Geräten läuft. Da die Anzahl der realen Geräten relativ klein war, konnte mit der Java-Applikation keine Analyse für eine Große Gruppe durchgeführt werden.

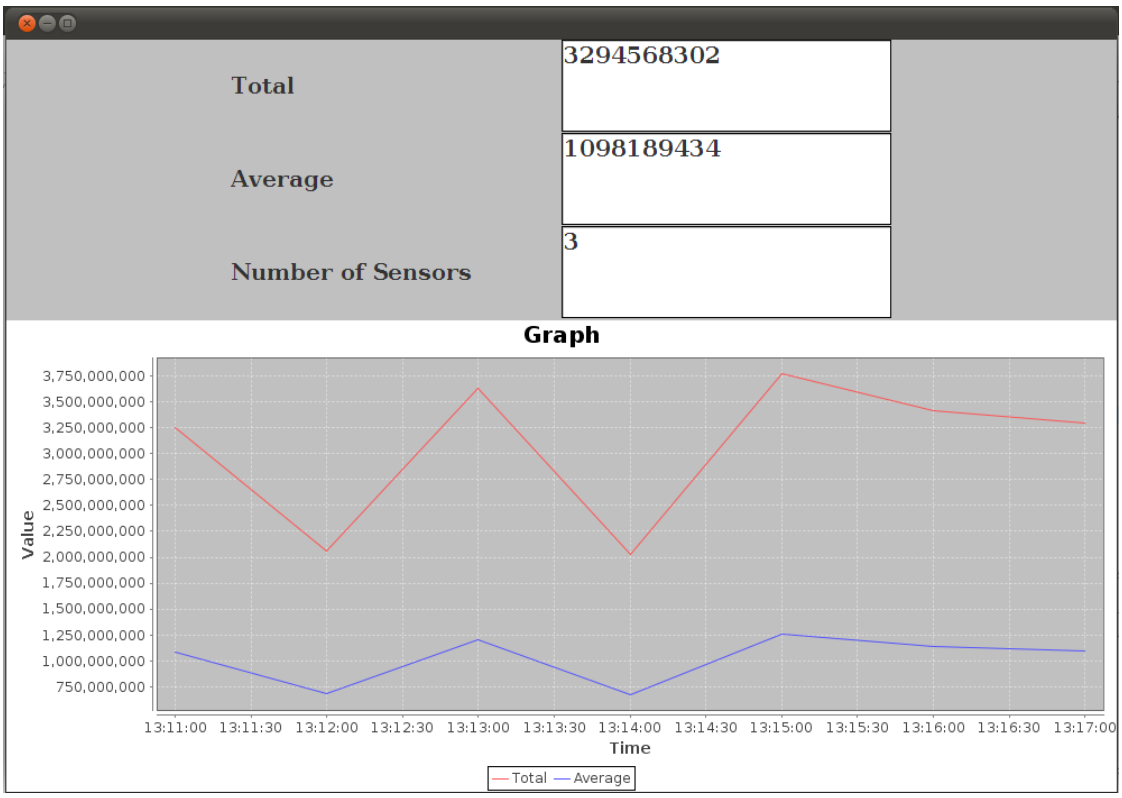

Abbildung 25: Visualisierung der Ergebnisse in Java. Messung mittels Zufallszahlengenerierung

6.6 Bewertung der Implementierung

In dieser Arbeit wurde die homomorphe Verschlüsselung nach Castellucia (Castelluccia u. a., 2005) implementiert. Die Applikationen wurden in Simulator TOSSIM und auf eingebetteten Geräten *Tmote Sky* und *TelosB* getestet. Die Applikationen haben sich als lauffähig in beiden Fällen gezeigt.

