Tanda Tangan Digital dalam Mencegah Pengatasnamaan Pengiriman SMS dengan Autentikasi IMEI

David Soendoro / 13507086
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
[1]if17086@students.if.itb.ac.id

***Abstraksi*—Tanda tangan digital atau *digital signature* dengan berbagai metode seperti RSA atau lainnya telah diakui merupakan salah satu algoritma pemastian kebenaran sebuah dokumen dikirim oleh seseorang. Seperti tanda tangan pada sebuah dokumen, tanda tangan digital akan dimiliki unik oleh setiap orang. IMEI sebagai penanda unik dari setiap perangkat mobile sesungguhnya dapat digunakan sebagai parameter autentikasi tanda tangan pada transfer data lewat jaringan seluler. Salah satu transfer data jaringan seluler itu adalah SMS atau Short Message Service. SMS terdiri dari header dan isi, dengan mengetahui hal tersebut kita dapat menggunakan isi ini menjadi penandatangan digital bila dicampurkan dengan IMEI. Adanya tanda tangan digital akan mencegah terjadinya spamming juga SMS yang tidak tepat sasaran. Pada tulisan ini, penulis akan mencoba mengimplementasikan ide tanda tangan digital dengan algoritma RSA sebagai pencegah spamming dan mencegah salah kirim SMS.**

***Index Terms*—Digtal Signature, IMEI, RSA, SMS.**

I. PENDAHULUAN

Perkembangan dunia perangkat mobile telah memasuki kehidupan kita sehari-hari dengan sangat cepat. Bahkan tingkat penjualan perangkat mobile seperti cellphone, atau smartphone sangatlah cepat. Mengetahui pangsa pasar yang begitu besar banyak oknum yang kurang bertanggung jawab yang menggunakan perangkat mobile ini untuk melakukan spamming dalam mempromosikan produknya dan mengganggu para pengguna. Sebenarnya untuk menanganinya provider dapat menggunakan sebuah metode penandatanganan digital terhadap segala SMS yang dikirimkan oleh pengguna. SMS spamming biasanya dilakukan oleh robot dan bukannya dari ponsel seperti metode biasa, sehingga SMS dari mereka tidak memiliki IMEI yang hanya dimiliki perangkat mobile. Dengan mengambil kunci privat dari tujuan SMS yang tertera pada header SMS, kita dapat memastikan tanda tangan digital yang dibuat secara *background* saat pengguna mengirimkan pesan singkat tersebut.

Menurut saya, inovasi ini merupakan sebuah kontribusi nyata yang sangatlah penting karena dalam dunia telekomunikasi modern ini, sebuah kenyamanan merupakan tuntutan yang sangat besar dari masyarakat pengguna perangkat mobile. Program yang akan dicoba dibuat memang dibuat untuk klien, namun sesungguhnya program ini sangat berguna bagi provider (pada server) untuk mencegah spamming dan salah pengiriman, karena dilakukan pengecekan tanda tangan digital dengan mengambil kunci privat target pengirim.

II. FORMAT SMS

Format SMS yang akan dibahas pada makalah ini adalah format pengiriman dan penerimaan SMS antar perangkat mobile GSM yang berasal dari provider yang sama. SMS, seperti perangkat telekomunikasi lainnya memiliki kesepaktan khusus dalam mengolah pesan-pesannya. Seringkali disebut AT-commands, perintah-perintah ini dapat kita gunakan setelah menyambungkan perangkat telekomunikasi data dengan komputer.
Secara teori, format SMS yang dikirim dan diterima oleh telepon seluler adalah seperti ini

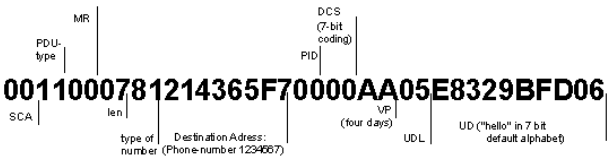

Figure 1 – format SMS secara teoritis

Penjelasan:

- SMS sebenarnya terdiri dari 2 bagian besar yaitu SMS header dan SMS body
- Penandatanganan digital dengan IMEI akan ditambahkan pada SMS body

Karena header SMS pada tulisan ini tidak banyak dipakai maka penjelasan mengenai header SMS tidak akan terlalu dalam, melainkan hanya berbagai macam singkatan dan fungsi singkatnya:

