Analisis Formal *Lightweight Mutual Authentication* RFID *Protocol* Menggunakan Scyther

Dinda Putri Adra Kusuma[1)], Annisa Dini Handayani[2)]

(1) Rekayasa Keamanan Siber, Politeknik Siber dan Sandi Negara, dinda.adra@gmail.com

(2) Rekayasa Keamanan Siber, Politeknik Siber dan Sandi Negara, annisa.dini@bssn.go.id

Abstrak

Salah satu faktor penting dalam protokol komunikasi adalah adanya autentikasi. Autentikasi merupakan suatu metode untuk memastikan bahwa entitas yang berkomunikasi merupakan entitas yang benar. Protokol yang memanfaatkan autentikasi salah satunya adalah protokol yang ada dalam Radio Frequency and Identification (RFID). RFID merupakan teknologi yang digunakan untuk melakukan identifikasi dan pengambilan data menggunakan gelombang radio. RFID menggunakan beberapa komponen dalam penerapannya, yaitu tag, reader, dan server (database). Dalam implementasi protokol pada RFID, dibutuhkan mekanisme autentikasi antar entitas yang berkomunikasi. Salah satu protokol RFID yang telah menerapkan mekanisme autentikasi adalah Lightweight Mutual Authentication RFID Protocol yang diajukan oleh Kang. Protokol milik Kang terdiri dari 3 varian, yaitu Single-tag, Double-tag, dan Multi-tag. Ketiga varian protokol ini diklaim aman terhadap serangan replay, relay, dan eavesdropping, serta memenuhi anonimitas tag dan forward security. Meskipun demikian, klaim keamanan tersebut belum dilengkapi dengan tingkat autentikasi yang dipenuhi oleh protokol milik Kang. Oleh karena itu, pada penelitian ini dilakukan pengujian pemenuhan tingkat autentikasi pada protokol Kang, khususnya varian Single-tag dan Double-tag menggunakan alat uji Scyther. Tingkat autentikasi yang diuji terdiri dari aliveness, weak-agreement, non-injective agreement, dan non-injective synchronization. Hasil penelitian ini, menunjukkan bahwa protokol Kang memenuhi keempat klaim keamanan dan tanpa serangan untuk varian Single-tag, serta memenuhi keempat klaim keamanan dan tanpa serangan dalam batasan untuk varian Double-tag.

Kata kunci: klaim autentikasi, *Lightweight Mutual Authentication* RFID *Protocol*, protokol autentikasi, RFID, Scyther

1. PENDAHULUAN

Radio Frequency and Identification (RFID) adalah alat identifikasi otomatis yang menggunakan gelombang radio untuk mendeteksi, melacak, mengidentifikasi, dan mengelola berbagai objek. Tujuan dari sistem RFID adalah mentransmisikan data dari perangkat portabel, yang disebut *tag*, ke RFID *reader* untuk menjalankan aplikasi tertentu [1]. Sistem RFID memiliki tiga komponen utama yaitu *tag*, *reader*, dan sistem *back-end*. *Tag* RFID adalah perangkat identifikasi yang dipasang ke objek. *Reader* RFID adalah perangkat yang digunakan untuk berkomunikasi dengan *tag*. Sistem *back-end* terkadang disebut sebagai *online database* yang bertujuan untuk mengumpulkan, menyaring, memproses, dan mengelola data RFID [1].

Hal terpenting dari sistem RFID adalah masalah keamanan. Seperti sistem informasi lain, RFID juga rentan terhadap berbagai serangan [1],[2]. Serangan terhadap sistem RFID antara lain serangan *replay*, *man-in-the-middle*, *cloning*, dan *denial of service* [3]. Salah satu cara untuk mengatasi serangan-serangan tersebut, adalah dengan mengajukan beberapa rancangan protokol keamanan, khususnya protokol autentikasi pada RFID.

