Nama: Farhan Nafis Rayhan
NIM: 13522037

Laporan Singkat Mesin Turing Kriptografi RSA

Mesin turing adalah model komputer teoritis berbentuk *finite automata* dengan *tape*. Model ini diciptakan oleh ilmuwan berkebangsaan inggris bernama Alan turing pada tahun 1936. Mesin turing dapat merepresentasikan model ideal dari *CPU (Central Processing Unit)* yang bertugas untuk mengatur manipulasi data dengan bantuan penyimpanan berurut untuk memori. Dalam teori bahasa formal, mesin turing merupakan automata yang cukup kuat untuk memproses *grammar* tak terbatas *(unrestricted grammar)*, yang merupakan kelas *grammar* paling umum dalam hirarki Chomsky. Mesin turing juga dikenal melalui konsep *Turing Completeness*, dimana model komputasi apapun yang dapat mensimulasikan mesin turing secara teoritis dapat pula menyelesaikan semua persoalan yang mampu diselesaikan oleh komputer. Inilah alasan mengapa hampir semua bahasa pemrograman merupakan *Turing Complete*.

Model ini memiliki sebuah pita dengan panjang tak hingga. Pita dibagi menjadi tak hingga buah sel dengan tepat satu simbol dalam setiap sel. Mesin Turing dapat dimasukkan sebuah input yang merupakan *string* simbol yang terdefinisi pada alfabet input. Awalnya pita akan terisi oleh input saja dan seluruh sel lainya merupakan simbol kosong *(Blank)*. Mesin Turing juga memiliki *finite control* atau *head* yang bisa berada dalam salah satu *state* yang terdefinisi. *Head* akan membaca sebuah simbol dalam pita, lalu merubah simbol tersebut menjadi simbol lainya (mungkin sama), dan bergeser tepat satu sel ke kiri maupun kanan. *Head* akan terus berpindah sampai ia memasuki *accepting state*, atau sampai ia bertemu transisi yang tidak terdefinisi. Proses ini dinamakan dengan *halt*, dimana pada tahap ini output dari Turing Machine akan diperoleh.

Sebuah Turing Machine didefinisikan oleh *7-tuple* berupa

$$M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$$

Dengan rincian sebagai berikut:

- Q : Himpunan dari *state* terdefinisi untuk *finite control*
- Σ : Himpunan dari input simbol, simbol yang hanya dapat digunakan sebagai input dan subhimpunan dari Γ.
- Γ : Himpunan dari seluruh simbol *tape*.
- δ : Fungsi transisi yang digunakan Mesin Turing, dengan definisi

$$\delta(q, X) = (p, Y, D)$$

  1. q : *State* sebelum transisi
  2. X : Simbol yang terbaca
  3. p : *State* setelah transisi
  4. Y : Simbol yang ditulis pada pita
  5. D : Arah perpindahan *head* setelah transisi. Direpresentasikan oleh R untuk kanan dan L untuk kiri

- q_0 : *Start State*, yaitu *state* awal dimana Mesin Turing mulai.
- B : Simbol *Blank*, merupakan simbol dalam Γ namun bukan dalam Σ.
- F : Himpunan *Accepting State*, atau subhimpunan dari Q

Secara umum, cara kerja Mesin Turing adalah sebagai berikut:

1. Awalnya, input dimasukkan ke dalam tape. Seluruh sel lainya berupa *Blank*
2. *Finite Control* bermula pada *initial state* dan *head* membaca state paling kiri dari input.
3. Menggunakan fungsi transisi, Mesin Turing akan:
   a. Menulis simbol baru pada pita.
   b. Menggeser *head* ke kiri atau kanan.
   c. Masuk ke *state* baru.
4. Langkah 3 diulang sampai terjadi *halt* akibat salah satu diantara 2 kasus berikut:
   a. Mesin Turing masuk ke dalam salah satu *accepting state*.

b. Mesin Turing menemukan transisi yang tidak terdefinisi.

5. Apabila Mesin Turing memiliki *accepting state*, sebuah bahasa bisa diterima mesin apabila ia masuk pada *accepting state*. Sedangkan apabila ia tidak punya *accepting state*, ia termasuk *acceptance by halting*.

Dalam proyek ini, penulis mengimplementasikan algoritma kriptografi RSA menggunakan mesin Turing. Algoritma RSA sendiri mengandalkan 2 buah prima besar P dan Q. Dimana $N = PQ$ juga $M = \varphi(N)$ lalu dipilih bilangan E dan D sehingga $ED = 1 \, mod \, M$. Maka enkripsi dan dekripsi dilakukan dengan cara berikut.

$$Plaintext^{E} = Ciphertext \, mod \, N$$
$$Ciphertext^{D} = Plaintext \, mod \, N$$

Untuk pemetaan permasalahan RSA menjadi Mesin Turing dalam tugas ini cukup banyak dilakukan penyederhanaan. Hal ini diperlukan karena batasan dari performa Mesin Turing sendiri yang umumnya sangat lambat. Salah satu penyederhanaan yang dilakukan adalah penggunaan prima yang kecil yaitu $P = 5$ dan $Q = 19$. Penulis sudah melakukan eksplorasi bahwa penggunaan prima yang lebih besar memakan waktu yang sangat lama bagi Mesin Turing. Alasan lain penggunaan prima tersebut adalah $N = PQ = 95$, karena input akan berbentuk string dari karakter ASCII, dan penulis berasumsi bahwa hanya karakter ASCII kode 32 - 126 yang akan digunakan, maka pemilihan angka 95 ini sudah cukup untuk menghindari *collision* saat modulo nantinya.

