Helmut Witten
Fachseminarleiter Informatik
Walther-Rathenau-Oberschule (Gymnasium)
25.04.02

Dokumentation der Unterrichtsvorbereitung

einer Unterrichtsstunde im Profilkurs Informatik

9.00 -9.45, 25.04.02

Unterrichtsreihe:

Einführung in die Kryptologie und Computersicherheit:
Das RSA-Verfahren

Inhalt:

Hinführung zum schnellen Potenzierungsalgorithmus „Square and Multiply“ durch das Verfahren der „ägyptischen“ Multiplikation

I. Analyse des zentralen Gegenstandes (Sachstruktur)

Zum Abschluss der Kryptologie-Unterrichtsreihe wird das RSA-Verfahren behandelt, dass das erste einsatzfähige Verfahren mit öffentlichem Schlüssel war und immer noch häufig eingesetzt wird (vgl. Singh, S. 329 ff.). In den letzten Stunden wurden die Prinzipien der asymmetrischen Kryptographie und deren Anwendungen (nur ein Schlüsselpaar pro Anwender, vertrauliche Kommunikation ohne Schlüsselübermittlung, digitale Signatur) erarbeitet.

Mit dem CAS-System Derive wurde in der letzten Doppelstunde demonstriert, dass man sehr leicht große Primzahlen erzeugen und multiplizieren kann. Dagegen ist es sehr schwer, ein Produkt aus sehr großen Primzahlen wieder zu faktorisieren (Der Rekord liegt zur Zeit bei einer 155-stelligen Zahl, die 1999 mit einem Aufwand von über 35 CPU-**Jahren** faktorisiert wurde, vgl. http://www.rsasecurity.com/rsalabs/ > RSA-Challenge.). Auf der Grundlage dieser Erfahrungstatsache (die im Übrigen bislang nicht bewiesen werden konnte) wurde die Einwegfunktion für das RSA-Verfahren konstruiert.

In dieser und den nächsten Stunden geht es darum, die Algorithmen zur Implementierung des RSA-Verfahrens zu verstehen und anzuwenden. Da Ver- und Entschlüsselung mit RSA auf (modularem) Potenzieren beruht, soll der Algorithmus „Square and Multiply“ zum schnellen Potenzieren erarbeitet werden. Ohne diesen Algorithmus ist eine effektive Implementierung von RSA nicht möglich (vgl. Anlage 1: Implementierung der RSA-Verschlüsselung mit Python. Dieses Programm soll erst in der folgenden Stunde eingeführt werden!).

Wenig bekannt ist, dass der Algorithmus „Square and Multiply“ sehr eng verwandt ist mit der sog. ägyptischen (oder äthiopischen oder auch russischen Bauern-) Multiplikation, die in vielen Büchern der Unterhaltungsmathematik beschrieben wird und noch heute verwendet wird (vgl. z. B. Olivastro, S. 20 ff.). Diese Methode führt das Multiplizieren auf Verdoppeln, Halbieren und Addieren zurück. Da das Verdoppeln und Halbieren mit Dualzahlen besonders einfach durchgeführt werden kann, wird diese Methode auch vielfach zur Implementierung der Multiplikation mit Computern verwendet (vgl. Bauer).

II. Passung zur Lerngruppe

Bei der Lerngruppe (11 Schüler) handelt es sich um den ersten Profilkurs Informatik an der WRO. Alle Schüler besuchen zusätzlich einen Basiskurs Informatik, in dem ihnen die Grund-

lagen der Programmentwicklung vermittelt werden. Die Gruppe arbeitet i. A. motiviert mit und zeigt sich an den im Unterricht behandelten Themen interessiert.

Ein Rahmenplan für den Profilkurs existiert noch nicht. Innerhalb der Vorbereitungsgruppe zu dem Pilotprojekt Leistungskurs Informatik wurden verschiedene Inhalte als geeignet eingeschätzt, u. a. die Kryptologie.

