

Protokoll 5

Praktikum Technische Informatik

Tas Soti, Richard Wilhelm, Naja v. Schmude

16. Juli 2008

1 “Diebstahlschutz” mit Flash-Speicher

1.1 Vorbereitung

Die die Daten in dem Flash-Speicher abgelegt werden sollen, muss man sich dazu ein wenig schlau machen. Wie bisher immer, muss man sich auch beim Flash erstmal zu diesem Dienst anmelden. Dies funktioniert mit der Funktion `adl_flhSubscribe(ascii* Handle, u16 NbObjectsRes)`. Dabei ist der Handle einfach ein String, durch den im Modul das Flashobjekt bzw. die Flashobjekte (man kann gleich mehrere Plätze auf einmal registrieren, die Anzahl wird durch `NbObjectsRes` angegeben) identifiziert sind. Die Flashobjekte, die zu einem Handle gehören, werden übrigens von 0 an hochgezählt.

Mit den Funktionen `adl_flhWrite`, `adl_flhErase` können die Objekte manipuliert werden, durch `adl_flhRead` wird der Inhalt ausgelesen. Ob ein Objekt beschrieben ist kann dann über die Methode `adl_flhExist` rausgefunden werden.

IMSI steht für *International Mobile Subscriber Identity* und bezeichnet eine eindeutige Nummer, die zur Identifizierung in GSM und UMTS Netzen herangezogen wird. Diese Nummer wird auf der SIM-Karte hinterlegt.

1.2 Aufgaben und Durchführung

- Ein AT-Kommando *AT+PROTECT* zum Aktivieren/Deaktivieren des Schutzes und zum Einstellen der Alarmrufnummer soll zur Verfügung gestellt werde.

```
void a11_apply(void)
{
    adl_flhSubscribe(a11Flash, 2);
    adl_atCmdSubscribe("AT+PROTECT", protect,
        ADL_CMD_TYPE_PARA | 0x0033); // PROTECT Befehl
        erstellen
}
```

```
void protect(adl_atCmdPreParser_t *parameter) {
    ascii a11oldIMSI[A11_IMSILENGTH];
    ascii a11newIMSI[A11_IMSILENGTH];
    ascii flashIMSI[A11_IMSILENGTH];
    ascii a11SMSNummer[A11_SMSNRLENGTH];
    ascii flashNR[A11_SMSNRLENGTH];

    // Parameter übergeben
    wm_strGetParameterString(a11oldIMSI, parameter->StrData,
        1);
    wm_strGetParameterString(a11newIMSI, parameter->StrData,
        2);
    wm_strGetParameterString(a11SMSNummer, parameter->
        StrData, 3);

    // Flash lesen
    adl_flhRead(a11Flash, 0, A11_IMSILENGTH, flashIMSI);
    adl_flhRead(a11Flash, 1, A11_SMSNRLENGTH, flashNR);

    if(wm_strcmp(a11oldIMSI,defaultIMSI) == 0) { // wir
        wollen aktivieren

        adl_atSendResponse(ADL_AT_RSP,"PROTECT_Aktivieren_mit_
            Passwort:_");
        adl_atSendResponse(ADL_AT_RSP,a11newIMSI);
        adl_flhWrite(a11Flash, 0, A11_IMSILENGTH, a11newIMSI);

        adl_atSendResponse(ADL_AT_RSP,"\nNummer_für_SMS:_");
        adl_atSendResponse(ADL_AT_RSP,a11SMSNummer);
        adl_flhWrite(a11Flash, 1, A11_SMSNRLENGTH,
            a11SMSNummer);
    }
    else if((wm_strcmp(flashIMSI, a11oldIMSI) == 0) && (
        wm_strcmp(a11newIMSI,defaultIMSI) == 0)) { // wir
        wollen deaktivieren
        adl_atSendResponse(ADL_AT_RSP,"Deaktivieren_von_
            PROTECT");
        adl_flhErase(a11Flash,0);
        adl_flhErase(a11Flash,1);
    }
    else if(wm_strcmp(flashIMSI, a11oldIMSI) == 0) { //
        richtiges Passwort, ansonsten werden Daten geändert
        adl_atSendResponse(ADL_AT_RSP,"PROTECT_Aktivieren_mit_
            Passwort:_");
        adl_atSendResponse(ADL_AT_RSP,a11newIMSI);
    }
}
```

```
    adl_flhWrite(a11Flash, 0, A11_IMSILENGTH, a11newIMSI);

    adl_atSendResponse(ADL_AT_RSP, "\nNummer für SMS: ");
    adl_atSendResponse(ADL_AT_RSP, a11SMSNummer);

    adl_flhWrite(a11Flash, 1, A11_SMSNRLENGTH,
                a11SMSNummer);
}
else {
    adl_atSendResponse(ADL_AT_RSP, "\r\nERROR");
}
}
```