Penelitian terkait protokol autentikasi pada RFID telah banyak dikembangkan. Misalnya, pada tahun 2004 Juels mengusulkan protokol yang efisien untuk melakukan autentikasi beberapa *tag* sekaligus pada RFID dengan metode pembuktian Yooking dan memanfaatkan MAC [4]. Kelemahan dari protokol ini adalah adanya kerentanan terhadap serangan *replay* [5]. Berdasarkan kelemahan tersebut, pada tahun 2007 Chien mendesain protokol berbasis komputasi *tag* [6]. Tidak lama setelah itu, Lopez menemukan empat kelemahan dari protokol milik Chien, yaitu tidak menjamin identifikasi yang jelas dari *tag*, tidak menjamin keamanan identitas pengguna, rentan terhadap peniruan *database back-end*, dan tidak menjamin kerahasiaan *tag* [7].

Kemudian pada tahun 2016, Kang mengajukan desain protokol yang bertujuan untuk memperkuat keamanan sistem dan meringankan proses enkripsi RFID konvensional. Protokol yang diajukan disebut *Lightweight Mutual Authentication* RFID *Protocol*, yang terdiri dari tiga skema, yaitu protokol autentikasi *Single-tag*, *Double-tag*, dan *Multi-tag* [8]. Ketiga skema tersebut dinyatakan aman terhadap berbagai ancaman keamanan dalam hal komunikasi RFID karena desainnya dibuat untuk memenuhi autentikasi *multiple tag* berdasarkan *Electronic Product Code Class* 1 *Generation* 2 (EPC C1G2) [8]. EPC C1G2 merupakan standar mengenai teknologi RFID yang dikeluarkan oleh organisasi EPCglobal [9]. Berdasarkan hal tersebut, skema yang diajukan oleh Kang berfokus pada keamanan autentikasi. Salah satu alat yang dapat digunakan untuk analisis protokol berdasarkan klaim keamanan autentikasi adalah Scyther. Scyther merupakan alat verifikasi dan

deteksi kelemahan pada suatu keamanan protokol. Aplikasi ini menggunakan input deskripsi protokol dan parameter opsional yang sesuai protokol untuk menghasilkan grafik dari setiap serangan [10].

Verifikasi pada Scyther dapat digunakan untuk memeriksa klaim kerahasiaan dan autentikasi. Kerahasiaan adalah bentuk klaim keamanan untuk meverifikasi bahwa informasi tertentu yang dikirim tidak dapat diambil oleh musuh atau penyerang. Autentikasi merupakan bentuk klaim keamanan untuk memverifikasi bahwa ada pihak-pihak tertentu yang sedang menjalankan komunikasi [11]. Klaim autentikasi terdiri dari *aliveness, weak-agreement, non-injective agreement*, dan *non-injective synchronization* [12].

Scyther dikatakan lebih efisien dibandingkan dengan aplikasi analisis lainnya, karena hasil yang ditampilkan berupa grafik sehingga dapat mempermudah analisis formal suatu protokol [10]. Selain itu, aplikasi ini dapat menjamin kebenaran protokol untuk jumlah sesi yang tidak terbatas. Sehingga, alat ini dapat digunakan untuk melakukan analisis formal terhadap skema protokol autentikasi *Single-tag* dan *Double-tag* yang didesain oleh Kang menggunakan Scyther.

2. LANDASAN TEORI

Bab ini membahas tentang studi literatur yang berkaitan dengan penelitian yang dilakukan, diantaranya *Lightweight Mutual Authentication RFID Protocol* yang didesain oleh Kang, alat verifikasi keamanan yaitu Scyther, serta beberapa penelitian terdahulu yang berkaitan dengan penelitian ini.

2.1. *Lightweight Mutual Authentication RFID Protocol*

Kang mengusulkan protokol autentikasi yang diklaim memiliki komputasi yang ringan untuk diterapkan ke RFID dan selanjutnya diberi nama *Lightweight Mutual Authentication* RFID *Protocol*. Protokol ini dibagi menjadi tiga, yaitu protokol autentikasi *Single-tag*, *Double-tag*, dan *Multi-tag* [8]. Penelitian ini akan menganalisis dua protokol yang didesain oleh Kang, yaitu protokol autentikasi *Single-tag* dan protokol autentikasi *Double-tag*.

Untuk mepermudah pemahaman tentang protokol autentikasi *Single-tag* dan *Double-tag*, berikut akan disajikan notasi-notasi yang digunakan dalam proses pertukaran nilai dan pembangkitan kunci sesi pada protokol.