Selain itu, Penulis hanya mengimplementasikan perhitungan eksponen dan modulo melalui Mesin Turing. Penulis merasa bahwa proses konversi antara karakter ASCII menjadi kodenya sulit untuk dilakukan melalui mesin turing konvensional, akibatnya proses tersebut dilakukan terlebih dahulu. Untuk sebuah plaintext, setiap karakter diubah menjadi kode ASCII nya lalu dikurang dengan 31. Hal ini dilakukan agar setiap bilangan menjadi kurang dari 95. Kemudian setiap angka tersebut akan dikonversi menjadi *binary* sebelum dimasukkan kedalam pita. Penulis merubah setiap karakter satu-persatu

sehingga dalam mesin turing hanya maksimal 1 karakter saja. Hasil eksplorasi menunjukkan bahwa memasukkan seluruh karakter dalam pita dalam sekali menjalankan mesin turing akan memakan waktu terlalu banyak. Hasil dari enkripsi berupa sebuah bilangan 2 digit, dan kemudian akan digabungkan dengan hasil setiap karakter lainya menjadi sebuah string angka. Untuk Dekripsi, string angka akan dipecah menjadi blok berukuran 2. Setiap blok dimasukkan sebagai input mesin, lalu dijalankan masing-masing. Hasilnya merupakan kode ASCII yang kurang sebesar 31, sehingga akan mudah membentuk kembali plaintext setelahnya.

Terakhir, terdapat pula penyederhanaan dalam representasi angka yang digunakan. Umumnya, mesin turing menggunakan sistem *unary* pada operasi bilangan, sebagaimana visi sebenarnya Alan Turing. Namun pada permasalahan RSA ini, karena operasi yang dilakukan cukup kompleks yaitu modulo dan eksponen, sistem unary akan memakan waktu terlalu lama. Hal ini juga dibuktikan oleh eksplorasi yang dilakukan penulis dimana operasi bagi 1 huruf saja bisa memakan hampir 1 jam. Akibatnya, penulis memutuskan untuk menggunakan sistem *binary*. Dengan modulo *N=95* representasi *binary* hanya akan memakan maksimal 14 digit per angka, berbeda dengan *unary* yang bisa memerlukan ratusan digit. Penggunaan binary juga memudahkan algoritma eksponen, yaitu dengan *binary exponentiation*. Algoritma ini memiliki kompleksitas waktu *O(log E)* yang jauh lebih rendah dibanding *O(E)* pada perpangkatan dengan *unary*. Akibatnya, penggunaan *binary* akan jauh lebih cepat dibanding *unary*.

Menurut algoritma RSA, perlu nilai E dan D yang memenuhi $ED = 1 \, Mod \, 72$. Penulis memilih *D = 17* karena akan menghasilkan pula *E = 17*. Angka yang sama memudahkan mesin turing karena eksponen akan selalu tetap. Sehingga, penulis hanya mengimplementasikan 1 buah mesin turing untuk melakukan kalkulasi algoritma RSA yang akan digunakan oleh baik enkripsi maupun dekripsi. Sadari pula bahwa representasi biner dari 17 adalah *10001*. Bilangan ini hanya memiliki 2 buah digit 1, sehingga pada algoritma perkalian dalam *Binary Exponentiation* hanya diperlukan 2 kali perkalian dengan hasil, dan total hanya 5 kali pengkuadratan.

Format input untuk *tape* mesin turing adalah sebagai berikut

Y ... X ... W 00000000000000 U 00000000000000 T Input Z BBBBBBBBBBBB1 S

(Spasi pada representasi pita bukan simbol blank, melainkan hanya untuk mempermudah visualisasi pita. Simbol blank sebenarnya tidak digunakan pada mesin ini)

Antara Y & X adalah eksponen yang digunakan dalam representasi biner, dalam kasus ini *10001*. Antara X dan W berupa 95 dalam bentuk biner dengan 7 buah karakter B (bukan Blank) didepannya sebagai pengisi, tepatnya 'BBBBBBB1011111'. Antara W & U adalah area penyimpanan hasil eksponen dalam representasi biner yang tak akan melebihi 14 digit. Area U & T adalah hasil dari perkalian. Antara T dan Z adalah kode ASCII karakter input yang telah dikurang 31, dalam bentuk biner. Sehingga sejatinya maksimal hanya terisi 7 digit biner, dan 7 digit di depannya diisi oleh karakter pengisi B. Sebagai contoh untuk karakter input A ketika masuk tape akan menjadi 'BBBBBBBB100010'. Antara Z dan S adalah angka pengali dalam representasi biner. Awal mula ia bernilai 1 dan diisi oleh karakter pengisi.

Dari pemetaan algoritma RSA seperti diatas, penulis menciptakan sebuah mesin turing. Mesin turing yang digunakan untuk Enkripsi maupun Dekripsi sama yaitu dengan struktur berikut

- *Q* : Himpunan *105* buah state dengan nama state *q0* sampai *q104*.

  Fungsi setiap state dapat dibagi menjadi beberapa kategori:

  1. *Initial Set up* : *State* q0 & *state* q70
  2. Perkalian : *State* q1 - q15 & q76 - q79
  3. *Shift* N untuk persiapan modulo : *State* q17 - q27 & q85

4. Modulo : q28 - q48 & q80 - q82
5. Penanda pengurangan eksponen: q49, q56, q72
6. Penyalin hasil perkalian ke area hasil eksponen : q50 - q55, q83, q84
7. Penyalin hasil perkalian ke area kedua pengali : q56 - q72
8. Pengontrol perulangan utama : q73 - q75
9. Penyalin nilai tersimpan ke area perkalian: q88 - q104

- Σ : $\{0, 1, B, Y, X, W, U, T, Z, S\}$
- Γ : $\Sigma \cup \{O, I, b, \}$
- δ : (Definisi fungsi transisi tertera setelah ini)
- q_0 : $q0$
- B : Space, atau ‘ ‘ (Tidak digunakan dalam fungsi transisi)
- F : {}

Definisi dari fungsi transisi secara lengkap adalah sebagai berikut:

