

# **KONSEP KEAMANAN PADA RADIO FREQUENCY IDENTIFICATION**

oleh:

Mohamad Supandri  
NIM : 23203119

Dosen  
Dr.Ir. Budi Rahardjo



**INSTITUT TEKNOLOGI BANDUNG  
2004**

## **ABSTRAK**

# **KONSEP KEAMANAN PADA RADIO FREQUENCY IDENTIFICATION**

**Oleh  
Mohamad Supandri**

**Departemen Teknik Elektro  
Institut Teknologi Bandung**

*Radio Frekuensi Identification* (RFID) sudah merambah pada kehidupan kita dan sangat bermanfaat pada manufacture, pengendali inventaris barang. Dengan teknologi pengembangan manufaktur silikon harga RFID dapat ditekan secara signifikan. Pada masa depan RFID “*Smart-label*” akan dapat menggantikan kedudukan *optical barcode* pada pelabelan barang. Kerugian penyebaran perlengkapan RFID pada label barang dapat membuka masalah ancaman, keamanan baru dan masalah privasi, meskipun di lingkungan pabrik yang tertutup sekalipun. Pada tulisan ini mengenalkan RFID dan beberapa serangan yang potensial yang mungkin terjadi pada sistem keamanan dan privasi pada penggunaan RFID, dan menawarkan beberapa konsep keamanan untuk RFID yang meliputi mekanisme *low cost access control*, pencegahan penelusuran *tag* oleh *reader* yang tidak sah, kunci *hash lock*, algoritma dua varian *tree-walking anti collision* untuk menanggulangi penyadap jarak jauh (*long range eavesdropper*), konsep sederhana memperkuat keamanan terutama dalam mendeteksi *Denial of service*.

## DAFTAR ISI

	Halaman
ABSTRAK	ii
DAFTAR ISI	iii
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Jenis RFID dan Bagian RFID	2
1.3 Cara Kerja RFID	2
1.4 Tujuan Penulisan	4
BAB II Masalah Serangan RFID	4
2.1 Karakteristik Aktor Penyerang	4
2.2 Ancaman dan Serangan (Threat and Attack)	5
BAB III KONSEP EAMANAN RFID	9
3.1 <i>Hash Lock</i>	9
3.2 <i>Randomized Hash Lock</i>	12
3.3 <i>Low-Cost Hash Fuctions</i>	13
3.3.1 <i>Definisi Hash</i>	13
3.3.2 <i>Desain Pendekatan Fungsi Hash</i>	14
3.3.3 <i>Cellular Automata</i>	16
3.3.4 <i>Non-Linear Feedback Shift Registers</i>	17
3.4 <i>Secure Anti Collision</i>	20
3.4.1 <i>Blinded Tree-Walking</i>	20
3.4.2 <i>Radomized Tree-Walking</i>	22
3.5 Usulan Keamanan RFID Lainnya	24
3.5.1 <i>Asymmetric Key Agrrement</i>	24
3.5.2 <i>Chafing dan Winnowing</i>	24
3.5.3 <i>Pendeteksi</i>	24
3.5.4 <i>Jeritan Tag (Screaming Tag)</i>	24
3.5.5 <i>Agent Security</i>	25
3.5.6 <i>Mencetak Master Key</i>	25
<i>BAB IV PENUTUP</i>	26
<i>DAFTAR PUSTAKA</i>	27

# BAB I

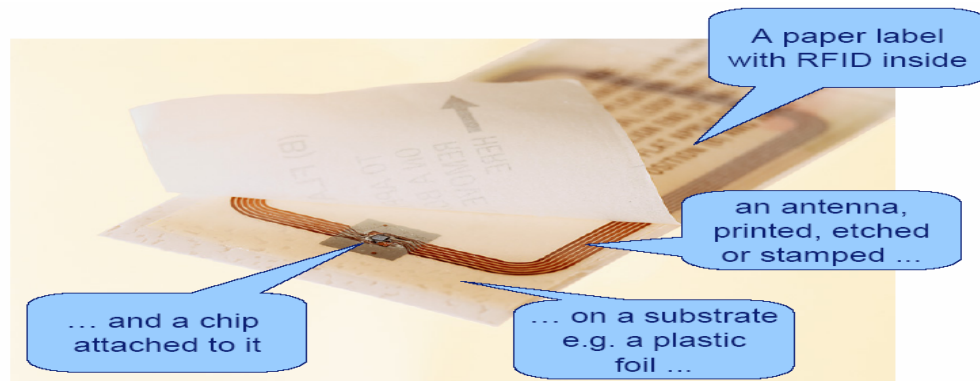
## PENDAHULUAN

### 1.1 Latar Belakang

*Radio Frekuensi Identification* (RFID) sudah banyak digunakan pada pabrik sangat bermanfaat untuk mendukung rantai manajemen dan pengendalian persediaan. RFID dapat mengidentifikasi objek secara otomatis, RFID dapat diprediksi akan mengganti *barcode* yang telah terlebih dahulu dikenal, Menurut Stephen A. Weis “ *One familiar optical barcode is the Universal Product Code (UPC) designed in 1973* [1] dan banyak di gunakan pada banyak produk untuk konsumen. Kemajuan produksi dari silikon membuat RFID berharga murah . Sistem RFID terdiri dari *Tag* frekuensi Radio atau *Transponder* dan *Tag reader* atau *receiver*.

*Tag reader* meminta isi yang dipancarkan oleh signal RF. *Tag* merespon dengan memancarkan kembali data *resident* secara lengkap meliputi serial nomor urut yang unik. RFID mempunyai beberapa keuntungan yang utama melebihi sistem *barcode* yaitu kemungkinan data dapat di baca secara otomatis tanpa memperhatikan garis arah pembacaan , melewati bahan *non konduktor* seperti karton kertas dengan kecepatan akses beberapa ratus *tag* per detik pada jarak beberapa ( $\pm 100$ ) meter . *Tag* RFID terbuat dari *microchip* dengan dasar bahan dari silikon yang mempunyai kemampuan fungsi identifikasi sederhana yang disatukan dalam satu desain. Kemampuan *tag* RFID untuk membaca dan menulis (*read/write*) menyimpan pada *storage* untuk mendukung enkripsi dan kontrol akses. Pada Gambar 1.1 diperlihatkan bagian bagian *tag* RFID.

RFID yang didesain dipadukan pada sistem identifikasi pada semua tingkat rantai persediaan semua lini dilibatkan akan dapat mempunyai manfaat tidak hanya untuk pabrikan tetapi juga untuk konsumen, pengawas obat dan makanan bahkan untuk pengolah limbah buangan.



Gambar 1.1 Bagian bagian *Tag* RFID <sup>[4]</sup>

## 1.2 Jenis RFID dan Bagian RFID

RFID awalnya terdiri dari dua jenis yaitu yang menggunakan baterai (aktif) dan tidak menggunakan baterai (pasif), yang tidak menggunakan baterai hanya dapat dibaca, sedangkan yang tidak menggunakan baterai dapat dibaca dan dituli [2]. kedua jenis ini dinamakan *Induktive Coupled RFID Tags*. Setiap bagian *Tag* terdiri dari.

### a) *Silicon Microprocessor*

Ini adalah sebuah *chip* yang terletak dalam sebuah *tag* yang berfungsi sebagai penyimpan data.

### b) *Metal Coil*

Sebuah komponen yang terbuat dari kawat aluminium yang berfungsi sebagai antena yang dapat beroperasi pada frekuensi 13,56 MHz. Jika sebuah *tag* masuk ke dalam jangkauan *reader* maka antena ini akan mengirimkan data yang ada pada *tag* kepada *reader* terdekat.

### c) *Encapsulating Material*

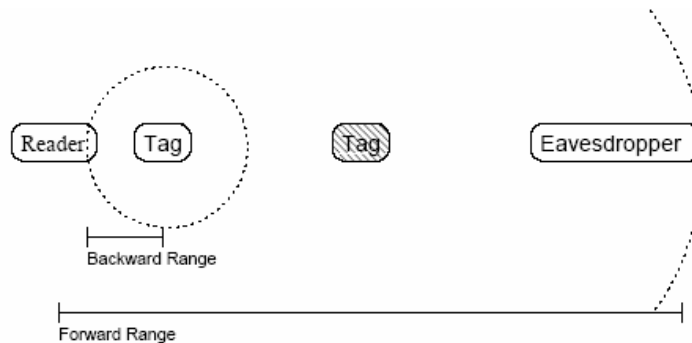
*Encapsulating Material* adalah bahan yang membungkus *tag* yang terbuat dari bahan kaca.