Die Simulation wurde mit einem Maximum von 1000 Knoten getestet, was völlig den gestellten Anforderungen an die Implementierung, entspricht. Das Ergebnis wurde immer richtig berechnet unabhängig von der Große der Gruppe. Dabei als Modul wurde $M = 2^{64}$ gewählt. Die Länge des Schlüssels in diesem Fall ist 64 Bit. $K \in [0 \dots 2^{64} - 1]$. Die maximale Große der Nachricht, die von Knoten gesendet werden kann, ist 28 Byte. Das heißt für den Fall, wenn der Schlüssel mit der Länge z.B. 256 Bit verwendet wird, ist ein spezielles Protokoll notwendig.

Die Hauptschwierigkeit bei der Implementierung lag daran, dass die Nachrichten in einem Radiokanal verloren gegangen sind. Der größte Verlust der Nachrichten lässt sich zwischen dem Energieversorger und dem Schlüssel-Aggregator feststellen. Die Einführung von eine Warteschlange für gesendete Nachrichten hat nur teilweise geholfen. Wenn die Anzahl der Zähler steigt, führt das zusätzlich zu eine Netzüberlastung. Mehrere Knoten versuchen die Pakete gleichzeitig zu senden und dadurch noch mehr Nachrichten verloren gehen. Um dieses Effekt zu vermeiden, wurde für einige *Timer* Zufallszahlengenerator verwendet. An der Stelle wird die Benutzung von einem speziellen Protokoll für obere Schicht benötigt. Das Protokoll muss die Nachrichten, die verloren gehen, wieder senden.

Das weitere Problem ist der Speicherbedarf von den Schlüssel-Aggregator. Schlüssel-Aggregator kann nicht alle 1000 Schlüssel speichern. Die Schlüssel der Zähler für denen eine Betsätigungstoken von dem Energieversorger angekommen sind, werden gleich addiert. Wenn der Speicher für die Schlüssel nicht ausreicht, keine Bestätigung an die Zähler gesendet wird. Die Zähler werden später die Schlüssel noch mal senden.

Der Algorithmus selbst lässt auch einige Fragen offen. Dazu gehört eine mögliche Synchronisation der Zähler und des Energieversorgers. Wenn die Summe von Messwerten periodisch berechnet werden muss, ist eine Zeitsynchronisation zwischen den Zähler und dem Energieversorger notwendig. Das Problem der Anonymität bei der Kommunikation der Zähler und des Energieversorgers muss ebenfalls zusätzlich untersucht werden. Wenn zwischen einem Zähler und einem Energieversorger einer Kanal aufgebaut wird, dann wird die Adresse von dem Zähler bekannt, damit nicht anonym. Wenn der Kanal doch die Anonymität garantiert, dann ersteht ein Problem der Verfolgung der duplizierten Nachrichten.

7 Zusammenfassung und Ausblick

7.1 Zusammenfassung der wichtigsten Ergebnissen

Im Rahmen dieser Arbeit wurden mehrere Lösungsansätze, die den Schutz der Privatsphäre in einem intelligenten Stromnetz bieten, analysiert. Dabei wurde untersucht, wie weit die theoretische Lösungen den Anforderungen des BSI Schutzprofils entsprechen. Das Vergleich hat gezeigt, das alle Lösungen verbessert werden müssen, damit sie über die volle Umsetzbarkeit für reale Bedingungen verfügen. Einige Konzepte wurden für die Implementierung als nicht geeignet bezeichnet. Die Architektur dieser Lösungen könnte nicht an die Anforderungen von dem BSI Schutzprofil angepasst werden oder die Algorithmen waren nicht für die Implementierung auf Mikroprozessoren geeignet.

Für die Implementierung auf Mikroprozessoren wurde der Algorithmus mit homomorpher Verschlüsselung nach Castelluccia gewählt. Der Algorithmus hat sich als lauffähig auf eingebetteten Geräten gezeigt. Die Applikation wurde mit einem Simulator mit 1000 Knoten getestet. Die Aggregationsergebnisse für verschiedene Testfälle unabhängig von der Gruppengröße haben immer die richtige Ergebnisse geliefert. Dabei wurde die Größe der Nachrichten und der Schlüssel so gewählt, dass einerseits die Sicherheit gewährleistet wurde anderseits die Ressourcen der eingebetteten Geräten ausreichend waren.