- δ(0, 1) = (70, I, R)
- δ(0, Y) = (0, Y, R)
- δ(70, U) = (1, U, R)
- δ(70, Y) = (70, Y, R)
- δ(70, X) = (70, X, R)
- δ(70, W) = (70, W, R)
- δ(70, 0) = (70, 0, R)
- δ(70, 1) = (70, 1, R)
- δ(70, B) = (70, B, R)
- δ(49, 0) = (56, B, R)
- δ(49, 1) = (50, B, R)
- δ(49, I) = (72, B, R)
- δ(49, B) = (49, B, L)
- δ(50, T) = (51, T, L)
- δ(50, X) = (50, X, R)
- δ(50, W) = (50, W, R)
- δ(50, U) = (50, U, R)
- δ(50, B) = (50, B, R)
- δ(50, 1) = (50, 1, R)
- δ(50, 0) = (50, 0, R)
- δ(50, O) = (50, O, R)
- δ(50, I) = (50, I, R)
- δ(51, 0) = (52, O, L)
- δ(51, 1) = (53, I, L)
- δ(51, O) = (51, O, L)
- δ(51, I) = (51, I, L)
- δ(51, U) = (83, U, R)
- δ(83, T) = (84, T, L)
- δ(83, I) = (83, 1, R)
- δ(83, O) = (83, 0, R)
- δ(84, W) = (56, W, R)
- δ(84, U) = (84, U, L)
- δ(84, 0) = (84, 0, L)
- δ(84, 1) = (84, 1, L)
- δ(84, I) = (84, 1, L)
- δ(84, O) = (84, 0, L)
- δ(52, U) = (54, U, L)
- δ(52, 0) = (52, 0, L)
- δ(52, 1) = (52, 1, L)
- δ(53, U) = (55, U, L)
- δ(53, 0) = (53, 0, L)
- δ(53, 1) = (53, 1, L)
- δ(54, 0) = (50, O, R)

- δ(54, 1) = (50, O, R)
- δ(54, O) = (54, O, L)
- δ(54, I) = (54, I, L)
- δ(55, 1) = (50, I, R)
- δ(55, 0) = (50, I, R)
- δ(55, O) = (55, O, L)
- δ(55, I) = (55, I, L)
- δ(56, U) = (71, U, R)
- δ(56, X) = (56, X, R)
- δ(56, W) = (56, W, R)
- δ(56, B) = (56, B, R)
- δ(56, 1) = (56, 1, R)
- δ(56, 0) = (56, 0, R)
- δ(56, I) = (56, 1, R)
- δ(56, O) = (56, 0, R)
- δ(57, T) = (60, T, R)
- δ(57, 0) = (57, 0, R)
- δ(57, 1) = (57, 1, R)
- δ(58, T) = (61, T, R)
- δ(58, 0) = (58, 0, R)
- δ(58, 1) = (58, 1, R)
- δ(59, T) = (62, T, R)
- δ(59, 0) = (59, 0, R)
- δ(59, 1) = (59, 1, R)
- δ(60, B) = (63, b, R)
- δ(60, 1) = (63, b, R)
- δ(60, 0) = (63, b, R)
- δ(60, b) = (60, b, R)
- δ(60, I) = (60, I, R)
- δ(60, O) = (60, O, R)
- δ(61, B) = (64, I, R)
- δ(61, 1) = (64, I, R)
- δ(61, 0) = (64, I, R)
- δ(61, b) = (61, b, R)
- δ(61, I) = (61, I, R)
- δ(61, O) = (61, O, R)
- δ(62, B) = (65, O, R)
- δ(62, 1) = (65, O, R)
- δ(62, 0) = (65, O, R)
- δ(62, b) = (62, b, R)
- δ(62, I) = (62, I, R)
- δ(62, O) = (62, O, R)
- δ(63, Z) = (66, Z, R)
- δ(63, 0) = (63, 0, R)
- δ(63, 1) = (63, 1, R)
- δ(63, B) = (63, B, R)
- δ(64, Z) = (67, Z, R)
- δ(64, 0) = (64, 0, R)
- δ(64, 1) = (64, 1, R)
- δ(64, B) = (64, B, R)
- δ(65, Z) = (68, Z, R)
- δ(65, 0) = (65, 0, R)
- δ(65, 1) = (65, 1, R)
- δ(65, B) = (65, B, R)
- δ(66, B) = (69, b, L)
- δ(66, 1) = (69, b, L)
- δ(66, 0) = (69, b, L)
- δ(66, b) = (66, b, R)
- δ(66, I) = (66, I, R)
- δ(66, O) = (66, O, R)
- δ(67, B) = (69, I, L)
- δ(67, 1) = (69, I, L)
- δ(67, 0) = (69, I, L)
- δ(67, b) = (67, b, R)
- δ(67, I) = (67, I, R)
- δ(67, O) = (67, O, R)
- δ(68, B) = (69, O, L)
- δ(68, 1) = (69, O, L)
- δ(68, 0) = (69, O, L)
- δ(68, b) = (68, b, R)
- δ(68, I) = (68, I, R)
- δ(68, O) = (68, O, R)
- δ(69, U) = (71, U, R)
- δ(69, 0) = (69, 0, L)
- δ(69, 1) = (69, 1, L)
- δ(69, O) = (69, O, L)
- δ(69, I) = (69, I, L)
- δ(69, T) = (69, T, L)
- δ(69, Z) = (69, Z, L)
- δ(69, B) = (69, B, L)