SCA PDU Type	Service Centre Address – information element Protocol Data Unit Type	Telephone number of the Service Centre
MR	Message Reference	Sucessive number (0..255) of all SMS-SUBMIT Frames set by the M20
OA DA	Originator Address Destination Address	Address of the originating SME Address of the destination SME
PID	Protocol Identifier	Parameter showing the SMSC how to process the SM (as FAX, Voice etc)
DCS	Data Coding Scheme	Parameter identifying the coding scheme within the User Data (UD)
SCTS VP	Service Centre Time Stamp Validity Period	Parameter identifying time when the SMSC received the message Parameter identifying the time from where the message is no longer valid in the SMSC
UDL	User Data Length	Parameter indicating the length of the UD-field
UD RP	User Data Reply Path	Data of the SM Parameter indicating that Reply Path exists
UDHI	User Data Header Indicator	Parameter indicating that the UD field contains a header
SRI	Status Report Indication	Parameter indicating if the SME has requested a status report
SRR VPF	Status Report Request Validity Period Format	Parameter indicating if the MS has requested a status report Parameter indicating whether or not the VP field is present
MMS	More Messages to Send	Parameter indicating whether or not there are more messages to send
RD MTI	Reject Duplicate Message Type Indicator	 Parameter describing the message type 00 means SMS-DELIVER 01 means SMS-SUBMIT

Figure 2 – format-format yang ada pada SMS header

III. IMEI

IMEI merupakan singkatan dari International Mobile Equipment Identity, merupakan sebuah rangkaian 15 digit angka desimal yang digunakan untuk mengidentifikasi perangkat seluler. Kehebatan IMEI adalah tidak ada duplikasi yang terjadi . IMEI sendiri sebenarnya terdapat dua versi, yakni versi teoritis, yaitu IMEI untuk perangkat keras dan IMEI versi peranti lunak atau lebih sering disebut IMEISV atu IMEI - Software Version.

IMEI biasa ditulis sebagai 14 digit untuk IMEI hardware ditambah 1 digit checksum untuk mempercepat pemeriksaan dan 16 digit untuk IMEISV ditambah 1 digit untuk checksum, formatnya adalah sebagai berikut

$$AA - BBBBBB - CCCCCC - D - EE$$

AA	BBBBBB
Reporting Body Identifier, indicating the GSMA-approved group that allocated the model TAC	The remainder of the TAC
CCCCCC	D
Serial sequence of the model	Luhn check digit of the entire number (or zero)

EE
Software Version Number (SVN).

Figure 3 - Format IMEI standard

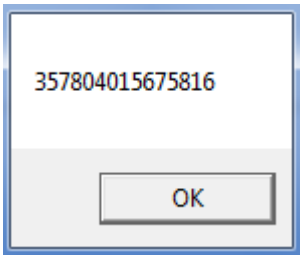

Figure 4 - IMEI SE-K530i yang diambil dengan program

IMEI bukanlah nomor sembarangan, IMEI merupakan suatu runtutan yang dapat diperiksa kembali kebenarannya, yaitu dengan menggunakan algoritma Luhn, yaitu dengan:

1. Kalikan 2 semua bilangan yang berada pada urutan genap
2. Bila jumlah dikali 2 lebih besar dari 9 maka jumlahkan bilangan-bilangan tersebut, contoh: 8 * 2 = 16 maka ubah menjadi 1 + 6 = 7
3. Jumlahkan bilangan pada seluruh urutan
4. Checksum adalah bilangan yang membuat jumlah pada nomor 3 dapat dibagi 10.

Contoh kasus:
IMEI:

3	5	7	8	0	4	0	1	5	6	7	5	8	1	?
3	10	7	8	0	4	0	2	5	12	7	10	8	2	?
3	1	7	8	0	4	0	2	5	3	7	1	8	2	51
														6

Maka IMEI pada contoh di atas adalah benar karena checksumnya adalah 6.

IV. DIGITAL SIGNATURE DENGAN RSA

Skema tanda tangan digital adalah sebuah skema matematis yang bertujuan memastikan sang pengirim pesan, tanpa tanda tangan digital, pihak-pihak yang kurang bertanggung jawab dapat mengatasnamakan orang lain untuk memeroleh keuntungan pribadi, untuk itulah diciptakan metode tanda tangan digital. Prinsip tanda tangan digital pada umumnya adalah seperti ini.