Für die Implementierung wurden einige Einschränkungen und Modifikationen vorgenommen. Dabei wurde ein Szenario gewählt, das Problem mit bösartigen Zähler löst und ist für einen normalen Fall mit vertrauenswürdigen Einheiten geeignet. Autorisierungsmechanismen wurden nicht implementiert. Die Gruppe wurde statisch gebildet. Die Tests mit eingebetteten Geräten wurden mit kleiner Anzahl der Knoten durchgeführt.

Die Hauptschwierigkeit bei der Implementierung war der Verlust der Nachrichten in einem Radiokanal. Sie wurde mittels eine Warteschlange für gesendete Nachrichten, zusätzliche Bestätigungsnachrichten und Zufallzahlengenerierung für einige *Timer* teilweise gemindert. Die weitere offene Fragen sind Synchronisation und anonymer Kanal zwischen den Zähler und dem Energieversorger, Speicherbedarf des Schlüssel-Aggregator, Verwaltung von großen Schlüsseln, Einführung des speziellen Protokolls für erneute Senden von verlorenen Nachrichten.

7.2 Ausblick in die zukünftige Forschung und Entwicklung

Die intelligente Messsysteme sind nicht mehr die Zukunft sondern die Gegenwart. Die politische sowie die wirtschaftliche Entwicklung fordern eine schnelle Erweiterung der intelligenten Netzen. Dafür werden klare Standarte und praktische Lösungen für die Umsetzung

der Anforderungen benötigt. Erfolgreiche Umstellung auf neue Technologie ist undenkbar ohne praktischer Umsetzung der theoretischen Grundlagen.

Das Bereich der intelligenten Messsysteme bietet unterschiedliche Möglichkeiten für weitere Forschung. Dabei stellt der Schutz der Privatsphäre, nach wie vor, im Mittelpunkt. Hier müssen die Losungen erweitert werden, die nicht nur die technische, aber auch de ethische Probleme lösen.

Um die Lücken zwischen der Theorie und dem Praxis zu schließen muss die Architektur der entwickelten intelligenten Systeme an die realen Anforderungen angepasst werden. Weitere Forschungsgebiete sind Autorisierungsmechanismen in intelligenten Netzen, Schutz des Systems gegen Angriffe, Verringerung des Kommunikationsaufwandes und Belastung des Netzes. Kryptographischer Support, Sicherheitsfunktionen, Aufteilung von kryptographischen Funktionen zwischen den Komponenten in einem intelligenten System, etc. Entwicklung von benutzerfreundlichen und ergonomischen graphischen Schnittstellen kommt auch in Frage. Für alle diese Teilaufgaben werden komplexe Modellierungen, Simulationen, Implementierungen und anschließenden Tests unter realen Bedingungen benötigt.

A Deutsch - Englisch Begriffsabbildung

Deutscher Begriff	Englischer Begriff
Authentizität	Authenticity
Dezentral steuerbare Verbraucher- oder Erzeugersysteme	Controllable Local Systems (CLS)
Energieversorger (EV)	Energy supplier (ES)
Integrität	Integrity
Intelligentes Messsystem	Smart meter system
Intelligentes Netz	Smart grid
Intelligenter Stromzähler	Smart meter
Kommunikationseinheit	Gateway
Lokales Netz (für Kommunikation)	LAN, Local Area Network
Öffentlicher Schlüssel	Publik key
Privater Schlüssel	Private key
Publik-Key-Infrastruktur	Publik key infrastructure (PKI)
Schlüssel-Aggregator (SA)	Key Aggregator (KA)
Sicherheitsmodul (z.B. eine Smart Card)	Security Module
Verbraucher	Consumer
Vertrauenswürdige dritte Partei	Trusted Third Party (TTP)
Vertraulichkeit	Confidentiality
Weitverkehrsnetz (für Kommunikation)	WAN, Wide Area Network
Wireless Personal Area Network (WPAN)	Wireless Personal Area Network (WPAN)