- δ(69, b) = (69, b, L)
- δ(71, I) = (73, I, R)
- δ(71, O) = (71, O, R)
- δ(71, T) = (74, T, L)
- δ(71, 0) = (57, O, R)
- δ(71, 1) = (58, I, R)
- δ(73, T) = (74, T, L)
- δ(73, I) = (73, I, R)
- δ(73, O) = (73, O, R)
- δ(73, 0) = (59, O, R)
- δ(73, 1) = (58, I, R)
- δ(74, U) = (79, U, L)
- δ(74, I) = (74, 0, L)
- δ(74, O) = (74, 0, L)
- δ(79, W) = (75, W, R)
- δ(79, I) = (79, 1, L)
- δ(79, O) = (79, 0, L)
- δ(79, 1) = (79, 1, L)
- δ(79, 0) = (79, 0, L)
- δ(75, S) = (2, S, L)
- δ(75, T) = (75, T, R)
- δ(75, U) = (75, U, R)
- δ(75, Z) = (75, Z, R)
- δ(75, 0) = (75, 0, R)
- δ(75, 1) = (75, 1, R)
- δ(75, B) = (75, B, R)
- δ(75, b) = (75, B, R)
- δ(75, I) = (75, 1, R)
- δ(75, O) = (75, 0, R)
- δ(1, S) = (2, S, L)
- δ(1, 0) = (1, 0, R)
- δ(1, 1) = (1, 1, R)
- δ(1, T) = (1, T, R)
- δ(1, B) = (1, B, R)
- δ(1, Z) = (1, Z, R)
- δ(2, 1) = (3, I, L)
- δ(2, 0) = (12, O, L)
- δ(2, O) = (2, O, L)
- δ(2, I) = (2, I, L)
- δ(2, B) = (78, B, R)
- δ(3, Z) = (4, Z, L)
- δ(3, 1) = (3, 1, L)
- δ(3, 0) = (3, 0, L)
- δ(3, B) = (3, B, L)
- δ(4, 0) = (76, O, L)
- δ(4, 1) = (77, I, L)
- δ(4, B) = (13, B, L)
- δ(4, T) = (13, T, L)
- δ(4, O) = (4, O, L)
- δ(4, I) = (4, I, L)
- δ(76, T) = (5, T, L)
- δ(76, Z) = (76, Z, L)
- δ(76, B) = (76, B, L)
- δ(76, O) = (76, O, L)
- δ(76, I) = (76, I, L)
- δ(76, 0) = (76, 0, L)
- δ(76, 1) = (76, 1, L)
- δ(77, T) = (6, T, L)
- δ(77, Z) = (77, Z, L)
- δ(77, B) = (77, B, L)
- δ(77, O) = (77, O, L)
- δ(77, I) = (77, I, L)
- δ(77, 0) = (77, 0, L)
- δ(77, 1) = (77, 1, L)
- δ(5, 0) = (8, O, R)
- δ(5, 1) = (8, I, R)
- δ(5, O) = (5, O, L)
- δ(5, I) = (5, I, L)
- δ(5, T) = (5, T, L)
- δ(5, B) = (5, B, L)
- δ(6, 0) = (8, I, R)
- δ(6, 1) = (7, O, L)
- δ(6, O) = (6, O, L)
- δ(6, I) = (6, I, L)
- δ(6, T) = (6, T, L)
- δ(6, B) = (6, B, L)
- δ(7, 1) = (7, 0, L)
- δ(7, 0) = (8, 1, R)
- δ(8, Z) = (4, Z, L)
- δ(8, 0) = (8, 0, R)
- δ(8, 1) = (8, 1, R)
- δ(8, I) = (8, I, R)
- δ(8, O) = (8, O, R)
- δ(8, T) = (8, T, R)
- δ(8, B) = (8, B, R)
- δ(9, S) = (2, S, L)
- δ(9, 0) = (9, 0, R)
- δ(9, 1) = (9, 1, R)
- δ(9, Z) = (9, Z, R)
- δ(9, B) = (9, B, R)
- δ(9, O) = (9, O, R)
- δ(9, I) = (9, I, R)
- δ(10, 0) = (10, 0, L)
- δ(10, 1) = (11, 0, L)
- δ(10, B) = (9, 0, R)
- δ(10, T) = (9, T, R)
- δ(11, 0) = (10, 1, L)
- δ(11, 1) = (11, 1, L)
- δ(11, B) = (9, 1, R)

- δ(11, T) = (9, T, R)
- δ(12, Z) = (10, Z, L)
- δ(12, 0) = (12, 0, L)
- δ(12, 1) = (12, 1, L)
- δ(12, B) = (12, B, L)
- δ(13, 1) = (14, 1, R)
- δ(13, 0) = (14, 0, R)
- δ(13, U) = (14, U, R)
- δ(13, B) = (13, B, L)
- δ(13, T) = (13, T, L)
- δ(13, O) = (13, 0, L)
- δ(13, I) = (13, 1, L)
- δ(14, Z) = (10, Z, L)
- δ(14, T) = (14, T, R)
- δ(14, B) = (14, B, R)
- δ(14, 1) = (14, 1, R)
- δ(14, 0) = (14, 0, R)
- δ(14, I) = (14, 1, R)
- δ(14, O) = (14, 0, R)
- δ(78, S) = (15, S, L)
- δ(78, I) = (78, 1, R)
- δ(78, O) = (78, 0, R)
- δ(78, B) = (78, B, R)
- δ(15, X) = (86, X, R)
- δ(15, Z) = (15, Z, L)
- δ(15, T) = (15, T, L)
- δ(15, U) = (15, U, L)
- δ(15, W) = (15, W, L)
- δ(15, 0) = (15, 0, L)
- δ(15, 1) = (15, 1, L)
- δ(15, O) = (15, 0, L)
- δ(15, I) = (15, 1, L)
- δ(15, B) = (15, B, L)
- δ(86, 1) = (87, 1, L)
- δ(86, 0) = (86, B, R)
- δ(86, B) = (86, B, R)
- δ(87, X) = (17, X, R)
- δ(87, B) = (87, B, L)
- δ(17, B) = (18, b, R)
- δ(17, b) = (17, b, R)
- δ(17, 1) = (85, 1, R)
- δ(18, U) = (19, U, R)
- δ(18, B) = (18, B, R)
- δ(18, W) = (18, W, R)
- δ(18, 1) = (18, 1, R)
- δ(18, 0) = (18, 0, R)
- δ(19, 0) = (20, O, L)
- δ(19, O) = (19, O, R)
- δ(19, 1) = (21, 1, L)
- δ(20, X) = (17, X, R)
- δ(20, U) = (20, U, L)
- δ(20, W) = (20, W, L)
- δ(20, 0) = (20, 0, L)
- δ(20, 1) = (20, 1, L)
- δ(20, B) = (20, B, L)
- δ(20, O) = (20, O, L)
- δ(20, b) = (20, b, L)
- δ(21, b) = (22, b, R)
- δ(21, O) = (21, 0, L)
- δ(21, I) = (21, 1, L)
- δ(21, W) = (21, W, L)
- δ(21, 1) = (21, 1, L)
- δ(21, 0) = (21, 0, L)
- δ(21, B) = (21, B, L)
- δ(21, U) = (21, U, L)
- δ(22, 0) = (23, B, L)
- δ(22, 1) = (24, B, L)
- δ(22, B) = (22, B, R)
- δ(22, W) = (27, W, L)
- δ(23, 1) = (25, 1, R)
- δ(23, 0) = (25, 0, R)
- δ(23, b) = (22, 0, R)
- δ(23, B) = (23, B, L)
- δ(24, 1) = (26, 1, R)
- δ(24, 0) = (26, 0, R)
- δ(24, b) = (22, 1, R)
- δ(24, B) = (24, B, L)
- δ(25, B) = (22, 0, R)
- δ(26, B) = (22, 1, R)
- δ(85, T) = (27, T, L)
- δ(85, U) = (85, U, R)
- δ(85, W) = (85, W, R)
- δ(85, 1) = (85, 1, R)
- δ(85, 0) = (85, 0, R)