Prinsip tanda tangan digital umumnya:

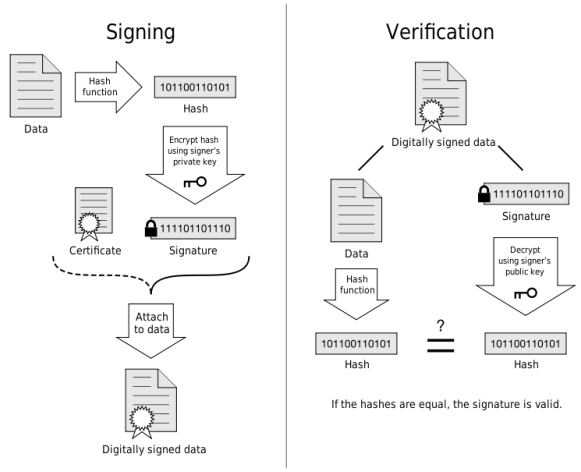

Figure 5 – prinsip tanda tangan digital

Langkah-langkah:

1. Pesan dihash
2. Hasil hash dienkripsi dengan algoritma RSA dengan kunci privat dan dikirim beserta sertifikat dari CA diselipkan di dalamnya.
3. Pada saat pesan diterima pesan dipecah menjadi 2 yaitu tanda tangan dan pesan.
4. Tanda tangan didekripsi sedangkan pesan dihash, keduanya harus menjadi nilai yang sama.
5. Apabila nilai sama baka autentikasi berhasil, jika tidak maka autentikasi gagal.

Metode ini sangat berguna dalam mengatasi spam yang seringkali mengganggu jaringan dan pengguna. Dalam dunia pesan singkat seluler, digital signature ternyata sangat dibutuhkan, salah satu contoh nyata adalah dalam memastikan pengirim SMS untuk anti-spam. Prinsip penandatanganan digital dalam telepon seluler adalah seperti ini

Prinsip tanda tangan digital pada telepon seluler:

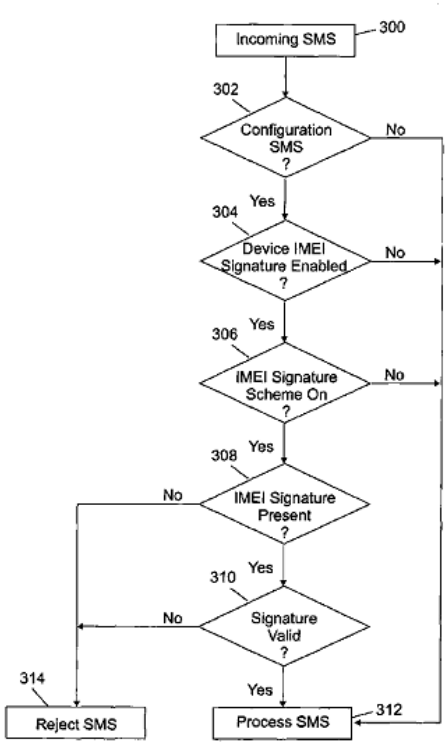

Figure 6 – prinsip tanda tangan digital pada telepon seluler

Penambahan tanda tangan digital pada pengiriman pesan singkat akan memperpendek jumlah karakter yang dapat dikirim dalam tiap smsnya, oleh karena itu tanda tangan digital hanya akan dilakukan apabila pengguna melakukan konfigurasi yang mana dapat dituliskan pada header SMS yang belum terpakai di mana akan dibuat kesepakatan antar provider dan user bit penanda akan digunakan atau tidaknya sebuah tanda tangan digital.

Poin pokok yang akan kita sorot dalam tulisan ini adalah saat tanda tangan IMEI ada dan kita harus mengelolanya, bagaimana mengenkripsi tanda tangan tersebut dan mengautentikasikannya kembali. Autentikasi akan dilakukan dengan menggunakan IMEI sebagai kunci privat dan IMEI hasil dekripsi akan diperiksa kembali kebenaranya.

Pada implementasi makalah ini, penulis menggunakan algoritma RSA sebagai metode penandatanganan digital. Prinsip RSA sendiri adalah seperti ini.