## 1.3 Cara Kerja RFID

Label *tag* RFID yang tidak memiliki baterai antenalah yang berfungsi sebagai pencatu sumber daya dengan memanfaatkan medan magnet dari pembaca (*reader*)

dan memodulasi medan magnet, yang kemudian digunakan kembali untuk mengirimkan data yang ada dalam *tag* label RFID.

Data yang diterima *reader* diteruskan ke *Database host* komputer. Pada Gambar 1.2 sekema proses kerja RFID



Gambar 1.2 Bagan kerja *reader* RFID <sup>[3]</sup>

Kerugian penyebaran penggunaan RFID yang universal akan memudahkan terbukannya *privasi*, *sepionase*, dan menimbulkan ancaman keamanan baru pada suatu lingkungan pabrik yang tertutup sekalipun. Penjualan eceran yang diberi label RFID dengan *tag* yang tidak dilindungi akan dapat dimonitor dan di-*tracked* oleh pesaing lain.

Pabrik mengeluarkan biaya pembuatan RFID lebih tinggi supaya dapat mendukung kriptografi seperti disampaikan Stephen A. Weis “*Most manufacturing processes currently deploying RFID systems are for higher value items, allowing tag costs to be in the US\$0.50-US\$1.00 range. Tags priced in this range could support basic cryptographic primitives or tamper-resistant packaging,*” [1]. *Tag* yang menghabiskan biaya besar ini diharapkan dapat mendukung sistem keamanan dengan kriptografi.

#### 1.4 Tujuan Penulisan

Berdasarkan uraian di atas paper tugas akhir Mata Kuliah Keamanan Sistem Lanjut EC 7010 bertujuan untuk mempelajari konsep keamanan RFID dengan cara mengidentifikasi macam serangan yang mungkin pada RFID dan mengusulkan konsep pengamanan RFID

## **BAB II**

### **MASALAH SERANGAN RFID**

Kemudahan dan keuntungan besar yang ditawarkan oleh sistem RFID akan menimbulkan pengorbanan privasi dan keamanan. Sifat rentan atau mudah diserang (*vulnerabilities*) untuk serangan fisik, pemalsuan, mengendus (*spoofing*), disadap (*eavesdropping*), memadati trafik atau serangan *deniel of service* dapat mengancam *tag* yang tidak terlindungi. Masing masing resiko serangan ini dapat mempengaruhi privasi dan keamanan baik secara individu maupun organisasi.

Secara tradisi banyak dokumen masalah kriptografi yang menjelaskan tentang karakter manusia yang akan mendatangkan jenis jenis serangan terhadap sistem RFID. Jenis jenis serangan dapat digambarkan seperti pembahasan bagian 2.1.

#### **2.1 Karakter Aktor Penyerang**

Pada bagian 2.1 digambarkan jenis jenis serangan terhadap *tag* RFID yang disebabkan berbagai karakter manusia, yang diperankan dalam aktor sebagai berikut :

**Rahwono** : Rahwono merupakan penyerang yang paling kuat, dia dapat melakukan penyerangan secara fisik terhadap *tag* RFID dan dapat melakukan penyerangan canggih dalam laboratorium yang meliputi : penyelidikan, memindahkan matrial (mencuri) dengan jalan menukar *tag*, mengetcing atau menggores matrial, serangan energi (radiasi atau interferensi), gangguan sirkuit.

Rahwono dalam penyerangannya tidak dapat melakukan di muka umum atau pada skala yang besar. Hal ini dapat diperdebatkan untuk diperhatikan mengenai privasi dan keamanan, ketika Rahwono mendapatkan *tag* dengan diam diam dan ditempelkan pada suatu pembungkus tanpa terdeteksi.

**Bagong** : Bagong tidak dapat menyerang *tag* secara fisik tetapi dapat secara aktif ikut serta dalam protokol atau menyamar seperti pemiliknya sendiri.

Bagong dapat melakukan *query* ke *tag* atau memberikan reaksi terhadap *query reader* sesuka hatinya.

**Cakil** : Cakil dapat memerankan peran yang pasif, dia tidak dapat secara aktif mengambil bagian pada protokol dan dibatasi pada menyadap (*eavesdropping*). Cakil hanya dapat mendengarkan logik 1 dan 0 yang dipancarkan dari protokol. Sebagai lawan radiasi elektro magnetik oleh serangan Rahwono .

**Durno** : Durno lebih lemah dibandingkan Cakil, Durno tidak dapat membaca isi dari pesan , tetapi masih dapat mendeteksi kehadiran mereka, dengan kata lain Durno dibatasi pada analisa trafik. Durno dapat mendeteksi banyaknya pesan yang dikirim dan waktu pesan dikirim. Durno dapat melakukan penyerangan “*location privacy*” dalam beberapa situasi. Durno merupakan suatu ancaman seperti juga Cakil..

**Denial** : Denial adalah yang paling lemah dari semua karakter, dia tidak dapat membaca maupun mendeteksi kehadiran pesan. Denial hanya dapat mengacaukan siaran (*broadcast*) dengan menghalangi pesan masuk atau dengan kata lain serangan *Denial of service* RFID mendapatkan aliran pesan yang lebih, serangan Denial ini dapat mendatangkan kerugian yang besar.

## **2.2 Ancaman dan Serangan (*Threats and Attacks*)**

Serangan yang diwujudkan oleh karakter manusia dapat mengancam keamanan dan privasi individu. Sebagai contoh dengan mempertimbangkan *tag* pada perdagangan eceran mempunyai label RFID yang tidak aman dan dibawa oleh konsumen. Jika aksesnya tidak terkontrol karakter Bagong dapat dengan sesukanya melakukan *query* terhadap *tag* untuk kepentingannya sendiri. Kekuatan serangan ini nampaknya tidak berbahaya ketika dilakukan pemeriksaan yang pertama. Atau mungkin suatu saat orang dapat mengintip isi barang belanjaan dalam kantong belanja seseorang. Disamping tetangga yang ingin tahu, sahabat, tenaga marketing yang cermat dapat memilihat produk yang mereka bawa. Misalkan bahwa isi barang dapat dijamin aman, barang kali dengan menggantikan informasi produk dengan *pointer database*. Karakter seperti di atas dapat membuka secara fisik ditreking dari *tag* yang mereka bawa. Jika *tag* merespon, maka dapat diramalkan dengann *pointer* yang sama *tag* akan merespon tiap *query* oleh *reader* yang telah diinstallkan oleh karakter Bagong yang melewati suatu area. Hal ini melanggar “*area privacy*” individu yang

seharusnya tidak dapat ditracking secara otomatis. Hal serupa ini akan muncul pada sistem komputerisasi seperti *bluetooth*.

Apakah penempatan privasi merupakan sesuatu yang beresiko serius. Pada hal secara reguler kita difilmkan oleh kamera vidio yang digabungkan dengan perangkat lunak pengenalan *message* muka. Seseorang dapat mengikuti kita sampai wilayah umum atau mengadakan suatu penyelidikan untuk melakukan investigasi

Perbedaan dengan teknologi RFID adalah treking dapat dilakukan secara otomatis dan didapatkan keakuratan yang tinggi.

Walaupun kebanyakan orang tidak memperdulikan jika ia ditreking dimuka umum. Menggolongkan seperti pasien AIDS, beribadah, bahkan penjual peralata *sex* perlu untuk dilindungi secara otomatis, maka perlu dipilih identifikasi data dipindahkan dari *tag* pada waktu pembelian.

Sebagai contoh, nomor urut dihapus, tetapi nomor produk dari kode pabrikan ditinggalkan tetap utuh, kostumer dapat mengambil keuntungan dari informasi tanpa menjadi treker oleh orang yang mempunyai ID yang unik. Kelemahan karakter Bagong masih dapat mempermalukan orang yang membawa produk.

Konbinasi gabungan dari merek yang saling berdekatan dengan traker (*reader*) secara bersamaan, salah satu merek dagang tidak dapat diidentifikasi. Tentu saja *tag* tidak dapat secara lengkap terdeteksi digerbang keluar.