Notasi	Keterangan
T	*Tag*
R	*Reader*
RN_T	Angka acak dari *tag*
RN_R	Angka acak dari *reader*
n	Jumlah rangkaian bitstream
$\{H\}^n$	*Bitstream* H sepanjang 4n
r	Bit n bagian pertama dari $\{H\}^n$
s	Bit n bagian kedua dari $\{H\}^n$
t	Bit n bagian ketiga dari $\{H\}^n$
u	Bit n bagian keempat dari $\{H\}^n$
ID	Angka unik (yang menunjukkan identitas)
ID'	Angka unik hasil XOR antara ID dengan bit *t*
C_i	Bit *challenge* ke-i
d	Bit yang dibangkitkan oleh *reader* untuk membentuk C_i
R_i	Bit *response* ke-i
SK	Kunci sesi
RB_T	Bit redudansi pada proses *challenge-reponse* dari *tag*
$K_{general\ manager}$	Kunci rahasia bersama untuk *general manager*
$K_{object\ class}$	Kunci rahasia bersama untuk *object class*
K_T	Kunci rahasia bersama untuk *tag*
S	Nilai rahasia
a, b, c	Koefisien polinomial
$P1, P2, P3, \ldots . PN$	Titik-titik representasi polinomial

Protokol autentikasi *Single-tag* merupakan protokol yang melibatkan sebuah *tag* dan *reader* dalam proses *challenge-response*, pembuatan kunci sesi, dan autentikasi bilateral berdasarkan metode *secret sharing* [8]. Proses jalannya protokol *Single-tag* adalah sebagai berikut.

1.	$T \rightarrow R$	Hello
2	$R \rightarrow T$	RN_{R_1}
3.	$T \rightarrow R$	RN_{T_1}
4.	T, R	Membangkitkan *bitstream* $\{H\}^n = S - Box(RN_{T_1} \|\| RN_{R_1})$, kemudian membaginya menjadi 4 bagian yang sama yaitu $r, s, t,$ dan u. R membangkitkan bit d untuk membentuk bit C_i. T membangkitkan bit ID' untuk membentuk bit R_{Resp_i}.
5.	$R \rightarrow T$	C_i (*Timer on*)
6.	$T \rightarrow R$	R_{Resp_i} (*Timer off*)
7.	R	Membangkitkan $y = ax^3 + bx^2 + cx + S$, kemudian memilih titik-titik $P1, P2, P3, P4$.
8	$R \rightarrow T$	$P1, P2, P3, P4$.

9.	T	Memperoleh $y = ax^3 + bx^2 + cx + S'$ Menghitung S' untuk mendapatkan S dari titik-titik $P1, P2, P3, P4$ untuk menghitung $SK_1 = S - Box(S \oplus RB_{T_A})$.

Protokol kedua yang diajukan oleh Kang adalah protokol autentikasi *Double-tag*. Secara umum, prosesnya sama dengan *Single-tag*, namun perbedaannya protokol ini terdiri dari tiga pihak, yaitu dua pihak *tag* (*tag A* dan *tag B*) dan satu pihak *reader*. Selain itu, proses autentikasi pada protokol *Single-tag* hanya terjadi satu kali antara *tag* dan *reader*, sedangkan pada *Double-tag* terjadi dua kali yaitu antara *tag A* dengan *reader* dan *reader* dengan *tag B*. Proses dari protokol ini adalah sebagai berikut.