1. Pilih dua bilangan prima yang besar, sebut saja *p* dan *q*, dan jangan sama angkanya. (rahasia)
2. Hitung $n = pq$. (tidak rahasia)
3. Hitung $\phi(n) = (p - 1)(q - 1)$. (rahasia)
4. Pilih satu bilangan *e* yang *coprime* terhadap $\phi(n)$. *Coprime* atau relatif prima artinya faktor pem-bagi terbesarnya adalah 1. Bilangan *e* dan *n* ini dipublikasikan sebagai *kunci publik*, tidak rahasia.
5. Hitung kunci privat *d* yang memenuhi persamaan $ed \equiv 1 \ (mod \ \phi(n))$. Cara lainnya adalah:

$$d = \frac{1 + k\,\phi(n)}{e}$$

Dengan mencoba-coba nilai k mulai dari 1, 2, 3, …, akan terdapat d yang bulat (bukan pecahan). Nilai d ini adalah *kunci privat* dan tentu saja bersifat rahasia.

Cara enkripsi adalah sebagai berikut. Bagi pesan dalam blok-blok $m1$, m2, dan deterusnya. Setiap blok menyatakan nilai angka di antara 0 dan $n - 1$. Kemudian blok cipherteks didapat dari

$$c_i = m_i{}^e \bmod n$$

Dekripsi menggunakan rumus yang sama, hanya kita menggunakan kunci privat d.

$$m_i = c_i{}^d \bmod n$$

Namun untuk tanda tangan digital, kita mempertukarkan e dan d pada kasus enkripsi dan dekripsi. e dan d memang berkoresponden, namun mereka hanyalah sepasang angka. Maka tidak masalah bila kita mempertukarkan e untuk dekripsi dan d untuk enkripsi.

Di atas telah disebutkan bahwa dalam penandatanganan digital dibutuhkan kunci publik dan kunci privat. IMEI seperti telah disebutkan sebelumnya akan digunakan dalam penandatanganan ini. Kode IMEI akan menjadi kunci publik dan akan dimasukkan pada akhir pesan dengan format tertentu sehingga penerima dapat mengautentikasi pesan tersebut. Setelah pesan berhasil diautentikasi, kebenaran IMEI akan diperiksa dengan menghitung checksum IMEI tersebut dan apabila telah melewati kedua uji tersebut barulah pesan akan diterima secara sepenuhnya.

V. IMPLEMENTATION

Dalam implementasi, penjelasan dan pelaksanaan akan dibagi menjadi 3 tahap. Tahap yang pertama yaitu mengambil data dari telepon seluler, tahap kedua adalah melakukan tanda tangan digital dan mengirimkannya ke provider, sedangkan tahap ketiga adalah simulasi provider(diwakilkan oleh program komputer) untuk mengautentikasi pesan.

Pengambilan data

Data yang harus diambil adalah data IMEI, pada percobaan ini kita akan menggunakan AT commands untuk mengambil nomor IMEI pada perangkat lunak, berikut adalah kode AT commands untuk mengambil IMEI pada telepon seluler dan hasil pengambilannya:

```
AT+CGSN – IMEI
AT+CGMR – IMEISV
```