Aus der vorangegangenen Reihe zur multiplikativen Verschlüsselung ist den Schülern die Analyse von vorgegebenen Algorithmen vertraut (erweiterter euklidischer Algorithmus bzw. Algorithmus von Berlekamp). Insofern könnte der „Square and Multiply"-Algorithmus in ähnlicher Weise analysiert werden.

Ich habe mich aber entschlossen, den „Umweg" über die ägyptische Multiplikation zu wählen, da sich so für die Schüler mehr Möglichkeiten des entdeckenden Lernens ergeben. Darüber hinaus bietet dieses Multiplikationsverfahren überraschende Anwendungen in der Computertechnik (s. o.).

Für die Stunde sind folgende didaktische Reduktionen vorgesehen:

- Auf die modulare Reduktion kann verzichtet werden, da sie vom Verständnis des algorithmischen Kerns ablenken würde. Außerdem müssten die Rechenregeln der modularen Multiplikation bzw. Potenzierung hergeleitet, zumindest aber plausibel gemacht werden.
- Die Verbindung zum Dualsystem und zum Computer soll in dieser Stunde nur dann gezogen werden, wenn die Schüler diese Verbindung selber entdecken oder mit leichter Einhilfe finden. Da die Dualzahlen in den beiden Basiskursen eingeführt wurden ist mir der Stand des Vorwissens auf diesem Gebiet unklar. Alternativ erhalten die Schüler einen „Forschungsauftrag" als fakultative Hausaufgabe: „Was könnte die ägyptische Multiplikation mit dem Computer zu tun haben?"
- Eine Implementierung der ägyptischen Multiplikation soll aus Zeitgründen nicht erfolgen. Vielmehr wird ein fertiges Pythonprogramm vorgelegt, das durch die Schüler analysiert wird (vgl. Anlage 2).

Die Übertragung der ägyptischen Multiplikation auf den Algorithmus zum schnellen Potenzieren sollen die Schüler dann weitgehend selbstständig vollziehen. Falls durch die Behandlung des Algorithmus im Dualsystem Zeitknappheit entstanden sein sollte, kann die Bearbeitung dieser Aufgabe auch alternativ als Hausaufgabe oder in der nächsten Stunde erfolgen.

III. Lern- und Erfahrungsmöglichkeiten (Zielbestimmung)

Nach den obigen Überlegungen ergibt sich als ***Grobziel*** der Stunde:

Die Lernenden kennen den Algorithmus zur „ägyptischen" Multiplikation, können ihn anwenden und seine Korrektheit nachweisen.

Nach einem „informierenden Einstieg" (Grell, S. 134 ff.) wird den Schülern die Geschichte vom Oberst, der in Äthiopien sieben Stiere zum Preis von 22 Maria-Theresien-Taler kaufen wollte, vorgelesen (Olivastro, S. 20 f.). An der Tafel wird das Vorgehen des Priesters zur Berechnung des Produkts notiert. Die Lernenden erhalten die Aufgabe, das Ergebnis zu überprüfen und das Verfahren mit anderen Faktoren durchzuführen.

Feinziel 1: Die Lernenden kennen das Verfahren zur „ägyptischen" (bzw. „äthiopischen") Multiplikation und können es anwenden.

Anschließend werden die Schüler aufgefordert, nach einer Begründung für die Korrektheit des Verfahrens zu suchen. Hier erwarte ich mindestens, dass dies mit Hilfe des Distributivgesetzes gelingt.

Feinziel 2: Die Schüler können die Korrektheit des Verfahrens nachweisen.

Evtl. „entdecken“ die Schüler (ggf. mit Einhilfe) den Zusammenhang mit dem Dualsytem und erkennen, dass die „ägyptische“ Multiplikation im Dualsystem besonders einfach ist. Der Lehrer informiert darüber, dass dieses Verfahren deshalb heute noch bei Computern Anwendung findet. Sollte sich die Lerngruppe damit schwertun, wird dieses Problem als „Forschungsauftrag“ (fakultative Hausarbeit) vergeben.