Da wir zwei Speicherbereiche im Flash brauchen, eins für die IMSI, auf die immer reagiert werden soll, und die Telefonnummer, an die die Warn-SMS geschickt werden soll, registrieren wir also zunächst die beiden Objekte durch `adl_flhSubscribe(a11Flash, 2);`. Zusätzlich registrieren wir unser neues AT-Kommando. Immer wenn nun AT+PROTECT ausgeführt wird, wird die Funktion `protect` aufgerufen. Hier lesen wir zunächst mit `wm_strGetParameterString` die übergebenden Parameter ein und besorgen uns auch die aktuellen Werte von IMSI und Nummer, die im Flash liegen. Dazu benutzen wir `adl_flhRead`. Der erste Parameter gibt hier den Handle an, im zweiten ist die ID zu finden. Im dritten ist die Länge des Flashobjekts definiert und als letztes wird der Buffer angegeben, in den das Objekt zwischengespeichert werden soll.

Wenn wir alle Werte eingelesen haben, können wir nach der gegebenen Semantik des Befehls schauen, welche Aktion ausgeführt werden soll. Stimmt z.B. die `defaultIMSI` mit der gerade eingelesenen `a11oldIMSI` überein, d.h. `wm_strcmp` liefert 0 zurück, so wollen wir den Schutz aktivieren. Dies tun wir, in dem wir einfach per `adl_flhWrite` die neue IMSI (`a11newIMSI`) und die übergebende Nummer (`a11SMSNummer`) in den Flashspeicher schreiben. Dabei gibt der erste Parameter wieder den oben definierten Handle an, der zweite identifiziert die Position innerhalb des Handles, der dritte steht für die Länge und schlussendlich wird der Buffer angegeben, aus dem der Wert gelesen werden soll.

Wenn wir stattdessen den Schutz deaktivieren wollen, so müssen wir Prüfen, ob überhaupt der erste Parameter (der ja die Passwort-Funktion hat) mit der im Flash gespeicherten IMSI übereinstimmt und gleichzeitig der zweite Parameter (der den neuen Wert der IMSI angeben soll) mit der Default-Belegung übereinstimmt. Dies wird wieder per `wm_strcmp` überprüft und wenn der Ausdruck zu wahr ausgewertet wird, werden durch `adl_flhErase` die entsprechenden Flashobjekte gelöscht.

Wenn wir einfach nur das "Passwort" ändern wollen, so reicht es zu überprüfen, ob der erste Parameter mit dem im Flash gespeicherten Wert übereinstimmt. Hier werden dann wieder, wie schon bereits beim Aktivieren beschrieben, die Flash-Objekte geschrieben. Tritt keiner der erwähnten Fälle auf, ist was schief gelaufen und ein Fehler wird geworfen.

- Bei einer falschen SIM soll eine Alarm-SMS verschickt werden.

```

/* CIMI Handler */
bool a11_cimi_Handler(adl_atResponse_t *paras)
{
    ascii currentIMSI[A11_IMSILENGTH];
    ascii flashIMSI[A11_IMSILENGTH];
    ascii flashNR[A11_SMSNRLENGTH];

    // Geht nicht: wm_strGetParameterString(input, paras->
        StrData, 1);
    // weil es keinen Praefix +CIMI gibt
    wm_strcpy(currentIMSI, paras->StrData);
    wm_strRemoveCRLF(currentIMSI, currentIMSI, (u16)
        wm_strlen(currentIMSI));

    // wenn currentIMSI ungleich "OK" und kein "ERROR"
        enthaelt
    if (wm_strcmp(currentIMSI, "OK") && !help_strstr(
        currentIMSI, "ERROR"))
    {
        currentIMSI[A11_IMSILENGTH-1] = '\0'; //
            Nullterminierung

        adl_atSendResponse(ADL_AT_RSP, currentIMSI);
        adl_atSendResponse(ADL_AT_RSP, "ist IMSI der SIM-Karte
            \n\n");

        //Flash auslesen
        if(adl_flhExist(a11Flash,0) == 0) { //Objekt existiert
            nicht -> deaktiviert
            adl_atSendResponse(ADL_AT_RSP, "PROTECT ist
                deaktiviert, egal was für ne IMSI wir haben\n\n")
                ;
        }
        else {
            adl_flhRead(a11Flash, 0, A11_IMSILENGTH, flashIMSI);
            adl_flhRead(a11Flash, 1, A11_SMSNRLENGTH, flashNR);

            // testen, ob IMSI übereinstimmen
            if(wm_strcmp(flashIMSI, currentIMSI) != 0) { //
                momentane IMSI und im Flash gespeicherte sind
                ungleich! -> SMS schicken
                adl_atSendResponse(ADL_AT_RSP, "PROTECT ist
                    aktiviert und es ist ne falsche SIM eingelegt.
                    SMS wird verschickt\n\n");
                // hier SMS schicken
            }
        }
    }
}