Pada pasar yang besar RFID berada pada pembuangan limbah dan industri daur ulang. Ancaman *tag* yang tidak aman, tidak dibatasi pelanggaran privasi individu, Suatu pedagang eceran (*retail*) yang diinstal sistem rak yang cerdas. Pada keadaan demikian suatu penyerangan yang dilakukan oleh karakter Rahwono tidak digunakan . Diasumsikan toko atau gudang mempunyai peralatan kamera vidio atau ada petugas keamanan yang mendeteksi melawan serangan teradap *tag hardware*. Tetapi bagaimanapun karakter Bagong dapat menyerang suatu sistem yang tidak aman dengan beberapa cara. Jika kekurangan *tag* yang dibaca akses kontrol, Bagong secara otomatis dapat melakukan *query* keseluruhan inventori toko. Dengan

pelaksanaan *scan* yang berkala, Bagong dapat memperoleh data penjualan yang merupakan informasi yang menguntungkan. Bagong dapat menjual dan menawarkan jasa sebagai mata mata.

Alternatif lain Bagong dapat menulis kembali *tag* dari isi yang mahal diganti dengan data produk yang murah. Serangan ini seperti dilaksanakan sama dengan serangan pada *barcode*. Ketika operator mengerti mungkin akan memberi tahu bahwa ada stiker (*label*) yang palsu pada suatu kotak atau paket . Maka Bagong dapat menulis kembali *tag* yang tidak aman dengan demikian Bagong dapat membuat data palsu pada *tag*. Item yang mahal diganti dengan harga yang murah.

Disamping memalsukan isi *tag* dengan kemampuannya ia dapat memudahkan pencurian. Asumsi toko telah mempunyai sistem *ceckout* otomatis. konsumen dapat memasukan item belanjanya sendiri dan dibayarkannya ketika keluar.

Rak yang cerdas akan melakukan treking ketika materi (barang) telah dipindahkan dari rak. Jika suatu *inconsistency* muncul, seperti suatu item dipindahkan (diambil) dan terdata tidak meninggalkan toko maka petugas keamanan dapat disiagakan. Karena Bagong mempunyai kemampuan untuk mengubah *tag* dia dapat mengalahkan gerbang otomatis. Bagong dapat memindahkan suatu item dari rak dan tempat penyimpanan (*deposito*) ke dalam suatu kantong penyamaran. Secara normal, suatu produk lenyap akan didaftarkan dalam keadaan suatu keganjilan . Bagaimanapun Bagong dapat mengganti kan produk yang lenyap dengan umpan (pengganti palsu) meniru RFID *tag* yang asli. Rak akan berfikir bahwa item telah digantikan dan membiarkan Bagong keluar pintu toko. Bagong dapat membangun alat tunggal yang akan meniru banyak *tag* dengan segera. Upan ini juga menyerang pos RFID Bagong mempunyai suatu keuntungan sebab alat umpannya bukan tunduk kepada yang fisiknya sama karena pembatasan biaya sistem RFID. Sedangkan suatu *tag* harus murah dan mudah disatukan dalam pengemasan. Bagong dapat membuat suatu alat yang aktif, dan ada suatu perangsang untuk mencuri bila alat umpan lebih murah dari barang dagangan yang mereka curi.

Meskipun Cakil lebih lemah dibandingkan dengan Bagong dia masih menimbulkan suatu ancaman keamanan dan privasi pribadi. Cakil tidak dapat sesukanya *query tag* milik konsumen tetapi dia dapat menyadap seperti *reader* yang sah. Sebagai contoh Cakil dapat menunggu di luar suatu apotik, pagar putar atau dimanapun *tag* dapat di*query*. Cakil dapat melakukan sepijone industri dengan perekaman suatu *query* inventori kepunyaan toko. Cakil dapat juga menyadap serangan aktif Bagong.

Durno juga berposisi sebagai ancaman keamanan dan privasi pribadi. Walaupun dia tidak dapat melakukan *query* atau menyadap Durno dapat mendeteksi kehadiran *tag* dan *query*. Individu dapat ditreking jika mereka secara konsisten membawa nomor alat yang spesifik, terutama jika mereka membawa nomor yang tinggi yang tidak biasa. Durno dapat mengumpulkan sedikit data inventori.

Penyerang yang paling lemah adalah Denial, Denial juga merupakan suatu ancaman, Denial tidak mendapatkan suatu informasi yang bermanfaat dari suatu sistem RFID, Tetapi dapat menimbulkan serangan *Denial of service* pada sistem RFID. Dia dapat membanjiri *Radio Frekuensi* (RF) dengan *noise* untuk mengacaukan komunikasi.

Analogi dengan *barcode* orang dapat merusak *barcode* dengan menulis atau mencabut. Denial dapat mengganggu sistem RFID secara otomatis tersebar luas melalui RF.

## **BAB III**

### **KONSEP KEAMANAN RFID**

Pada BAB III diasumsikan bahwa *tag* peka terhadap serangan fisik. Pembahasan selanjutnya difokuskan pada serangan aktif dan serangan penyadapan (*eavesdropping*). Serangan ini dapat melanggar privasi pribadi seperti halnya kebocoran data inventori yang sensitip. Serangan *traffic analysis* juga merupakan suatu ancaman. Fakta *Denial of service* dapat juga merupakan serangan yang berpotensi mengganggu dan mahal penangannya.

*Active querying attacks may be addressed by limiting who is permitted to read tag data through access control* [2]. Serangan berupa *query* aktif mungkin ditujukan karena pembatasan otoritas orang yang diijinkan untuk membaca data *tag* melewati kontrol akses. Penyadap (*eavesdroppers*) mungkin dihadapkan pada keyakinan untuk tidak membocorkan/menyiarkan data *tag* secara bebas.

Bagian 3.1 akan dibahas masalah mekanisme *low cost access control* didasari pada fungsi *hash*, mekanisme ini dikenal dengan *hash lock*.

Bagian 3.2 membahas *Radomized Hash Lock* yaitu mencegah treking *tag* oleh *reader* yang tidak sah.

Bagian 3.3 menggambarkan persyaratan *hash properties* dan mengeksplor berbagai pendekatan untuk membangun fungsi *hash* murah yang cocok untuk kunci *hash* (*hash lock*)

Bagian 3.4 usulan dua varian dari algoritma *tree-walking anticollision* yang menawarkan keamanan yang lebih besar melawan penyadap jarak jauh (*long range eavesdropper*).

Bagian 3.5 berbagai konsep sederhana yang memperkuat keamanan terutama dalam mendeteksi *Denial of Service attack*.

#### **3.1 Hash Lock**

Mekanisme akses kontrol adalah dasar frekuensi pada *public-key cryptography primitive* atau persyaratan keadaan distribusi kunci simetri primitif. Arus *tag* mengurangi perhitungan sumber daya untuk mendukung mekanisme akses kontrol

tradisional. *Hash lock* adalah mekanisme akses kontrol yang sederhana berdasarkan *one way hash function* (fungsi *hash* yang searah).

*Tag* dalam skema *hash lock* (kunci *hash*) masing masing akan dilengkapi dengan fungsi *hash*. Dalam prakteknya optimalisasi *hardware* akan mencukupi untuk kriptografi *hash*. Masing masing *hash* memungkinkan *tag* dalam desain akan menyediakan sebagian cadangan memori untuk temporer *metaID*. *Tag* akan beroperasi juga pada *state locked* dan *unlocked* (keadaan terkunci dan tidak terkunci). *State* ini mungkin dapat digambarkan untuk desain *tag* yang berbeda. *Tag* yang tidak terkunci diasumsikan mempunyai kemampuan yang lengkap fungsinya untuk *reader* yang dekat.

*Tag* yang memiliki kunci *tag*, pertama kali memiliki kunci secara acak, kemudian menghitung nilai *hash* dan kunci. output *hash* sebagai *metaID*, ketika  $metaID \leftarrow hash(key)$ . Pemilik *tag* akan menyimpan *metaID* pada *tag* dan *toggle* kedalam keadaan terkunci. Kotak kedalam keadaan status terkunci. Penulisan *metaID* dapat terjadi di atas *interface* RF atau di atas *physical contact channel* untuk penambahan keamanan. Ketika sedang menerima nilai *metaID*, *Tag* memasuki keadaan terkunci. Ketika *tag* merespon semua *query* dengan hanya menggunakan *metaID* dan tidak merespon fungsi yang lain Pada akhirnya pemilik *tag* akan menyimpan kunci dan *metaID* dalam *back-end database*. Ringkasan protokol ini pada gambar 3.1

1. Reader R selects a random *key* and computes  $metaID := hash(key)$ .
2. R writes *metaID* to *Tag* T.
3. T enter the locked state.
4. R stores the pair (*metaID.key*) locally.