B Schnelle Referenz für TinyOs

TinyOs Befehl (Linux)	Bedeutung
make <platform>	kompilieren von Quellcode
make telosb	kompilieren von Quellcode (telosb)
make <platform> install	installieren
make telosb install	auf telosb installieren
motelist	alle angeschlossene Geräte anzeigen
make micaz sim	Quellcode für die Simulation kompilieren
python <pythonfile.py>	ein Skript in *Python* starten
make <platform> docs	Dokumentation generieren
make telosb docs	Dokumentation generieren (telosb)
java net.tinyos.tools.Listen -comm serial@/dev/ttyUSB0:telosb	*Listen Tool* für telosb-Gerät auf USB0 starten
java net.tinyos.sf.SerialForwarder -port 9003 -comm serial@/dev/ttyUSB1:telosb	*Serial Forwarder* für telosb-Gerät auf USB1, port 9003 starten

C Inhalt der CD-ROM

Dieser Arbeit ist eine CD-ROM beigefügt. Auf der CD-ROM befinden sich Ordner und Dateien mit folgendem Inhalt:

- Der Ordner *Bachelorarbeit* enthält dieses Dokument als PDF-Datei
- Der Ordner *Software* enthält den Prototyp, der für diese Arbeit entwickelt und getestet wurde.
- Im Ordner *Software/WSN/src* liegen die Quellecode-Dateien für TOSSIM.
- Der Ordner *Software/WSN/src/doc/nesdoc/telosb* enthält die *Graphviz* Dokumentation.
- Der Ordner *Software/PC-mote* enthält die Applikation mit der JAVA- Visualisierung.

Literatur

[TelosB 2004] *Telos (Rev B): Datasheet*. Mai 2004. – URL http://www2.ece.ohio-state.edu/~bibyk/ee582/telosMote.pdf. – zuletzt abgerufen am 20.08.2011

[ETP 2006] *European SmartGrids Technology Platform - Vision and Strategy for Europe's Electricity, European Commission*. 2006. – URL http://ec.europa.eu/research/energy/pdf/smartgrids_en.pdf. – zuletzt abgerufen am 20.08.2011

[TmoteSky 2006] *Tmote Sky: Datasheet*. Juni 2006. – URL http://www.bandwavetech.com/download/tmote-sky-datasheet.pdf. – zuletzt abgerufen am 20.08.2011

[SGSR 2009] *Smart Grid System Report, U.S. Department of Energy*. Juli 2009. – URL http://energy.gov/sites/prod/files/oeprod/DocumentsandMedia/SGSRMain_090707_lowres.pdf. – zuletzt abgerufen am 20.08.2011

[PP 2011] *Protection Profile for the Gateway of a Smart Metering System, Federal Office for information security.* Mai 2011. – URL http://www.bsi.bund.de. – zuletzt abgerufen am 20.08.2011

[EnWG Stand 2008] *Energiewirtschaftsgesetz, §21b (3a)*. Stand 2008. – URL http://lexetius.com/EnWG/21b. – zuletzt abgerufen am 20.08.2011

[Ateniese u. a. 2000] ATENIESE, G. ; CAMINISCH, J. ; JOYE, M ; TSUDIK, G: A practical and provably secure coalition-resistant group signature scheme. In: *Proceedings of the 20th Annual International Cryptology Conference in Advance Cryptography*, 2000, S. 255–270

[Bohli u. a. 2010] BOHLI, Jens-Matthias ; SORGE, Christoph ; UGUS, Osman: A Privacy Model for Smart Metering. In: *IEEE International Conferences on Communications*, May 2010