- δ(85, I) = (85, 1, R)
- δ(85, O) = (85, 0, R)
- δ(27, X) = (28, X, R)
- δ(27, 1) = (27, 1, L)
- δ(27, 0) = (27, 0, L)
- δ(27, b) = (27, 0, L)
- δ(27, B) = (27, 0, L)
- δ(27, U) = (27, U, L)
- δ(27, W) = (27, W, L)
- δ(28, 0) = (29, O, R)
- δ(28, 1) = (30, I, R)
- δ(28, O) = (28, O, R)
- δ(28, I) = (28, I, R)
- δ(28, W) = (34, W, L)
- δ(29, U) = (31, U, R)
- δ(29, 1) = (29, 1, R)
- δ(29, 0) = (29, 0, R)
- δ(29, W) = (29, W, R)
- δ(30, U) = (32, U, R)
- δ(30, 1) = (30, 1, R)
- δ(30, 0) = (30, 0, R)
- δ(30, W) = (30, W, R)
- δ(31, 1) = (80, I, R)
- δ(31, 0) = (33, O, L)
- δ(31, O) = (31, O, R)
- δ(31, I) = (31, I, R)
- δ(32, 0) = (41, 0, L)
- δ(32, 1) = (33, I, L)
- δ(32, O) = (32, O, R)
- δ(32, I) = (32, I, R)
- δ(33, b) = (28, b, R)
- δ(33, X) = (28, X, R)
- δ(33, O) = (33, O, L)
- δ(33, I) = (33, I, L)
- δ(33, U) = (33, U, L)
- δ(33, W) = (33, W, L)
- δ(33, 1) = (33, 1, L)
- δ(33, 0) = (33, 0, L)
- δ(80, T) = (81, T, L)
- δ(80, 1) = (80, I, R)
- δ(80, 0) = (80, O, R)
- δ(81, X) = (82, X, R)
- δ(81, U) = (81, U, L)
- δ(81, W) = (81, W, L)
- δ(81, I) = (81, I, L)
- δ(81, O) = (81, O, L)
- δ(81, 1) = (81, 1, L)
- δ(81, 0) = (81, 0, L)
- δ(81, b) = (81, b, L)
- δ(82, W) = (34, W, L)
- δ(82, 1) = (82, I, R)
- δ(82, 0) = (82, O, R)
- δ(82, I) = (82, I, R)
- δ(82, O) = (82, O, R)
- δ(82, b) = (82, b, R)
- δ(34, O) = (35, 0, R)
- δ(34, I) = (36, 1, R)
- δ(34, 0) = (34, 0, L)
- δ(34, 1) = (34, 1, L)
- δ(34, X) = (42, X, R)
- δ(35, T) = (37, T, L)
- δ(35, 0) = (35, 0, R)
- δ(35, 1) = (35, 1, R)
- δ(35, W) = (35, W, R)
- δ(35, U) = (35, U, R)
- δ(35, O) = (35, O, R)
- δ(35, I) = (35, I, R)
- δ(36, T) = (38, T, L)
- δ(36, 0) = (36, 0, R)
- δ(36, 1) = (36, 1, R)
- δ(36, W) = (36, W, R)
- δ(36, U) = (36, U, R)
- δ(36, O) = (36, O, R)
- δ(36, I) = (36, I, R)
- δ(37, O) = (40, 0, L)
- δ(37, I) = (40, 1, L)