Gambar 3.1 Protokol untuk mengunci *hash lock*<sup>[2]</sup>

Untuk membuka *tag*, pemilik pertama kali *query metaID* dari *tag* dan menggunakan nilai untuk mencari kunci ke dalam *back-end database*. Pemilik mengirim nilai kunci ke *tag*, nilai *hash* yang diterima dibandingkan pada penyimpan *metaID*, jika nilainya cocok (*match*),  $hash(key) = metaID$ , maka *tag* membuka dan

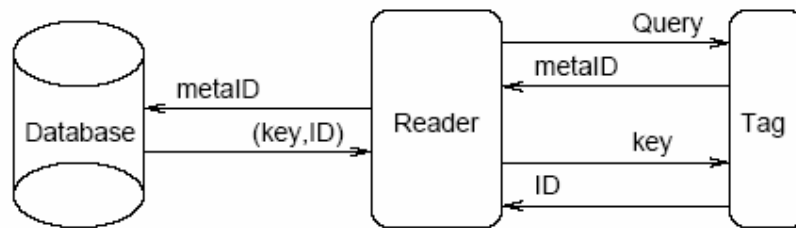
menawarkan semua kemampuan kepada pembaca yang terdekat . Ringkasan protokol ini digambarkan pada Gambar 3.2 dan Gambar 3.3.

Untuk mencegah pembajakan *tag* yang tidak terkunci, *tag* harus terbuka dengan singkat untuk melaksanakan fungsi sebelum dikunci kembali.

1. Reader R queries *Tag* T for T its *metaID*.
2. R looks up (*metaID*,*key*) locally.
3. R sends *key* to T.
4. If ( $hash(key) == metaID$ ), T unlock itself.

Gambar 3.2 Protokol untuk membuka *hash lock*<sup>[2]</sup>

Didasari dengan kesukaran membalikan *one-way hash function*, skema ini mencegah *reader* yang tidak sah isi *tag*, meskipun tidak mencegah musuh dapat melakukan query *tag* untuk *metaID* kemudian menyadap *tag* pada *reader* yang sah dalam pengulangan serangan *reader* yang sah akan mengungkapkan kunci pada penyadap. Bagaimanapun *reader* harus memeriksa isi *tag* melawan *back-end database* untuk memverikasi bahwa hubungan dengan *metaID* yang sesuai. Pendektesian *inconsistency* sedikit siaga pada pembaca yang *spoofing* yang mungkin telah menyerang.



Gambar 3.3 *Reader* membuka kunci *hash-locked tag*<sup>[1]</sup>

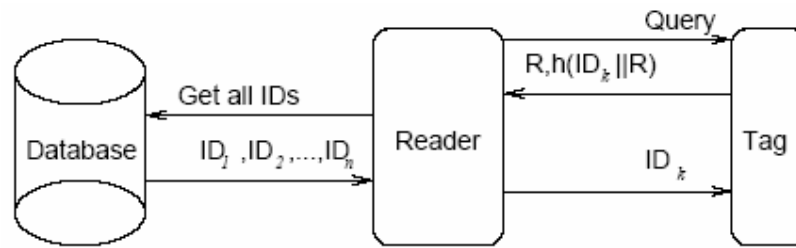
### 3.2 Randomized Hash Lock

Untuk menghindari tracking *tag* memotivasi menambah mode operasi, ketika pada mode ini diasumsikan *tag* tidak akan merespon pada *query* oleh para pemakai yang tidak sah, tetapi masih dapat diidentifikasi oleh *reader* yang sah. Presentasi *heuristik* praktis pada *one-way hash*, sesuai dengan dengan konsumen dengan jumlah *tag* yang kecil, diusulkan juga secara teori lebih varian yang lebih kuat berdasarkan pada *pseudo-random functions* (PRFs).

Pada bagian 3.1 *tag* dilengkapi dengan *one-way hash function*, tetapi sekarang juga mempunyai *random number generator*. Diasumsikan bahwa *tag reader* yang sah “mengetahui apa yang ia punyai” sebelum *scanning tag*. *Tag* yang tidak terkunci dapat dikunci dengan sederhana oleh *reader* tidak memerlukan suatu protokol. Untuk membuka *reader* pertama kali mengirim *query* sederhana *tag* merespon pada *query* dengan pembangkit (*generator*) untuk kesempatan  $R$  yang dipilih *uniform* secara acak. *Tag* kemudian meng”*hash*” untuk kesempatan  $R$  diurut (*concatened*) dengan *tag ID*. Akhirnya *tag* mengirim kembali pada *reader* yang sesuai keduanya untuk kesempatan  $R$  dan *out hash* bahwa pasangan adalah  $(R, h(ID||R))$ . Ketika *reader* yang sah menerima  $(R, h(ID||R))$ , pelaksanaan dari semua  $ID$  oleh masing masing *hash* urutannya dengan  $R$  sampai ditemukan yang cocok (*match*). ini dapat membuka *tag* dengan mengirim nilai  $ID$ . sebagai alternatif, karena *reader* telah mengetahui nilai  $ID$ , dapat meninggalkan *tag* yang terkunci. Ringkasan protokol ini seperti pada Gambar 3.4 dan digambarkan pada Gambar 3.5

1. Reader  $R$  queries *Tag*  $T$ .
2.  $T$  generates a random nonce  $R$  and computes *hash*  $(ID||R)$ .
3.  $T$  send  $(R, hash(ID||R))$  to  $R$ .
4.  $R$  computes *hash* $(ID_i||R)$  for its known  $ID_i$  values.
5. If  $R$  finds a match such that  $hash(ID_j||R) = hash(ID||R)$ ,  $R$  sends  $ID_j$  to  $T$ .
6.  $T$  unlock itself if it receives  $ID_j = ID$ .

Gambar3.4 Protokol *Randomized Hash Lock* <sup>[2]</sup>



Gambar 3.5 Reader membuka tag yang memiliki ID is  $k$  dalam skema ramized hash lock<sup>[2]</sup>

### 3.3 Low-Cost Hash Functions

Usulan pada bagian 3.1 dan 3.2 mengandalkan pada *low-cost hash function* seperti dasar pembangunan blok. Tipe implementasi komersial standar *hash function* seperti Secure Hash Algoritama (SHA-1) mengambil pada penawaran dari 20.000 – 30.000 *gate*. Hal ini melebihi dari sumber daya yang ada dari seluruh desain *low-cost* RFID.

Seperti yang telah dibahas. *Low-cost tags* diasumsikan akan menjadi persyaratan untuk menyelenggarakan performen 100 – 200 pembacaan setiap detik dan akan mempunyai 200 – 2000 *gate* yang tersedia untuk *security*. Banyak implementasi komersial dari *hash functions* mengoptimalkan kecepatan, melebihi jumlah *gate*. Sistem RFID membolehkan fleksibilitas yang besar dalam hal ini. Mungkin 10.000 *clock cycles* mungkin tersedia untuk fungsi *security*. Relatif kelebihan dari *clock cycles* menganjurkan menggunakan beberapa *gate*. *Cycles* yang banyak seperti prinsip desain akan mempertimbangkan *low-cost hash designs*.

#### 3.3.1 Definisi Hash

Pada minimum fungsi *hash h* adalah efisiensi perhitungan fungsi yang memetakan kewenangan panjang input untuk menetapkan panjang output. Yaitu  $h : \{0,1\}^* \rightarrow \{0,1\}^n$ . Banyak hal sepele fungsi jatuh karena kehilangan definisi. Menurut tiga *properties* mencoba untuk lebih berguna :

- a) *Preimage resistance* – untuk semua output  $y$ , adalah perhitungan yang mungkin untuk menemukan beberapa input  $x$  seperti  $h(x) = y$  memberikan tidak ada hubungan input yang diketahui.
- b)  $2^{\text{nd}}$ -*preimage resistance* – memberikan  $x$ , adalah perhitungan yang tidak mungkin untuk menemukan  $x' \neq x$  seperti  $h(x) = h(x')$ .
- c) *collision resistance* – adalah perhitungan ketidak mungkinan untuk menemukan beberapa perbaikan dari input  $x$  dan  $x'$  seperti  $h(x) = h(x')$ . Catatan bebas memilih kedua input.