1.	$T_A \rightarrow R$	Hello
2	$R \rightarrow T_A$	RN_{R_1}
3.	$T_A \rightarrow R$	RN_{T_1}
4.	T_A, R	Membangkitkan *bitstream* $\{H\}^n = S - Box(RN_{T_{A_1}} \| RN_{R_2})$, kemudian membaginya menjadi 4 bagian yang sama yaitu $r, s, t,$ dan u. R membangkitkan bit d untuk membentuk bit C_i. T membangkitkan bit ID' untuk membentuk bit R_{Resp_i}
5.	$R \rightarrow T_A$	C_i (*Timer on*)
6.	$T_A \rightarrow R$	R_{Resp_i} (*Timer off*)
7.	$R \rightarrow T_B$	$RB_{T_A} \oplus RN_{R_2}$
8	$T_B \rightarrow R$	$RN_{T_{B_1}}$
9.	$T_B.R$	Membangkitkan *bitstream* : $\{H\}^n = S - Box(RN_{T\,B_1} \| RB_{T_A} \oplus RN_{R_2})$ kemudian membaginya menjadi 4 bagian yaitu bit $r, s, t,$ dan u. R membangkitkan bit d untuk membentuk bit C_i. T membangkitkan bit ID' untuk membentuk bit R_{Resp_i}.
10.	$R \rightarrow T_B$	C_i (*Timer on*)
11.	$T_B \rightarrow R$	R_{Resp_i}(*Timer off*)
12.	R	Membangkitkan $y = ax^3 + bx^2 + cx + S1$, kemudian memilih titik-titik $P1, P2, P3, P4$.
13.	$R \rightarrow T_B$	$P1, P2, P3, P4$.
14.	T_B	Memperoleh $y = ax^3 + bx^2 + cx + S'$ Menghitung S' untuk mendapatkan S dari titik-titik $P1, P2, P3, P4$ untuk menghitung $SK_1 = S - Box(S \oplus RB_{T_A})$.
15.	R	Membangkitkan $y = ax^3 + bx^2 + cx + S_2$, kemudian memilih titik-titik $P5, P6, P7, P8$.
16.	$R \rightarrow T_A$	$P5, P6, P7, P8$.
17.	T_A	Memperoleh $y = ax^3 + bx^2 + cx + S'_2$ Menghitung S'_2 untuk mendapatkan S_2 dari titik-titik $P1, P2, P3, P4$ untuk menghitung $SK_2 = S - Box(S_2 \oplus RB_{T_A})$.

2.2. Alat Verifikasi Scyther

Scyther adalah alat yang digunakan untuk analisis keamanan protokol berdasarkan asumsi kriptografi yang sempurna, artinya jika diasumsikan semua fungsi kriptografi itu sempurna maka musuh tidak akan bisa mempelajari apapun dari suatu pesan terenkripsi kecuali musuh mengetahui kunci dekripsi [13]. Alat ini dapat digunakan untuk menemukan kerentanan dari suatu protokol. Dalam praktiknya, alat ini terbukti dapat menampilkan kerentanan protokol yang direpresentasikan dalam bentuk grafik. Grafik ini diperoleh berdasarkan pemodelan protokol ke bahasa Scyther [10].

Kelebihan dari Scyther yaitu dapat digunakan untuk membuktikan kebenaran protokol dengan jumlah sesi yang tidak terbatas. Alat ini juga berguna untuk memperlihatkan jejak serangan yang dapat terjadi dalam suatu protokol yang akan dianalisis. Selain itu, Scyther juga memfasilitasi analisis multi-protokol [10].

Ada tiga fitur utama yang mendukung proses verifikasi pada Scyther, yaitu:

a. *Verify Protocol* untuk memverifikasi apakah klaim keamanan dalam deskripsi protokol berlaku atau tidak.
b. *Automatic claims*, untuk menghasilkan klaim keamanan secara otomatis sesuai dengan verifikasi protokolnya.
c. *Characterize Roles*, untuk menganalisis protokol dengan menunjukkan karakteristik lengkap.

Verify Protocol dan *Characterize Roles* adalah fitur yang akan digunakan dalam analisis formal pada protokol autentikasi *Single-tag* dan *Double-tag*. Aspek keamanan yang akan diverifikasi menggunakan Scyther adalah autentikasi. Klaim autentikasi yang dapat dibuktikan dengan Scyther, meliputi:

a. *Aliveness*, adalah bentuk autentikasi yang bertujuan untuk menetapkan bahwa mitra komunikasi yang dituju telah mengeksekusi beberapa peristiwa, dengan kata lain pihak yang akan berkomunikasi yakin bahwa mitranya benar-benar ada.
b. Weak-agreement, adalah bentuk autentikasi yang menyatakan bahwa inisiator setuju untuk berkomunikasi dengan responden.
c. Non-injective agreement, adalah bentuk klaim autentikasi yang memastikan bahwa komunikasi kedua pihak harus memiliki suatu nilai yang disetujui dalam pertukaran pesan.
d. *Non-injective synchronization, menunjukkan bahwa semua komunikasi dalam protokol berjalan sesuai dengan spesifikasi* protokol. Kesesuaian eksekusi protokol ditandai dengan urutan *send event* dan *receive event* yang benar.