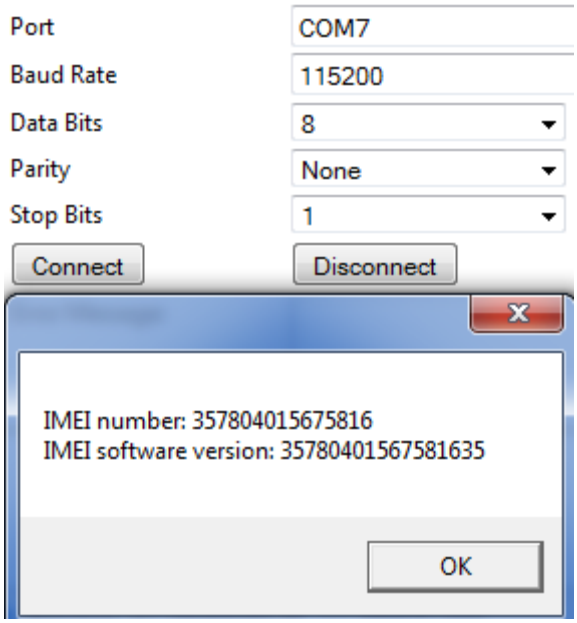

Figure 7 – Nomor IMEI dan IMEISV

Simulasi user

Setelah nomor IMEI didapatkan, langkah selanjutnya adalah melakukan pencarian kunci privat dari IMEI tersebut, berikut adalah kode yang digunakan dalam melakukan pembangkitan kunci privat:

```
Random rand = new Random();
e = _e;
d = BigInteger.Zero;
p = new BigInteger(e.BitLength / 2, rand);
q = BigInteger.ProbablePrime(e.BitLength / 2,
rand);
if
(e.Gcd(totientFunction()).CompareTo(BigInteger.O
ne) == 0)
        d = e.ModInverse(new
BigInteger(totientFunction().ToString()));

while
(d.Gcd(totientFunction()).CompareTo(BigInteger.O
ne) != 0)
{
        p = new BigInteger(e.BitLength / 2,
rand);
        q = BigInteger.ProbablePrime(e.BitLength
/ 2, rand);
        if
(e.Gcd(totientFunction()).CompareTo(BigInteger.O
ne) == 0)
        {

        Console.WriteLine(e.Gcd(totientFunction()
));
                d = e.ModInverse(new
BigInteger(totientFunction().ToString()));
        }
}

n = p.Multiply(q);
```

Figure 8 – pembangkitan kunci pivat dan modulo dari kunci public

Berikut adalah hasil pembangkitan dari IMEI yang dihasilkan sebelumnya:

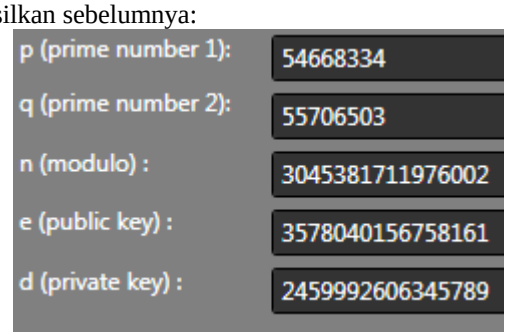

Figure 9 – karena kunci publik harus merupakan bilangan ganjil maka ditambahkan angka 1 di belakang kode IMEI

Setelah mendapatkan kunci-kunci yang dibutuhkan, tambahkan ke depan pesan singkat, kunci publik yang akan di kirim beserta tanda tangan digital, sehingga pesan menjadi:

```
[pesan][tanda tangan digital][kunci publik]
```

Berikut adalah contoh pesan yang akan dikirim oleh program:

```
Halo dunia! <ds>402c54ab9bcb8
9768bf78ec8af 4962be5e0f8a2
80008f36db6d5 </ds> 3578040156758161
```

Simulasi provider

Setelah pesan kita kirim dengan format seperti yang telah dituliskan di atas, provider akan menerima pesan dan melakukan parsing terhadap tanda tangan digital dan kunci publik yang merupakan kode IMEI.
Berikut adalah algoritma pengautentikasian tanda tangan:

```
public String Decrypt(String cipherText)
{
        String plainText = "";
        int blockLength = n.ToString().Length;
        String[] cipherBlock = cipherText.Split('
');

        for (int i = 0; i < cipherBlock.Length -
1; i++)
        {
                BigInteger numbers = new
BigInteger(cipherBlock[i], 16);
                //Console.WriteLine(numbers);
                plainText +=
decryptFunction(numbers);
        }

        return plainText;
}

private BigInteger decryptFunction(BigInteger
numbers)
{
        BigInteger output = null;
        output = numbers.ModPow(new
BigInteger(d.ToString()), new
BigInteger(n.ToString()));

        return output;
}
```

```
mySHA1 sha = new mySHA1();
//SHA1 sha = new SHA1CryptoServiceProvider();
result = sha.ComputeHash(data);

Org.BouncyCastle.Math.BigInteger bi = new
Org.BouncyCastle.Math.BigInteger(BitConverter.To
String(result).Replace("-", ""), 16);

hashOutput = bi.ToString();

//hashOutput is the hash result from the data
//now decrypting signature
String signature = textBox_signatureIn.Text;
RSA.RSA rsa = new RSA.RSA(public_key,
modulo_public);

if (hashOutput.Replace("0", "") !=
rsa.Decrypt(signature).Replace("0", ""))
{
        MessageBox.Show("Signature is not
valid!", "paperless - warning input error",
MessageBoxButton.OK, MessageBoxImage.Warning);
}
else
{
        MessageBox.Show("Signature valid!",
"paperless - warning input error",
MessageBoxButton.OK, MessageBoxImage.Warning);
}
```

Figure 10 - algoritma RSA dalam mengautentikasi tanda tangan yang hasilnya akan dicocokkan dengan SHA-1 dari data yang dikirim.