*One way hash function* (OWHF) adalah fungsi yang menawarkan *preimage* dan  $2^{\text{nd}}$  – *preimage resistance*. Hal ini memungkinkan gagasan sederhana menjadikan “sulit untuk membalikan” (*difficult to invert*). *Collision resistant hash function* (CRHF) adalah fungsi *hash* yang mana  $2^{\text{nd}}$  –*preimage resistant* dan *collision resistant*. Meskipun tidak dibutuhkan sebagian besar CRHF adalah salah satu jalan dalam praktik. Meskipun sulit untuk membalikan *one way hash function* tidak memerlukan menyembunyikan informasi. Sebagai contoh, andaikata diberikan *one way hash function*  $h'$ . Definisi *hash function* kedua  $h(x||y) = h'(x)||y$ . Output dari  $h$  adalah sulit untuk dikembalikan (*invert*) berdasarkan kebalikan dari  $h'$ . Meskipun setengah kedua dengan jelas kebocoran informasi tentang input. Dalam konteks RFID mengingat *randomized hash lock schema* dibagian 3.2 di bawah sekema ini pasangan  $(R, h(R||ID))$  dikirim oleh *tag*. Jika  $h$  didefinisikan seperti di atas nilainya menjadi  $(R, h'(R)||ID)$ . Kebocoran informasi ini dengan jelas takluk bertujuan menggunakan fungsi *hash* dalam tempat pertama. Dalam teori *randomized hash lock* dapat mengandalkan pada *pseudo-random function* atau *perfect one way function* [16]. Walaupun pada teori ini tidak aman, *heuristic hash function* cukup menyembunyikan informasi pada praktik.

### 3.3.2 Desain Pendekatan

Beberapa variasi pendekatan untuk membangun fungsi *hash* yang telah digunakan dalam praktik. Banyak pendekatan ini dihindarkan berharga mahal untuk *low-cost tag* (*tag* berharga murah). Dua kelompok utama *hash* dengan teori yang sangat dibutuhkan seperti *hash* berdasar pada *modular aritmatik* atau *NP-Completeness*, dan

*heuristic hash* yang digunakan dalam praktik. Pada bagian ini analogi dengan perbedaan antara *public key* dan *symmetric cryptosystem*. Pada kenyataanya keduanya adalah contoh masing masing kelas *hash* yang mana mengandalkan pada dibawah *public-key* dan *symmetric primitives*.

Teori dasar *hash* meliputi keduanya berdasarkan pada *modular* aritmatik, aljabar matrik dan “*hard*” problem seperti *Knapsack problem*. Modular aritmatik dasar *hash* mengandalkan pada dasar faktorial yang kuat. Atau menemukan algoritma diskrit dalam medan Galois. *Cryptosystems* adalah didasari pada permasalahan yang sama. Intisari ukuran dari matematika modular dasar dari *hash* tergantung pada ukuran modulus. Kecukupan moduli yang lebar akan memberikan banyak keleluasaan ruang yang tersedia pada *tag* RFID. Operasi aritmatik modular terlalu banyak perhitungan yang intensif dan akan banyak memerlukan banyak *gate* untuk mengimplemntasikan pada *low-cost tags*. Mengandalkan di bawah kekerasan lingkaran, kisi kisi atau kurva elip tidak banyak memberikan banyak janji untuk *tag* untuk masa yang akan datang .

Teori keamanan *hash* dapat juga berdasarkan dasar pada aljabar matrik . Sebagai contoh diberikan  $n \times n$  *secret matrix*  $K$  , pesan *hash*  $M$  dapat didefinisikan seperti  $H(M) = M^tKM$ . Kerugiannya, pesan *hash* sebagai contoh : 128 bit akan memerlukan ukuran kunci matrik mendekati 512 byte. Ukuran operasi matrik melebihi apa yang memungkinkan untuk *low-cost tag*. Meskipun berbasis matrik kecil, atau *S-box* atau *sub-hash function* dapat digunakan untuk membangun blok dalam desain fungsi *hash*. Kelas ketiga *hash* mengandalkan pada *hardness of Knapsack problem*. *Knapsack problem* adalah keadaan sebagai berit : Diberikan bilangan integer  $S = S_1, \dots, S_k$  dan integer  $n$ , tentukan beberapa subset  $T \subseteq S$  seperti  $\sum_{t_i \in T} t_i = n$ . Perhitungan yang berat dan kelengkapan penyimpanan *Knapsack* berbasis *hashes* adalah juga melebihi sumber *low-cost tag*.

*Heuristic hashes* mungkin dapat menyumbangkan fungsi atau mungkin mengandalkan pada *block cipher* (pesan dibagi ke dalam block, dan block terakhir di padding ke ukuran standard yang digunakan, dan setiap block dienkrrip secara independent. Block pertama tersedia untuk transmisi setelah enkripsi selesai)[5] seperti dasar pembangunan *block*. Karena berasumsi bahwa *block chipper* akan

menjadi terlalu mahal untuk diimplementasikan, Saat ini bukan suatu pilihan yang memungkinkan. Walaupun demikian hal *block chipper* akan menjadi efisien penggunaan sumber dayanya jika desain masa datang dapat digunakan *block chipper* kedua *hashes* dan enkripsi.

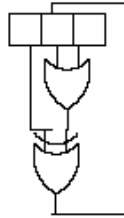
Dalam serangan tidak ditemukan *collision* dalam Message Digest 5 (MD5) dapat ditemukan di RFCs 1319-1321, panjang digest :128 bit [5], mungkin penting dalam step pertama.

satu lagi yang lain umum digunakan penampilan fungsi *hash*. Umumnya, algoritma ini telah digunakan untuk mengoptimalkan kecepatan dan kemudahan mengimplementasi perangkat lunak. Kebanyakan implementasi kecil biaya sumber disediakan RFID. Dua jalan yang mungkin cocok untuk system *low-cost* yaitu : *Cellular Automata* dan *Non-Linear Feedback Shift Registers* yang akan dibahas dibagian 3.3.3 dan 3.3.4.

### 3.3.3 Cellular Automata

*Cellular Automata* (CA) adalah terbatas pada keadaan mesin yang memiliki keadaan transisi yang tergantung berdiri sendiri (solely) saling berdekatan satu dengan yang lain. Sistem *cellular automata* yang sederhana adalah *binary* dan satu-demensi. Tiap tiap *state cell* tergantung pada *state* miliknya sendiri dan hal itu langsung berdekatan. Maka *cell* milik *i* bernilai dalam *step t +1* adalah  $C_{i,t+1}$ , tergantung pada  $C_{i-1,t}, C_{i,t}, C_{i+1,t}$ . Karena tiap delapan kemungkinan *state* mempunyai dua kemungkinan output, di sana 256 total binay *one-dimensional CA system*. Banyak dari sistim ini mempunyai tingkah laku yang dapat diperiksa. Meskipun ada beberapa mempertunjukan properties sifat acak (*random*) atau kacau (*chaotic*). Pada keterangan, Wolfram menganalisa fakta-fakta cell dari CA Rule #30, ditentukan seperti  $C_{i,t+1} = C_{i-1,t} \oplus (C_{i,t} \vee C_{i+1,t})$ , diimplementasikan dalam *cyclic register*.

Register yang berukuran *n* secara kasar memerlukan  $2n$  *logic gates* untuk mengimplemtasikan CA. Gambar 3.6 menunjukan diagram dari register yang diimplementasikan dalam Rule #30 pada *single cell*.