3. METODE PENELITIAN

Penelitian ini berfokus pada keamanan autentikasi yang dapat dipenuhi oleh protokol autentikasi *Single-tag* dan *Double-tag*. Klaim autentikasi yang digunakan berdasarkan klaim yang didukung oleh alat verifikasi Scyther. Analisis yang dilakukan merujuk pada hasil verifikasi yang ditampilkan oleh Scyther.

Metode yang digunakan dalam penelitian ini adalah studi pustaka dan konfirmatori. Studi pustaka dilakukan dengan mempelajari buku, artikel ilmiah, dan sumber lain yang mendukung penelitian ini, sedangkan metode konfirmatori merupakan jenis penelitian yang bertujuan untuk mengonfirmasi penelitian sebelumnya [3]. Konfirmatori dilakukan dengan membuktikan bahwa analisis keamanan yang ditulis dalam makalah milik Kang sesuai dengan hasil analisis formal menggunakan alat Scyther.

Beberapa langkah yang perlu dilakukan untuk mencapai tujuan yang diharapkan pada penelitian ini, yaitu:

a. Identifikasi protokol dan Scyther

Pada langkah identifikasi ini diawali dengan mengumpulkan buku, jurnal, makalah, dan sumber lainnya yang mendukung penelitian ini, khususnya makalah milik Kang mengenai *Lightweight Mutual Authentication* RFID *Protocol* dan makalah yang membahas mengenai Scyther. Sumber-sumber tersebut dipelajari atau diidentifikasi untuk mendasari hal-hal yang dilakukan pada langkah selanjutnya.

b. Menentukan deklarasi global

Setelah mempelajari protokol dan Scyther, selanjutnya adalah menentukan deklarasi global apa saja yang akan digunakan untuk mendukung pemodelan pada Scyther. Deklarasi ini bertujuan untuk mengidentifikasi nilai-nilai (contoh: *S-Box*) yang ada dalam protokol namun tidak didukung oleh Scyther. Hasilnya berupa Scyther yang akan membaca nilai yang dieksekusi sesuai dengan yang dideklarasikan.

c. Memodelkan protokol dengan Scyther

Langkah ini memiliki input berupa pemodelan skema *Lightweight Mutual Authentication* RFID *Protocol* pada Scyther. Pemodelan ini dituliskan sesuai dengan pesan/nilai yang dikirimkan dan aspek-aspek yang dibutuhkan dalam pengiriman protokol. Hasilnya berupa verifikasi protokol atau karakterisasi pihak yang didapat dari fitur Scyther.

d. Mengidentifikasi klaim keamanan

Tahap ini dilakukan identifikasi klaim keamanan terkait tingkat autentikasi yang didukung oleh Scyther, yang selanjutnya dapat digunakan untuk memverifikasi protokol sesuai dengan tujuannya. Setelah itu menulis bentuk pemodelan, sehingga ketika dieksekusi akan menampilkan hasil verifikasi protokol sesuai klaim keamanan yang dituliskan.

e. Verifikasi

Setelah selesai memodelkan protokol dan menulis klaim keamanannya, selanjutnya adalah memverifikasi protokol menggunakan fitur *Verify Protocol* dan *Characterization Roles* untuk mendukung analisis protokolnya.

f. Analisis

Berdasarkan hasil verifikasi dari protokol autentikasi *Single-tag* dan *Double-tag*, langkah selanjutnya adalah menganalisis. Analisis dapat dilakukan dengan melihat jendela hasil verifikasi. Pada jendela tersebut akan terlihat peran atau klaim yang aman atau klaim yang masih rentan terhadap serangan tertentu.

g. Penarikan kesimpulan

Pada tahap ini dilakukan penarikan kesimpulan terhadap analisis yang dilakukan, untuk menentukan level autentikasi yang dipenuhi oleh protokol Kang.

4. HASIL DAN PEMBAHASAN

Pada bab ini dibahas mengenai deskripsi protokol Kang dan analisis dari hasil implementasi pada Scyther. Analisis diambil dari hasil dari fitur *Verify Protocol* dan *Characterize Roles*. Tujuan menggunakan fitur *Verify Protocol* adalah untuk melihat hasil klaim keamanan pada tingkat autentikasi yang telah dituliskan dan tujuan menggunakan *Characterize Roles* adalah untuk memastikan bahwa pihak yang dieksekusi menjalankan protokol sesuai yang dimodelkan.