- δ(37, 0) = (37, 0, L)
- δ(37, 1) = (37, 1, L)
- δ(38, O) = (39, 1, L)
- δ(38, I) = (40, 0, L)
- δ(38, 0) = (38, 0, L)
- δ(38, 1) = (38, 1, L)
- δ(39, O) = (39, I, L)
- δ(39, I) = (40, O, L)
- δ(40, W) = (34, W, L)
- δ(40, O) = (40, O, L)
- δ(40, I) = (40, I, L)
- δ(40, U) = (40, U, L)
- δ(40, 1) = (40, 1, L)
- δ(40, 0) = (40, 0, L)
- δ(41, b) = (42, b, R)
- δ(41, X) = (42, X, R)
- δ(41, U) = (41, U, L)
- δ(41, W) = (41, W, L)
- δ(41, 1) = (41, 1, L)
- δ(41, 0) = (41, 0, L)
- δ(41, O) = (41, 0, L)
- δ(41, I) = (41, 1, L)
- δ(42, W) = (43, W, L)
- δ(42, 0) = (42, 0, R)
- δ(42, 1) = (42, 1, R)
- δ(43, 1) = (48, 1, L)
- δ(43, 0) = (16, 0, L)
- δ(16, b) = (44, b, R)
- δ(16, X) = (44, X, R)
- δ(16, 1) = (16, 1, L)
- δ(16, 0) = (16, 0, L)
- δ(44, 0) = (45, 0, R)
- δ(44, 1) = (46, 0, R)
- δ(45, W) = (47, W, L)
- δ(45, 0) = (45, 0, R)
- δ(45, 1) = (46, 0, R)
- δ(46, W) = (47, W, L)
- δ(46, 0) = (45, 1, R)
- δ(46, 1) = (46, 1, R)
- δ(47, X) = (28, X, R)
- δ(47, 1) = (47, 1, L)
- δ(47, 0) = (47, 0, L)
- δ(48, X) = (49, X, L)
- δ(48, 0) = (48, 0, L)
- δ(48, 1) = (48, 1, L)
- δ(48, b) = (48, B, L)
- δ(72, W) = (88, W, R)
- δ(72, X) = (72, X, R)
- δ(72, B) = (72, B, R)
- δ(72, 1) = (72, 1, R)
- δ(72, 0) = (72, 0, R)
- δ(88, 0) = (89, O, R)
- δ(88, 1) = (90, I, R)
- δ(88, O) = (88, O, R)
- δ(88, I) = (88, I, R)
- δ(88, U) = (100, U, R)
- δ(89, T) = (91, T, R)
- δ(89, 0) = (89, 0, R)
- δ(89, 1) = (89, 1, R)
- δ(89, O) = (89, O, R)
- δ(89, I) = (89, I, R)
- δ(89, U) = (89, U, R)
- δ(90, T) = (92, T, R)
- δ(90, 0) = (90, 0, R)
- δ(90, 1) = (90, 1, R)
- δ(90, O) = (90, O, R)
- δ(90, I) = (90, I, R)
- δ(90, U) = (90, U, R)
- δ(91, 0) = (93, O, L)
- δ(91, 1) = (93, O, L)
- δ(91, B) = (93, O, L)
- δ(91, I) = (91, I, R)
- δ(91, O) = (91, O, R)
- δ(92, 0) = (93, I, L)
- δ(92, 1) = (93, I, L)
- δ(92, B) = (93, I, L)
- δ(92, I) = (92, I, R)
- δ(92, O) = (92, O, R)
- δ(93, U) = (94, U, R)
- δ(93, T) = (93, T, L)
- δ(93, 1) = (93, 1, L)
- δ(93, 0) = (93, 0, L)
- δ(93, I) = (93, I, L)

- δ(93, O) = (93, O, L)
- δ(94, 0) = (95, O, R)
- δ(94, 1) = (96, I, R)
- δ(94, O) = (94, O, R)
- δ(94, I) = (94, I, R)
- δ(95, Z) = (97, Z, R)
- δ(95, T) = (95, T, R)
- δ(95, 1) = (95, 1, R)
- δ(95, 0) = (95, 0, R)
- δ(95, I) = (95, I, R)
- δ(95, O) = (95, O, R)
- δ(95, B) = (95, B, R)
- δ(96, Z) = (98, Z, R)
- δ(96, T) = (96, T, R)
- δ(96, 1) = (96, 1, R)
- δ(96, 0) = (96, 0, R)
- δ(96, I) = (96, I, R)
- δ(96, O) = (96, O, R)
- δ(95, B) = (96, B, R)
- δ(97, 0) = (99, O, L)
- δ(97, 1) = (99, O, L)
- δ(97, B) = (99, O, L)
- δ(97, O) = (97, O, R)
- δ(97, I) = (97, I, R)
- δ(98, 0) = (99, I, L)
- δ(98, 1) = (99, I, L)
- δ(98, B) = (99, I, L)
- δ(98, O) = (98, O, R)
- δ(98, I) = (98, I, R)
- δ(99, W) = (88, W, R)
- δ(99, Z) = (99, Z, L)
- δ(99, T) = (99, T, L)
- δ(99, U) = (99, U, L)
- δ(99, 1) = (99, 1, L)
- δ(99, 0) = (99, 0, L)
- δ(99, I) = (99, I, L)
- δ(99, O) = (99, O, L)
- δ(99, B) = (99, B, L)
- δ(100, T) = (101, T, R)
- δ(100, O) = (100, 0, R)
- δ(100, I) = (100, 0, R)
- δ(101, I) = (102, 1, R)
- δ(101, O) = (101, B, R)
- δ(102, Z) = (103, Z, R)
- δ(102, O) = (102, 0, R)
- δ(102, I) = (102, 1, R)
- δ(103, I) = (104, 1, R)
- δ(103, O) = (103, B, R)
- δ(104, S) = (2, S, L)
- δ(104, O) = (104, 0, R)
- δ(104, I) = (104, 1, R)

Selanjutnya penulis akan membahas proses enkripsi dan dekripsi RSA dengan menggunakan Mesin Turing. Perlu diingat bahwa banyaknya transisi yang terlibat akan mengakibatkan penjelasan *step by step* menjadi terlalu panjang. Oleh karena itu, penulis hanya akan membahas langkah - langkah secara garis besar saja, dan menjelaskan tiap langkah secara detail. Selain itu, mesin turing yang diimplementasikan untuk enkripsi maupun dekripsi sama (Nilai *D* dan *E* sama), sehingga penulis hanya akan membahas salah satunya saja. Mesin turing ini juga melakukan enkripsi/dekripsi untuk setiap huruf/blok

ciphertext secara satu-persatu. Sebagai contoh sederhana, penulis akan mengenkripsi sebuah karakter ‘A’, sadari bahwa contoh kecil ini sudah merepresentasikan proses enkripsi ataupun dekripsi string dengan panjang berapapun.