```

```

        adl_smsSend(a11smsValue, flashNR, "Jemand_hat_
        versucht_dein_Handy_mit_ener_falschen_SIM_zu_
        benutzen", ADL_SMS_MODE_TEXT);
    }
    else {
        adl_atSendResponse(ADL_AT_RSP, "PROTECT_ist_
        aktiviert_und_es_ist_die_richtige_SIM_in_
        Verwendung!\n\n");
    }
}
}

bool a11smsHandler(ascii* smsTel, ascii* smsTimeLength,
    ascii* smsText) {
    return TRUE;
}

void a11smsControlHandler(u8 Event, u16 Nb) {
    if(Event == ADL_SMS_EVENT_SENDING_OK)
        adl_atSendResponse(ADL_AT_RSP, "\r\nSMS_erfolgreich_
        verschickt.");
}

```

Wenn wir jetzt auch eine Warn-SMS schicken wollen, wenn die falsche SIM-Karte eingelegt ist, müssen wir zunächst die IMSI abfragen, die in dem Modul gerade eingelegt ist. Der AT-Befehl AT+CIMI liefert genau die IMSI. Wir rufen also zusätzlich in der a11_apply den Befehl adl_atCmdCreate("AT+CIMI", FALSE, a11_cimi_Handler, "", NULL); auf. Der bereits angelegte Handler a11_cimi_Handler liest schon die IMSI aus und speichert sie in dem Buffer currentIMSI zwischen. Jetzt wollen wir gucken, ob überhaupt der Schutz aktiviert ist und wir die richtige SIM eingelegt haben. Das Prüfen auf Deaktivierung ist ganz einfach, da wir einfach nur mit adl_flhExist nachschauen müssen, ob das angegebene Flash-Objekt existiert. Dies funktioniert, da wir ja beim Deaktivieren den Flash löschen. Es reicht zu gucken, ob der erste Flash-Platz unseres Handles leer ist, da dort immer die IMSI eingespeichert wird. Wenn der Schutz aktiviert ist, passiert nichts weiter. Wenn er jetzt aber aktiviert ist, müssen wir zunächst den Inhalt der Flashobjekte auslesen und zwischen speichern, wieder mit adl_flhRead. Dann wird mit wm_strcmp verglichen, ob die aktuelle IMSI mit der im Flash abgelegten übereinstimmt oder nicht. Wenn sie ungleich sind, wird die SMS an die im Flash hinterlegte Nummer gesendet. Ansonsten passiert nix.

Für die SMS Funktionalität benötigen wir wieder die zwei Handler, den a11smsHandler und den a11smsControlHandler.

Ein Testlauf könnte so aussehen:

```

// Modul-Initialisierung: Gerät funktionsbereit!
262074992510671 ist IMSI der SIM-Karte
PROTECT ist deaktiviert, egal was für ne IMSI wir haben

```

```
at+protect=0000,1234,01727543559
```

```
PROTECT Aktivieren mit Passwort: 1234
```

```
Nummer für SMS: 01727543559
```

```
// Modul-Initialisierung: PIN-Eingabe abgeschlossen
```

```
// Modul-Initialisierung: Gerät funktionsbereit!
```

```
262074992510671 ist IMSI der SIM-Karte
```

```
PROTECT ist aktiviert und es ist ne falsche SIM eingelegt.
```

```
    SMS wird verschickt
```

```
SMS erfolgreich verschickt.
```

```
at+protect=1234,0000,4567
```

```
Deaktivieren von PROTECT
```

```
at+protect=0000,262074992510671,01727543559
```

```
PROTECT Aktivieren mit Passwort 262074992510671
```

```
Nummer für SMS: 01727543559
```

```
at+protect=262074992510671,1234,01727543559
```

```
PROTECT Aktivieren mit Passwort: 1234
```

```
Nummer für SMS: 01727543559
```

```
// Modul-Initialisierung: PIN-Eingabe abgeschlossen
```

```
// Modul-Initialisierung: Gerät funktionsbereit!
```

```
262074992510671 ist IMSI der SIM-Karte
```

```
PROTECT ist aktiviert und es ist ne falsche SIM eingelegt.
```

```
    SMS wird verschickt
```

```
SMS erfolgreich verschickt.
```