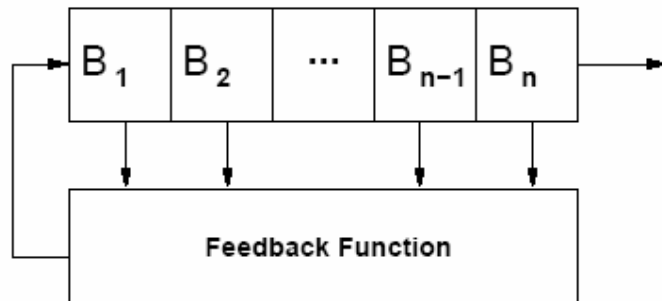
Gambar 3.6 Implementasi CA pada *single cell* dengan ukuran tetap *cyclic register*<sup>[2]</sup>

Cellhash dan CA-berbasis hash yang lain mungkin merupakan paradigma yang cocok untuk *tag* RFID yang berbea murah. Skala *CA hash* dan juga intisari pertambahan ukuran *hash*. Keuntungan yang lain bahwa *tag* mungkin siap mempunyai register digunakan untuk *anti-collison* yang mana juga digunakan dalam *hashing*.

Satu kekurangan dari *CA hashes* pada *tag* RFID adalah banyak perhitungan paralel yang mungkin membutuhkan terlalu banyak tenaga (*power*). Untungnya CA mungkin mempunyai kemampuan diseri pada peningkatan performen. Disana tidak jelas jalan untuk mengurangi *cycles*. Pada kenyataannya CA perhitungannya tidak dapat diperkecil. Output mereka hanya dapat ditemukan dengan langkah nyata melewati tiap tiap perhitungan. Reader mengimplementasikan kunci *random hash* menggunakan *CA hash* yang akan beroperasi (*run*) melewati banyak *cycle* untuk tiap satu dari *tag* yang mereka ketahui. Sudah tentu hal ini kelemahan yang melekat *randomized hash lock*.

### 3.3.4 *Non-Linear Feedack Shift Registers*

Umpan balik pergeseran register terdiri dari pergeseran register dan fungsi umpan balik. Bit khusus dalam pergeseran register “disadap” (*tapped*) dan diumpankan kedalam fungsi umpan balik setiap waktu output dibutuhkan. Fungsi umpan balik akan mengeluarkan bit yang mana input kembali kedalam register seperti pada Gambar 5.8

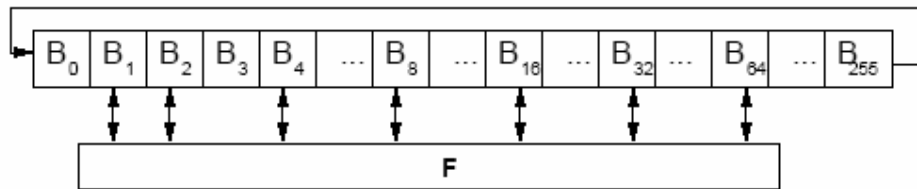


Gambar 5.8 *Feedback shift register*<sup>[2]</sup>

Fungsi umpan balik mungkin linier atau non linier. Struktur hasil ditunjuk sebagai *Linear Feedback Shift Register (LFSR)*. Fungsi umpan balik yang lebih komplit dapat digunakan. Sebagai contoh *multiple bit* dapat dipindahkan dengan *substitution box* atau (S-Box), atau *permutation box* (P-Box). Sama pada *cellular automata*, *low hash fuction* akan mengandalkan pada fungsi umpan balik sederhana ditambahkan berkali-kali.

Dua konsep berguna dalam *heuristic hash design* dikenalkan oleh Claude Shannon lebih 50 tahun lalu. Konsep tersebut adalah *confusion* dan *diffusion*. *Confusion* adalah sifat dari hubungan statistik diantara input hash dan output hash dapat menjadi komplek untuk musuh yang luar biasa (berani). *Diffusion* adalah sifat bahwa yang mempengaruhi input single bit yang akan disebarkan diantara bit yang banyak pada output. Dengan kata lain pola prediksi input yang disembunyikan dalam output.

Mengambil konsep ini dalam keadaan dari NLFSR desai berbasis hash, harapannya bahwa kesulitannya fungsi umpan balik dapat menciptakan *confusion*. Penambahan fungsi melebihi *cycles* dan pada posisi yang tepat pada titik ketukan (*tap point*) akan membantu mempengaruhi bit input. Kita menawarkan pendekatan desain yang cocok dari P-box kecil, dengan *tap point* pada dua power. P-box ini akan beroperasi pada register kerja pada dua kali ukuran akhir *hash out put*. Setelah penambahan banyak kali, setengah register kerja akan dibuang. Sisa setengahnya akan menjadi output fungsi *hash* seperti terlihat pada Gambar 3.9



Gambar 3.9 Usulan Arsitektur NLFSR<sup>[2]</sup>

Gambar 3.9 gambaran *hash* berbasis NLFSR dengan memendekkan ukuran 128 bit. Fungsi F adalah 7 bit *permutation*. Dalam iterasi tunggal bit 1, 2, 3, 4, 8, 16, 32 dan 64 disadap dan input kedalam F. Hasil output ditulis kembali pada bit yang sama. Akhirnya register digeser ke kanan. Setelah iterasi jumlah besar, bagian setengah register yang kedua dipotong.. Sisanya adalah *hash output*.

Sebelum dipotong, fungsi *hash* adalah *permutation*. Pembalikan *hash* sedikit memerlukan bekerjanya fungsi dalam membalikan. Ini menjamin bahwa informasi tidak hilang selama iterasi, yang mana akan batas jarak hasil akhir. Paling sedikit dua *tap point* pada bit 1 dan bit 2 menjamin bahwa setiap bit input dapat mempengaruhi setiap bit output. Penambahan *tap point* dimaksudkan memperpanjang setiap bit lebih cepat.

Pemilihan dari 7 *tap point* adalah tepat sekali pada pembatasan sumber daya. Tujuh ke tujuh *permutation* memerlukan melihat tabel dari 128 x 7, yang mendekati 900 bit. Standar permutasi dapat di-*hardware*-kan kedalam tag relatif murah. Diasumsikan setiap iterasi mengambil dua putaran, fungsi ini dapat diiterasi sekitar 5000 kali. Dua ratus lima puluh enam (256) bit register akan digeser melewati kompleksitas sekitar 20 kali. Hal ini *diffuse* dapat diulang pada masing masing bit input kejadian melebihi output.

Hal ini adalah salah satu contoh suatu kemungkinan desain *low-cost* Fungsi F, *tap point* dan ukuran register mungkin dapat bervariasi. Mungkin hanya asumsi yang mana tidak realistis bahwa tag akan mempunyai pergeseran register 256 bit. Saat ini tag hanya mempunyai 96 bit yang hanya dapat dibaca memarynya. Meskipun hal tersebut sudah standar untuk dibangun 128 bit dan 256 bit EPC tag

Sama seperti *CA hashes* mempunyai problem yang potensial dengan NLFSR berbasis *randomized hash lock* adalah *reader* akan memerlukan perhitungan hash output untuk setiap *tag* yang diketahui. Tidak sama dengan sistem, NLFSR dapat diparalel pada sisi *reader* untuk penawaran yang performen yang lebih besar. Diasumsikan bahwa *reader* akan mempunyai sumber yang besar untuk membandingkan *tag*. Paralelisasi NLFSR hal yang perlu dalam menjualkan jumlah *gate* yang besar untuk mempercepat performen kecepatan

### **3.4 Secure Anti Collision**

Persamaan algoritma *Tree-walking anti collision* yaitu ketika *tag* merespon bersamaan pada *query reader*, konflik signal komunikasi menyebabkan interferen. Mempunyai cacat keamanan yang mempunyai hak asymmetry diantara forward dan backward channel. Frekuensi operasi sudah ditentukan, jarak jangkauan penyadap dapat memonitor transmisi dari jarak 100 meter dan mendapat isi setiap tag

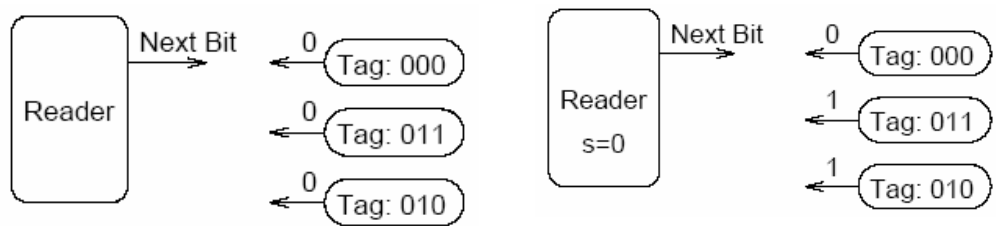
#### **3.4.1 Blinded Tree-Walking**

Presentasi variasi dari kesamaan dengan *tree-walking* yang mana tidak menyiarkan (*broadcast*) *tag* ID yang tidak aman pada *forward channel* dan tidak mempengaruhi performen. Skema yang asli nampak dengan nama “*Silent Tree-Walking*”. Dengan assumsi populasi *tag* digunakan bersama pada awalan ID umum, seperti kode produk atau *manufacturer ID*. *Tag* tunggal, *reader* meminta semua *tag* untuk menyiarkan (*broadcast*) bit mereka selanjutnya. Jika tidak ada *collision* maka tag menggunakan bersama nilai dalam bit.