Sebelum masuk pita, huruf ‘A’ diubah menjadi karakter ASCII nya yaitu 65, lalu dikurangi oleh 31 menjadi 34. Dalam biner, representasinya adalah *1000010*. Secara garis besar, langkah enkripsi ‘A’ adalah seperti berikut (simbol biru menandakan lokasi *head*):

(Spasi pada representasi pita bukan simbol blank, melainkan hanya untuk mempermudah visualisasi pita. Simbol blank sebenarnya tidak digunakan pada mesin ini)

1. Awal mula, pita adalah sebagai berikut

   Y 10001 X BBBBBBB1011111 W 00000000000000 U 00000000000000 T BBBBBBBB10010 Z BBBBBBBBBBBBB1 S

2. Tambahkan digit terakhir tak bertanda di TZ menuju digit terakhir pada UT, lalu tandai

   Y 10001 X BBBBBBB1011111 W 00000000000000 U 000000000000I T BBBBBBBB10001O Z BBBBBBBBBBBBI S

3. Ulangi sampai *head* bertemu B pada TZ, penjumlahan berakhir

   Y 10001 X BBBBBBB1011111 W 00000000000000 U 00000000IOOOIIO T BBBBBBBBIOOOIO Z BBBBBBBBBBBBI S

4. Geser TZ sebanyak 1 digit ke kiri

   Y 10001 X BBBBBBB1011111 W 00000000000000 U 00000000IOOOIIO T BBBBBBBIOOOIIO Z BBBBBBBBBBBBI S

5. Ulangi langkah 2 - 4 sampai bertemu B pada ZS

   Y 10001 X BBBBBBB1011111 W 00000000000000 U 00000000IOOOIIO T BBBBBBBIOOOIIO Z BBBBBBBBBBBBI S

6. Pergi ke X sambil menghilangkan tanda

   Y 10001 X BBBBBBB1011111 W 00000000000000 U 00000000100010 T BBBBBBBB100010 Z BBBBBBBBBBBBB1 S

7. Bolak-Balik antara XW dengan UT, tandai setiap karakter pertama sampai bertemu angka 1

   Y 10001 X bbbbbbb1011111 W 00000000000000 U OOOOOOO0100010 T BBBBBBBB100010 Z BBBBBBBBBBBBB1 S

8. Bolak-Balik antara XW dengan UT, tandai setiap karakter pertama sampai bertemu angka 1

   Y 10001 X bbbbbbb1011111 W 00000000000000 U OOOOOOO0100010 T BBBBBBBB100010 Z BBBBBBBBBBBBB1 S

9. (Angka pada XW lebih besar dari UT, tidak perlu modulo) hilangkan tanda, pergi ke X

   Y 10001 X bbbbbbb1011111 W 00000000000000 U 00000000100010 T BBBBBBBB100010 Z BBBBBBBBBBBBB1 S

10. Cek karakter sebelum X lalu tandai B, jika 1, lakukan penyalinan UT ke WU, jika 0 skip proses ini

    Y 1000B X bbbbbbb1011111 W 00000000000000 U 00000000100010 T BBBBBBBB100010 Z BBBBBBBBBBBBB1 S

11. Karena bertemu 1, cek juga setelahnya. Karena sebelahnya bukan Y, lakukan penyalinan

    Y 1000B X bbbbbbb1011111 W 00000000100010 U 00000000100010 T BBBBBBBB100010 Z BBBBBBBBBBBBB1 S

12. Salin karakter pertama dari UT ke TZ dan ZS, jika 0 tandai b, ubah karakter pada WU menjadi O

    Y 1000B X bbbbbbb1011111 W 00000000100010 U O0000000100010 T bBBBBBBB100010 Z bBBBBBBBBBBBB1 S

13. Ulangi sampai bertemu 1 pertama di UT, mulai salin sesuai sebenarnya

    Y 1000B X bbbbbbb1011111 W 00000000100010 U OOOOOOOOO00010 T bbbbbbbbI00010 Z bbbbbbbbIBBBB1 S

14. Ulangi sampai seluruh karakter UT tersalin

Y 1000B X bbbbbbb1011111 W 00000000100010 U OOOOOOOOOOOOO T bbbbbbbbIOOOIO Z bbbbbbbbIOOOIO S