Berikut adalah hasil program ketika dijalankan untuk mengautentikasi pesan:

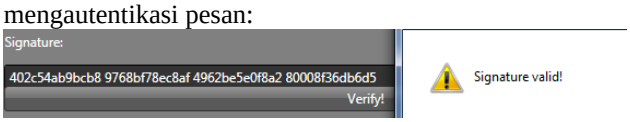

Figure 11 – autentikasi tanda tangan

Setelah melakukan autentikasi terhadap tanda tangan, program akan melanjutkannya dengan autentikasi terhadap IMEI. Autentikasi ini dilakukan untuk memastikan bahwa pengirim adalah pemilik telepon seluler yang sah yaitu dengan memeriksa checksum dari IMEI tersebut.

```
int cnt = IMEI.Length;
int sum = 0;
int checksum = 0;
int validator = 0;
int[] arrayIMEI = new int[cnt];

for (int i = 0; i < cnt; i++)
{
        if(i == cnt - 1)
```

```
        {
                checksum =
int.Parse(IMEI.Substring(i,1));
        }
        else
        {
                arrayIMEI[i] =
int.Parse(IMEI.Substring(i,1));
                if (i % 2 != 0)
                {
                        arrayIMEI[i] +=
arrayIMEI[i];
                        if(arrayIMEI[i] > 9)
                                arrayIMEI[i] =
(arrayIMEI[i] / 10) + (arrayIMEI[i] % 10);
                }
                sum += arrayIMEI[i];
        }
}
validator = 10 - (sum % 10);

return checksum == validator;
```

Figure 12 – Algoritma pengecekan IMEI

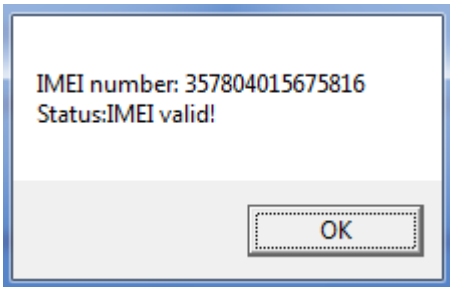

Figure 13 – Hasil run pengecekan IMEI

VI. Kesimpulan dan Saran

Setelah membuat paper tentang IMEI sebagai parameter tanda tangan digital pada pesan singkat melalui telepon seluler saya mendapati bahwa penggunaan tanda tangan digital sangatlah berguna, namun perlu diingat juga bahwa dalam penulisan pesan singkat tempat sangatlah terbatas sehingga penambahan tanda tangan dan kunci public akan sangat memberatkan pengguna.

Penulis sangat berharap ada yang melanjutkan penelitian tentang masalah ini terutama dalam mengurangi atau bahkan bila bisa memasukkan parameter tanda tangan digital maupun kunci public yang sekaligus adalah IMEI ini ke dalam header SMS atau setidaknya mengurangi panjangnya.

Selain itu, setelah mengerjakan makalah ini juga saya menemukan suatu hal yang menarik yaitu pembangkitan kunci privat melalui kunci public saja dan sebaliknya. Merupakan sebuah langkah kecil dalam dunia kriptanalis karena dari satu macam kunci public bisa dibangkitkan berbagai macam kunci privat dan sebaliknya, namun saya percaya masih ada kemungkinan untuk memperkirakan dengan lebih akurat kunci publik dan privat dari sebuah algoritma RSA. Sekian kiranya tulisan ini penulis cukupkan sampai di sini, sebelum benar-benar mengakhirinya penulis ingin membagikan suatu kata yang sangat berarti bagi teman-teman sesama pelajar,

”That's one small step for a man, a giant leap for mankind”

– Neil Armstrong –

VII. Ucapan Terima Kasih

David Soendoro berterima kasih atas segala bantuan yang didapat dari teman-teman kuliah dalam meminjamkan sarana dalam mencari data hingga akhirnya makalah ini dapat tersusun.

Terima kasih juga penulis tujukan bagi bapak Rinaldi Munir M.T., dosen mata kuliah Kriptografi yang telah mengajarkan materi sebagai dasar pembuatan makalah ini.

References

ActiveXperts SMS and MMS Toolkit, http://www.gsmfavorites.com/documents/sms/packetformat/

Marmigere, Gerard – United State Patent, *System and Method for Digital Signature Authentication of SMS Messages*

Munir, Rinaldi IF5034 – Kriptografi, Bandung 2004

NN - IMEI Allocation & Approval Guidelines

Pernyataan

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 17 Mei 2010

ttd

David Soendoro / 13507086