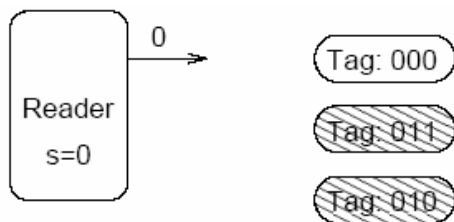
Penyadap yang jaraknya jauh hanya dapat memonitor pada *forward channel* dan tidak akan mendengar *respon tag*. Jadi *reader* dan *tag* secara efektif menggunakan bersama bit bernilai rahasia. Ketika *collision* terjadi, *reader* memerlukan spesifikasi yang mana porsi populasi *tag* dapat diproses. Jika tidak ada *collision*, *reader* akan lebih sederhana meminta bit berikutnya, karena semua *tag* menggunakan nilai yang sama untuk bit sebelumnya. Karena kita berasumsi tag menggunakan bersama awalan yang umum, *reader* dapat menghasilkan rahasia yang digunakan bersama pada *backward channel*. Penggunaan bersama awalan kode rahasia dapat digunakan

untuk menyembunyikan nilai unik bagian ID. Andaikata kita mempunyai dua tag dengan nilai ID  $b_1b_2$  dan  $\overline{b_1b_2}$ . Reader akan menerima  $b_1$  akan menerima kedua tag tanpa *collision*, dan akan mendeteksi *collision* pada bit berikutnya. Karena  $b_1$  rahasia dari penyadap yang berjarak jauh, *reader* dapat mengirim dua  $b_1 \oplus b_2$  or  $b_1 \oplus \overline{b_2}$  pada keinginan tunggal tanpa menyatakan bit yang lainnya. Gambar 3.10 ilustrasi performen blinded tree-walking pada dua bit

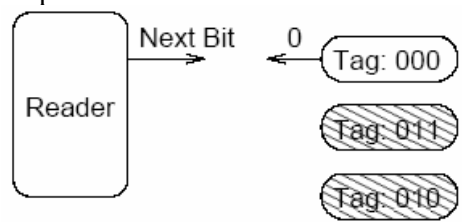
Penyadap dalam jarak *backward channel* akan jelas menghasilkan segala ID. Meskipun skema *blinded tree-walking efektif* proteksi dari serangan penyadap jarak jauh dari *forward channel* dengan sedikit penambahan kompleksitas. Performen adalah identik pada reguler *tree-walking*, karena *tag* akan disatukan ketika hal itu telah disiarkan dalam ID pada *backward channel*.



1. Reader meminta bit pertama.  
Tidak ada collision



2. Reader menyimpan bit rahasia pertama dan mendeteksi collision pada bit kedua



3. Reader memilih tag dengan 0 dalam posisi bit kedua oleh pengirim ( $0 \text{ XOR } s$ ). Tag yang lain tidak aktif

4. Reader menyatukan sisa tag

Gambar 3.10 *Reader* menyatukan *Tag* 000 dengan algoritma *blinded tree-walking*<sup>[2]</sup>

### 3.4.2 Randomized Tree-Walking

Dalam skema ini, sebagian besar bit *pseudo-ID* akan disiarkan (*broadcast*) melewati *forward channel*. Tidak diasumsikan tentang *tag* awal atau manajemen kunci rahasia pada sisi *reader* diperlukan. Randomisasi mendatangkan resiko penambahan biaya komunikasi. Performennya dapat diperdagangkan dengan sedikit kebocoran. Diminta jaminan bahwa tidak ada *tag* yang dilihat berkali-kali, *reader* harus memperbaiki *power* untuk *tag* yang dekat sampai selesai dibaca seluruhnya. Hal yang mendasar *tag* akan mempunyai *pseudo-ID* baru tiap bagian *tree-traversal*. Sekali power terputus, semua *tag* akan melupakan *pseudo-ID* mereka.

Bit *pseudo-ID* mungkin sebenarnya dibangkitkan selama bekerja. *Reader* akan meminta semua *tag* untuk bit yang acak. Jika *collision* terdeteksi. Ia akan langsung semua bit khusus akan tertidur (tidak aktif). *Tag* harus menjaga *track* dari posisi bit yang mereka ambil untuk tidak aktif. Setelah tidak ada *collision*, *reader* akan diyakinkan bahwa *tag* telah menyatu. Dalam kejadian yang tidak sama bahwa dua *tag* menciptakan bit random yang sama, *reader* akan masih dapat mendeteksi *collision* pada ID reguler. Probabilitas kejadian dapat membuat kewenangan kecil pada performen biaya komunikasi.

```
Traverse(i, count)
   $b_i :=$  Read random bit i from all active tags.
  if collision on  $b_i$  is detected:
    Suspend all tags with  $b_i == 1$ .
    Each suspended tag stores i.
    Traverse(i+1, 0).
    Wake up all tags suspended on bit i.
    Traverse(i+1, 0).
  else if no collision on  $b_i$  is detected:
    if (count > threshold) Tree-Walk remaining tags.
    else Traverse(i+1, count+1).
```

Gambar 3.11 Algoritma *Randomized Tree-Walking*<sup>[2]</sup>

Algoritma *traversal reader* terlihat pada Gambar 3.11. Fungsi “*Traverse*” menerima dua argumen : *i* adalah aliran posisi bit dan *count* adalah jumlah bit yang berurutan

tampa *collision*. Jika *count* melebihi beberapa prioritas ambang pintu, *reader* akan diasumsikan *tag* telah disatukan dan akan mencoba membaca ID.

Kerugian utama dari *randomized tree-walking* adalah penambahan biaya komunikasi dari *pseudo-ID*. *Recall* bahwa kedua *reguler* dan *blinded tree-walking* hanya menyiarkan panjang ID *tag*. Isu minor yang lain bahwa *tag* akan memerlukan bit yang kecil *state* akan menahan *track* dari ketika kita menunda.

Pilihan panjang *pseudo-ID* tergantung pada ukuran kedekatan populasi. Untuk populasi dari  $n$  tag, andaikan  $m$  *pseudo-ID* bit yang digunakan. Jumlah *tag* random memilih secara khusus *pseudo-ID* akan menurut Poisson distribution.

Jika  $\lambda = \frac{n}{2^m}$  [2], diharapkan jumlah dari *pseudo-ID* dengan  $k$  tag akan sekitar  $2^m e^{-\lambda} \frac{\lambda^k}{k!}$ . [2] andaikata  $n = 2000$  dan  $m = 16$ , maka  $\lambda = .03$ . Diharapkan jumlah *pseudo-ID* dengan  $k$  diberikan pada tabel 3.1 2000 tag dengan 96 bit ID berisi mendekati 192.000 bit data. Collision akan bocor mendekati 30 bit dari data pada masing masing *tree traversal*. Hal ini dapat diterima dalam banyak aplikasi, meskipun melebihi banyaknya *tree traversal* kebocoran akan dapat bertambah. Pilihan panjang *pseudo-ID* dan bagaimana menganangani collision *pseudo-ID* dapat ditinggalkan pada tag user tergantung pada privasi khusus dan keperluan perfoemen.