[Boneh u. a. 2006] BONEH, Dan ; GOH, Eu-Jin ; NISSIM, Kobbi: *Evaluating 2-DNF Formular on Cipherstexts*. April 2006

[Castelluccia u. a. 2005] CASTELLUCCIA, Claude ; MYKLETUN, Einar ; TSUDIK, Gene: *Efficient Aggregation of encrypted data in Wireless Sensor Networks*. 2005

[Culler 2006] CULLER, David E.: TinyOS: Operating System Design for Wireless Sensor Networks. In: *Sensor magazine*, May 2006

[Finster und Conrad 2010] FINSTER, Sören ; CONRAD, Michael: Privacy-aware Realtime Smart Metering. In: *VDE-Kongress 2010 - E-Mobility: Technologien - Infrastruktur - Märkte*, 2010

[Forte 2010] FORTE, V.J.: Smart grid at National Grid. In: *Innovative smart grid technologies (ISGT)*, Januar 2010, S. 1–4

[Garcia und Jacobs 2010] GARCIA, Flavio D. ; JACOBS, Bart: Privacy-friendly Energy-metering via Homomorphic Encryption. In: *LNCS proceedings of Security and Trust Management (**STM 2010**)*, 2010

[Gay u. a. 2003] GAY, David ; LEVIS, Philip ; CULLER, David ; BREWER, Eric: *nesC 1.1 Language Reference Manual*. 2003. – URL `http://nescc.sourceforge.net/papers/nesc-ref.pdf`. – zuletzt abgerufen am 20.08.2011

[Li u. a. 2010] LI, Fengjun ; LUO, Bo ; LIU, Peng: Secure Information Aggregation for Smart Grids Using Homomorphic Encryption. In: *1st IEEE Conference on Smart Grid Communications (**SmartGridComm ' 10**)*, October 2010

[L.Rivest u. a. 1978] L.RIVEST, Ron ; SHAMIR, Adi ; M.ADLEMAN, Leonard: A Method of Obtaining Digital Signatures and Public-Key Cryptosystems. In: *Communications of the ASM* Bd. 21, 1978, S. 120–126

[Mármol u. a. 2011] MÁRMOL, Félix G. ; SORGE, Christoph ; PETRLIC, Rnald ; UGUS, Osman ; WESTHOFF, Dirk ; PÉREZ, Gregorio M.: Privacy Enhaced Arhitecture for Smart Metering, 2011. – Eingereicht für Veröffentlichung

[Müller Februar 2011] MÜLLER, Klaus J.: Sicherheit im Smart Grid. In: *18. DFN-Workshop Sicherheit in vernetzten Systemen*, Februar 2011

[Paillier 1999] PAILLIER, Pascal: Public Key Cryptosystems Based on Composite Degree Residuosity Classes. In: *Advances in Cryptology-EUROCRYPT '99*, Springer-Verlag, 1999

[Peng und Jian-ping 2010] PENG, Zhang ; JIAN-PING, Ju: Secure Data Aggragation for Sensor Networks. In: *ISP2010 Proceedings*, 2010

[Petrlic Februar 2011] PETRLIC, Ronald: Datenschutz im intelligenten Stromnetz. In: *18. DFN-Workshop Sicherheit in vernetzten Systemen*, Februar 2011

[Xue-Song u. a. 2010] XUE-SONG, Zhoe ; LI-QIANG, Cui ; YOU-JIE, Ma: Research on smart grid technology. In: *International conference on computer and system modeling (ICCASM, 2010)*, 2010, S. v3–599–v3–603

Versicherung über Selbständigkeit

Hiermit versichere ich, dass ich die vorliegende Arbeit im Sinne der Prüfungsordnung nach §22(4) ohne fremde Hilfe selbstständig verfasst und nur die angegebenen Hilfsmittel benutzt habe.

Hamburg, 21. August 2011	
Ort, Datum	Unterschrift