4.1. Hasil Verifikasi Protokol Autentikasi *Single-tag*

Pemodelan Protokol Autentikasi *Single-tag* diawali dengan mendeklarasikan secara global terkait nilai-nilai yang akan digunakan saat pertukaran. Adapun nilai S-box yang tidak didukung oleh Scyther akan dimisalkan sebagai fungsi hash. Fungsi matematis yang dicantumkan dalam desain akan diabaikan karena Scyther tidak mendukung adanya perhitungan matematis pada proses verifikasinya. Sehingga, pemodelan ini menggunakan enkripsi simetrik dengan asumsi bahwa pembagian kunci

rahasia yang dibagikan, digunakan untuk proses enkripsi dan dekripsi menggunakan satu kunci yang sama. Berikut disajikan hasil Verify Protocol pada Tabel 1.

Untuk membuktikan kebenaran protokol yang dimodelkan serta kebenaran hasil verifikasi, maka hasil verifikasi dapat dilihat pada fitur *Characterize Roles.* Fitur ini menampilkan hasil yang berfokus pada pihak-pihak yang dimodelkan pada protokol. Status hasil "*Reachable*" pada pihak I dan R, menandakan bahwa kedua pihak tersebut memiliki pemodelan protokol yang dapat dijalankan oleh Scyther dengan benar. Pola yang terdapat pada kolom komentar merupakan gambar komunikasi antar pihak yang dihasilkan dari pemodelan, atau bisa juga membentuk gambar pola serangan yang mungkin terjadi pada protokol. Berikut akan ditampilkan hasil verifikasi *Characterize Roles* pada Tabel 2.

Tabel 1 Hasil *Verify Protocol Single-tag*

Proto-kol	Role	Klaim	Hasil	Komentar
single-tag	I	*Alive*	*Ok*	Tidak ada serangan
		Weakagree	*Ok*	Tidak ada serangan
		Niagree	*Ok*	Tidak ada serangan
		Nisynch	*Ok*	Tidak ada serangan
	R	*Alive*	*Ok*	Tidak ada serangan
		Weakagree	*Ok*	Tidak ada serangan
		Niagree	*Ok*	Tidak ada serangan
		Nisynch	*Ok*	Tidak ada serangan

Tabel 2 Hasil *Characterize Roles Single-tag*

Proto-kol	Role	Klaim	Hasil	Komen-tar
single-tag	I	*single-tag, I1*	*Reachable*	Ter-dapat 1 pola
	R	*singletag, R1*	*Reachable*	Ter-dapat 1 pola

Hasil pemodelan protokol autentikasi *Single-tag* pada fitur *Verify Protocol* memiliki hasil yang sama antara pihak I dan pihak R. Kedua pihak tersebut menguji klaim autentikasi *aliveness, weakagree (weak-agreement), niagree (non-injective agreement),* dan *nisynch (non-injective synchronization).* Status dan komentar hasil verifikasi menunjukkan tidak ada serangan yang dapat mengganggu komunikasi pihak I dan pihak R pada tingkat autentikasi tersebut.

4.2. Hasil Verifikasi Protokol Autentikasi *Double-tag*

Pada dasarnya, pemodelan protokol autentikasi *Double-tag* sama seperti *Single-tag*. Perbedaan terletak pada jumlah *tag* yang di autentikasi dan jumlah kunci sesi yang dihasilkan. Hasil *Verify Protocol* disajikan pada Tabel 3.

Tabel 3 Hasil *Verify Protocol Double-tag*

Protokol	Role	Klaim	Hasil	Komentar
double-tag	IA	*Alive*	*Ok*	Tidak ada serangan dalam batasan
		Weakagree	*Ok*	Tidak ada serangan dalam batasan
		Niagree	*Ok*	Tidak ada serangan dalam batasan
		Nisynch	*Ok*	Tidak ada serangan dalam batasan
	R	*Alive*	*Ok*	Tidak ada serangan dalam batasan
		Weakagree	*Ok*	Tidak ada serangan dalam batasan
		Niagree	*Ok*	Tidak ada serangan dalam batasan
		Nisynch	*Ok*	Tidak ada serangan dalam batasan
	IB	*Alive*	*Ok*	Tidak ada serangan dalam batasan
		Weakagree	*Ok*	Tidak ada serangan dalam batasan
		Niagree	*Ok*	Tidak ada serangan dalam batasan
		Nisynch	*Ok*	Tidak ada serangan dalam batasan