Tabel 3.1 perkiraan distribusi dari 2000 tag melalui random 16 bit *pseudo-ID*[2]

k	Pseudo-IDs with k Tag	Comment
0	63599	Sebagian besar <i>pseudo-ID</i> tidak akan diseleksi
1	1907	Sebagian tag akan membangkitkan <i>pseudo-ID</i> unik
2	28	Beberapa pasangan akan bertubrukan pada <i>pseudo-ID</i> yang sama
3	0,29	Lebih dua tag akan jarang bertubrukan

### 3.5 Usulan keamanan RFID lainnya

#### 3.5.1 *Asymmetric Key Agreement*

*Reader* dapat mengambil keuntungan dari *asymmetry* dari *forward* dan *backward channel* mentransmit nilai yang sensitif seperti *key*. Andaikata *reader* memerlukan untuk mentransmit nilai  $v$  pada *tag* tunggal. Bahwa *tag* dapat membangkitkan nilai random  $r$  seperti satu bagain waktu dan transmit nilai tersebut dalam keadaan bersih pada *backward channel*. *Reader* sekarang dapat mengirim  $v \oplus r$  melewati *forward channel*. Jika penyadap berada diluar *backward channel*, mereka hanya dapat mendengar  $v \oplus r$ , dan  $v$  secara teori akan aman.

#### 3.5.2 *Chaffing dan Winnowing (melukai dan memisahkan)*

Penangkis lain penyadap pada *forward channel* adalah menyiarkan (*broadcast*) aba-aba “*chaff*” dari *reader*. bermaksud membingungkan atau memisahkan informasi yang terkumpul dari penyadap. Dengan negosiasi penggunaan rahasia secara bersama, aba-aba (tanda) dapat difilter atau dipisahkan “*winnowed*” oleh *tag* menggunakan MAC sederhana. [2]

#### 3.5.3 Unit Pendeteksi (Detection Unit)

RFID dan lingkungannya memungkinkan dilengkapi dengan peralatan untuk mendeteksi yang tidak berhak (*unauthorized*) membaca atau anomali transmisi pada operasi frekuensi *tag*. Hak untuk signal yang kuat, mendeteksi *reader* lain adalah perkara sederhana. Tergabung pada permintaan yang tidak berhak dan mendeteksi “*Jamming*” unit dalam pasangan *smart* akan membantu mendeteksi dan identifikasi serangan *denial of service* dalam seting perdagangan.

#### 3.5.4 Jeritan Tag (*Screaming Tag*)

Unit untuk mendeteksi serangan mungkin dapat diperluas untuk mendeteksi jika *tag* dilumpuhkan, ide ini ditulis oleh Sarma yaitu mendesain *tag* akan menjerit “*screaming*” ketika dilumpuhkan (*killed*). Jeritan dapat berupa sinyal penuh pada frekuensi yang telah ditentukan. Penggambungan pendeteksi jeritan (*scream*

*detection*) kedalam rak *smart* lebih lanjut membantu mendeteksi serangan *denial of service*

### **3.5.5 Agent Security**

Unit pendeteksi dapat digabungkan untuk *standard* masa depan pada semua RFID *reader* mungkin lengkap ke dalam telepon seluler atau PDA. Peralatan yang legitimet (Sah) dapat mendeteksi, *log* dan *filter reader* lain yang mencoba melakukan *query*. *Reader* dapat beraksi melewati pembatas untuk membaca. Pada intinya *reader* dan *tag* akan menjadi *Wireless Personal Area Network* (WPAN). *Reader* dapat beraksi menjembatani antara WPAN dan dunia luar Menempatkan perlengkapan menyamai isi *tag* yang saling berdekatan, mengeliminasi keperluan untuk *query tag* pada semua dan memperpanjang jarak efektifitas.

### **3.5.6 Mencetak Master Key**

Memungkinkan *end user* (pemakai) untuk mengakses fungsi dari tambahan imbuhan item *tag* yang telah dibeli, kunci master dapat dicetak dalam pembungkus produk, mungkin seperti *barcode* atau bilangan desimal. Setelah pembelian item, konsumen dapat menggunakan *master key* membuka *tag* dari kunci *hash* (*hash lock*). *Master key* juga dapat berfungsi seperti mengembalikan mekanisme kunci. membolehkan *user* untuk mengunci *tag* apabila mereka kehilangan kunci. Karena *master key* harus dibaca secara *optical* dari dalam pembungkus, musuh tidak dapat mendapatkan *tag* tanpa mendapatkan paket itu sendiri. Untuk keamanan lebih lanjut, semua fungsi yang menggunakan *master key* dapat dijadikan syarat untuk menggunakan *physical contac channel* , dari pada *RF*.

## BAB IV PENUTUP

RFID yang dapat diramalkan akan menggantikan pemakain *optical barcode*, karena RFID mempunyai beberapa keuntungan yaitu kemungkinan data dapat dibaca secara otomatis tanpa memperhatikan garis pembacaan, Dapat melewati bahan non konduktor dan mempunyai kecepatan pembacaan beberapa ratus *tag* per detik dengan jarak dapat mencapai 100 meter.

*Tag* RFID mempunyai kemampuan identifikasi yang disatukan dengan dalam satu desain. Penyebaran RFID yang universal akan memudahkan timbulnya ancaman, keamanan maupun gangguan privasi.

Gangguan yang kemungkinan merupakan ancaman untuk RFID antara lain.

- Serangan secara fisik yaitu dengan cara *tag* diambil, ditukar, digores.
- Serangan aktif tidak secara fisik yaitu ikut serta dalam protokol atau menyamar sebagai pemilik atau *reader* yang sah, dengan melakukan query seperti *reader* sesuka hati, dapat memodifikasi atau pengubahan isi dari *tag* RFID.
- *Eavesdropping* (penyadapan) yaitu mendapatkan duplikasi pesan tanpa ijin.
- *Replaying* menyimpan pesan yang ditangkap untuk digunakan pada pemakaian pengguna lain.
- Serangan *Denial of Service* membajiri saluran atau sumber lain dengan pesan yang bertujuan untuk menggagalkan pengaksesan pemakain lain.

Penanganan keamanan perlindungan terhadap *reader* tak legal, autentikasi jaminan identitas *tag*, pencegahan treking *tag* oleh reader yang tak sah, serangan terhadap *Denial of service*. Ditawarkan mekanisme keamanan *low cost access control* didasari pada *fungsi hash*, *Randomized hash lock*, *Screaming Tag*, pendeteksi adanya serangan *Denial of service*.

## DAFTAR PUSTAKA

- [1] Weis, Stephen A; Sarma, Sanjay E; Rivest, Ronald I and Engeles, Daniel W, Security and Privacy Aspect of Low-Cost Radio Frequency Identification System, Laboratory for Computer Science Massachusetts Institute of Technology Cambridge, MA 02139 USA; Auto-ID Center Massachusetts Institute of Technology Cambridge, MA 02139 USA, <http://www.eicar.org/rfid/kickoffcd/04%20-%20Hintergrundinformationen/09%20-%20RFID%20Systems%20and%20Security%20and%20Privacy%20Implications.pdf>, 29 September 2004 jam 10.30.
- [2] Weis, Stephen August, *Security in Radio-Frequency Identification Devices*, Massachusetts Institute of Technology, May 2003, <http://www.eicar.org/rfid/kickoffcd/04%20-%20Hintergrundinformationen/13%20-%20Security%20and%20Privacy%20in%20RFID%20devices.pdf>, 29 September 2004 jam 10.30
- [3] Gildas AVOINE, *Security Issues in RFID Systems, Seminar on Security Protocols and Applications 2004*, <http://lasecwww.epfl.ch/gavoine/rfid/> 29 September 2004 jam 10.30
- [4] Rivas, Mario, *RFID – its Applications and Benefits*, Philips, 2004
- [5] Budi susanto, *Jaringan Komputer Keamanan Jaringan Modul 12*, .....
- [6] Juels, Ari, *Minimalist Cryptography for Low-Cost RFID Tags*, RSA Laboratories, Bedford, MA 01730, USA, 5. <http://www.eicar.org/rfid/kickoffcd/04%20-%20Hintergrundinformationen/11%20-%20Minimalist%20Cryptography%20for%20RFID%20Tags.pdf>, 29 September 2004 jam 11.00
- [7] Juels, Ari and Brainard, John, *Soft Blocking: Flexible Blocker Tags on the Cheap*, RSA Laboratories Bedford, MA 01730, USA e-mail: {ajuels,jbrainard}@rsasecurity.com, <http://www.eicar.org/rfid/kickoffcd/14%20%Soft%Blocking%20%20Flexible%20Blocker%20tags%0n%20%on%20%20Cheap.pdf>, 30 September 2004
- [8] Fadila Mutiarwati, *Smart Label Pengganti Barcode*, *PC Media*, Januari 2004