15. Hilangkan semua tanda, pergi ke S (Ulangi perulangan utama)

Y 1000B X BBBBBBB1011111 W 00000000100010 U 00000000000000 T BBBBBBBB100010 Z BBBBBBBB100010 S

16. Tandai digit pertama, karena 0 jangan lakukan penjumlahan

Y 1000B X BBBBBBB1011111 W 00000000100010 U 00000000000000 T BBBBBBBB100010 Z BBBBBBBB10001O S

17. Geser angka pada TZ 1 digit ke kiri

Y 1000B X BBBBBBB1011111 W 00000000100010 U 00000000000000 T BBBBBBB1000100 Z BBBBBBBB10001O S

18. Tandai angka selanjutnya di ZS, karena 1, lakukan penjumlahan ke UT

Y 1000B X BBBBBBB1011111 W 00000000100010 U 00000001000100 T BBBBBBB1000100 Z BBBBBBBB1000IO S

19. Saat penjumlahan, apabila ada *carry*, lakukan perubahan angka 1 ke 0 yang menghantar ke kiri, sampai bertemu 0.

Y 1000B X BBBBBBB1011111 W 00000000100010 U 000000OOOOIOO T BBB1000IOOOOOO Z BBBBBBBBIOOOIO S

20. Ulangi sampai bertemu B di ZS (Perkalian selesai)

Y 1000B X BBBBBBB1011111 W 00000000100010 U 00010010000100 T BBBIOOOIOOOOOO Z BBBBBBBBIOOOIO S

21. Hilangkan tanda sampai X

Y 1000B X BBBBBBB1011111 W 00000000100010 U 00010010000100 T BBB10001000000 Z BBBBBBBB100010 S

22. Bolak-Balik antara XW dengan UT, tandai setiap karakter pertama sampai bertemu angka 1

Y 1000B X bbbbBBB1011111 W 00000000100010 U OOO10010000100 T BBB10001000000 Z BBBBBBBB100010 S

23. Geser angka dalam XW sampai bertemu b

    Y 1000B X bbb1011111000 W 00000000100010 U OOO10010000100 T BBB10001000000 Z BBBBBBBB100010 S

24. Tandai angka paling kiri di XW, cek angka paling kanan di UT

    Y 1000B X bbbI011111000 W 00000000100010 U OOOI0010000100 T BBB10001000000 Z BBBBBBBB100010 S

25. Ulangi sampai ditemukan digit yang lebih besar di XW (Menandakan XW > UT, tidak bisa dilakukan pengurangan)

    Y 1000B X bbbIOI1111000 W 00000000100010 U OOOIOO10000100 T BBB10001000000 Z BBBBBBBB100010 S

26. Hilangkan tanda sampai X

    Y 1000B X BBB1011111000 W 00000000100010 U 00010010000100 T BBB10001000000 Z BBBBBBBB100010 S

27. Cek angka sebelum W, apabila 0 lanjutkan, jika 1, selesai modulo

    Y 1000B X BBB10111110000 W 00000000100010 U 00010010000100 T BBB10001000000 Z BBBBBBBB100010 S

28. Geser angka XW 1 digit ke kanan

    Y 1000B X BBB0101111100 W 00000000100010 U 00010010000100 T BBB10001000000 Z BBBBBBBB100010 S

29. Lakukan pengurangan

    Y 1000B X BBBOIOIIIIIOO W 00000000100010 U 000OIOOOOOIOOO T BBB10001000000 Z BBBBBBBB100010 S

30. Ulangi sampai angka terakhir XW adalah 1 (Modulo selesai)

    Y 1000B X BBBBBOIOIIIII W 00000000100010 U OOOOOOOOOIOOOO T BBB10001000000 Z BBBBBBBB100010 S

31. Hilangkan tanda selagi pindah ke X

    Y 1000B X BBBBBB1011111 W 00000000100010 U 00000000010000 T BBB10001000000 Z BBBBBBBB100010 S

32. Ubah digit pertama di kiri X menjadi B, karena 0 tidak perlu dilakukan penyimpanan hasil perkalian

    Y 100BB X BBBBBB1011111 W 00000000100010 U 00000000010000 T BBB10001000000 Z BBBBBBBB100010 S

33. Langsung skip ke tahap penyalinan hasil perkalian ke area perkalian selanjutnya

    Y 100BB X BBBBBB1011111 W 00000000100010 U OOOOOOOOOOOOO T bbbbbbbbbbIOOOO Z bbbbbbbbbbIOOOO S

34. Kanan ke S selagi menghilangkan tanda

    Y 100BB X BBBBBB1011111 W 00000000100010 U 00000000000000 T BBBBBBBBB10000 Z BBBBBBBBB10000 S

    (2 iterasi perungalan utama selanjutnya tidak dijabarkan karena tahapnya sereupa namun akan ditunjukkan hasilnya)

35. Hasil iterasi ketiga

    Y 10BBB X BBBBBB1011111 W 00000000100010 U 00000000000000 T BBBBBBB1000010 Z BBBBBBB1000010 S

36. Hasil iterasi keempat

    Y 1BBBB X BBBBBB1011111 W 00000000100010 U 00000000000000 T BBBBBBB1010001 Z BBBBBBB1010001 S

37. Lakukan tahap perkalian seperti sebelumnya

    Y 1BBBB X BBBBBB1011111 W 00000000100010 U 01100110100001 T BIOIOOIOOOOOO Z BBBBBBBIOIOOOI S

38. Lakukan tahap modulo

    Y 1BBBB X BBBBBB1011111 W 00000000100010 U 00000000000110 T BIOIOOIOOOOOO Z BBBBBBBIOIOOOI S

39. Hilangkan tanda sampai X

Y 1BBBB X BBBBBB1011111 W 00000000100010 U 00000000000110 T B1010010000000 Z BBBBBBB1010001 S

40. Kiri sampai ditemukan angka pertama setelah X, karena 1, cek juga sebelahnya. Karena sebelahnya Y, masuk ke tahap penyalinan angka yang tersimpan menuju area perkalian
    Y BBBBB X BBBBBB1011111 W 00000000100010 U 00000000000110 T B1010010000000 Z BBBBBBB1010001 S

41. Saling angka pertama setelah W ke setelah T
    Y BBBBB X BBBBBB1011111 W O0000000100010 U 00000000000110 T O1010010000000 Z BBBBBBB1010001 S

42. Saling angka pertama setelah U ke setelah Z
    Y BBBBB X BBBBBB1011111 W O0000000100010 U O0000000000110 T O1010010000000 Z OBBBBBB1010001 S

43. Ulangi sampai karakter pertama yang ditemukan setelah W adalah U
    Y BBBBB X BBBBBB1011111 W OOOOOOOOIOOOIO U OOOOOOOOOOOIIO T OOOOOOOOIOOOIO Z OOOOOOOOOOOIIO S

44. Hilangkan tanda dan ubah O menjadi B, lalu ke S
    Y BBBBB X BBBBBB1011111 W O0000000100010 U O0000000000110 T BBBBBBBB100010 Z BBBBBBBBBBB110 S

45. Lakukan perkalian seperti sebelumnya
    Y BBBBB X BBBBBB1011111 W O0000000100010 U 000011001100 T BBBBBIOOOIOOO Z BBBBBBBBBBBIIOS

46. Lakukan Modulo
    Y BBBBB X BBBBBB1011111 W O0000000100010 U 00000000001110 T BBBBBIOOOIOOO Z BBBBBBBBBBBIIOS

47. *Head* berpindah ke X
    Y BBBBB X BBBBBB1011111 W O0000000100010 U 00000000001110T BBBBBIOOOIOOO Z BBBBBBBBBBBIIOS

48. Pencarian atas 0/1 pertama di kiri X, namun malah ditemukan Y dimana transisi δ(49, Y) tidak terdefinisi

    Y BBBBB X BBBBBB1011111 W O0000000100010 U 00000000001110T BBBBBIOOOIOOO Z BBBBBBBBBBBIIOS

49. Mesin melakukan halting. Nilai antara U & T dibaca nilai desimalnya yaitu 14

    Y BBBBB X BBBBBB1011111 W O0000000100010 U 00000000001110 T BBBBBIOOOIOOO Z BBBBBBBBBBBIIOS

50. 14 merupakan hasil enkripsi dari 'A', terbukti oleh benarnya

$$34^{17} \, mod \, 95 \; = \; 14$$

    Tahapan yang sama berlaku untuk Dekripsi, juga dapat dilakukan untuk plain/cipher text dengan ukuran lebih dari 1.