Feinziel 3 (Eventualziel alternativ zu Feinziel 4): Die Lernenden verstehen, dass die „ägyptische“ Multiplikation im Dualsystem besonders einfach ist und daher heute noch bei der Computermultiplikation angewendet wird.

Nach diesen Vorbereitungen erhalten die Schüler das Python-Programm zur „ägyptischen“ multiplikation mit den Aufträgen, sich erstens von der Korrektheit des Programms zu überzeugen und zweitens das Programm so zu modifizieren, dass man damit Potenzieren statt Multiplizieren kann. Bei wenig Zeit kann diese Aufgabe auch als Hausaufgabe gestellt oder in der nächsten Stunde bearbeitet werden.

Feinziel 4 (Eventualziel alternativ zu Feinziel 3): Die Schüler können den Algorithmus zur „Ägyptischen“ Multiplikation zu dem Algorithmus zur schnellen Potenzierung („Square and Multiply“) modifizieren.

Literatur:

Bauer, F. L.: Multiplikation und Dualsystem. In: Informatik-Spektrum Band 17 Heft 4 (August 1994), S. 245 ff.

Grell, J.; Grell, M.: Unterrichtsrezepte. Weinheim und Basel: Beltz ^{2}1999.

Olivastro, D.: Das chinesische Dreieck – Die kniffligsten mathematischen Rätsel aus 10000 Jahren. München: Droemersche Verlagsanstalt 1995 (Lizenzausgabe Zweitausendeins).

Singh, S.: Geheime Botschaften – die Kunst der Verschlüsselung von der Antike bis zu den Zeiten des Internet. München, Wien: Hanser 2000.

Anlage 1

```
# Ver- und Entschlüsseln nach RSA mit gegebenem n,d und e

#
# H. Witten 24.4.2002

from string import upper

def modpot(x,y,n):
    '''
    Schnelles Potenzieren nach der Methode "square and multiply".
    Bei jedem Schritt wird modulo n reduziert! '''
    pot=1
    while y > 0:
        if  y % 2 == 1:         # Falls Exponent ungerade:
            pot = (pot * x) % n # Multiplizieren!
            y = y - 1           # Nun ist der Exponent wieder gerade
        else:
            x = (x * x) % n     # Quadrieren der Basis und
            y = y / 2           # halbieren des Exponenten
    return pot

# Schlüssel für RSA (ggf. ändern)
n = 55
e = 27
d = 3

while 1: # Endlosschleife
    Richtung = raw_input('V)erschlüsseln, E)ntschlüsseln oder F)ertig? ')
    Richtung = upper(Richtung)
    if Richtung[0] == 'V':
        print 'Nachricht (Zahl) < ', n,'eingeben: ',
        m = input()
        print 'Verschlüsselte Nachricht:',modpot(m,e,n)
    elif Richtung[0] == 'E':
        print 'Chiffre (Zahl) < ', n,'eingeben: ',
        c = input()
        print 'Entschlüsselte Nachricht:',modpot(c,d,n)
    else: break
print; print 'Fertig!'; print
```

Anlage 2:

```
# Multiplizieren nach der ägyptischen Methode
#
# H. Witten, 24.4.2002

def multi(x,y):
    prod = 0
    while y > 0:
        if  y % 2 == 1:     # Zweiter Faktor ist ungerade
            prod = prod + x # Ein "gutes" Haus: Addieren!
            y = y - 1       # Nun ist er wieder gerade!
        else:
            x = x + x       # Erster Faktor wird verdoppelt
            y = y / 2       # Zweiter Faktor wird halbiert
    return prod

a = input('Bitte den ersten Faktor eingeben: ')
b = input('... und nun noch den zweiten: ')
print
a_mal_b = multi(a,b)
print
print a,'*',b,'=',a_mal_b
```