Fitur lainnya yang digunakan untuk verifikasi adalah fitur *Characterize Roles*. Fitur ini digunakan untuk membuktikan kebenaran protokol yang dimodelkan. Tabel 3 menunjukkan bahwa hasil verifikasi fitur ini terhadap protokol autentikasi *Double-tag* memiliki status yang *Ok/Reachable*. Status tersebut berlaku untuk ketiga pihak yang dimodelkan yaitu *role* IA, *role* R, dan *role* IB. Artinya, pemodelan protokol autentikasi *Double-tag* dapat dijalankan oleh Scyther dengan benar. Keterangan yang ada dalam kolom komentar merupakan hasil pola yang terbentuk dari protokol yang dimodelkan. Hasil *Characterize Roles* pada protokol autentikasi *Double-tag*, akan ditampilkan pada Tabel 4.

Tabel 4 Hasil *Characterize Roles Double-tag*

Proto-kol	Role	Klaim	Hasil	Komentar
double-tag	IA	*double-tag*, IA1	*Reach-able*	Minimal ada 1 pola
	R	*double-tag*, R1	*Reach-able*	Minimal ada 1 pola
	IB	*double-tag*, IB1	*Reach-able*	Minimal ada 1 pola

Hasil pemodelan protokol autentikasi *Double-tag* pada fitur *Verify Protocol* memiliki hasil yang sama antara pihak IA, R, dan IB. Ketiga pihak tersebut menguji klaim autentikasi yang diujikan pada protokol autentikasi *Single-tag*. Status dan komentar hasil verifikasinya menunjukkan tidak ada serangan dalam batasan yang dapat mengganggu komunikasi antara ketiga pihak pada tingkat autentikasi tersebut.

4.3. Analisis Protokol Autentikasi *Single-tag* dan *Double-tag*

Dari hasil verifikasi yang didapat pada Subbab IV.3.1 dan IV.3.2, terlihat bahwa protokol *Single-tag* dan *Double-tag* memiliki pola yang hampir sama pada hasil *Characterize Roles*. Pola tersebut membentuk skema yang menunjukkan mekanisme protokol yang dimodelkan menjalankan proses pengiriman dan penerimaan antara satu pihak dengan pihak lainnya. Gambar yang ditampilkan merupakan hasil pola pada salah satu pihak.

Menurut Kang [8], protokol autentikasi *Single-tag* dan *Double-tag* aman terhadap serangan *replay*, *relay*, dan *eavesdropping*. Aspek keamanan lain yang mendukung protokol ini adalah adanya anonimitas *tag* dan *forward security*. Kedua aspek keamanan ini pada protokol *Single-tag* dan *Double-tag* dibuktikan melalui klaim autentikasi. Hasil verifikasi pada Scyther menunjukkan bahwa baik protokol autentikasi *Single-tag* maupun *Double-tag* aman terhadap berbagai serangan autentikasi. Adapun hasil teoritis yang telah dinyatakan oleh Kang memiliki hubungan dengan klaim autentikasi pada Scyther. Hubungan tersebut dapat dilihat pada Tabel 5

Tabel 5 Hubungan antara hasil Scyther dengan klaim keamanan oleh Kang

No	**Keamanan**	***Single-tag***	***Double-tag***	**Scyther**
1	*Replay Attack*	Aman	Aman	*Non-injective synchroi-zation*
2	*Relay Attack*	Aman	Aman	*Non-injective synchroi-zation*
3	*Tag Anonymity*	Aman	Aman	*Aliveness,Weak-agree-ment*
4	*Eaves-dropping*	Aman	Aman	*Aliveness*
5	*Forward Security*	Memenuhi	Memenuhi	*Non-injective agree-ment*

Berdasarkan Tabel 5 pemenuhan anonimitas *tag* terjadi karena adanya *S-Box* yang digunakan untuk mengamankan nilai acak milik *tag* dan *reader*. Akibatnya, hanya pihak asli yang dapat mengetahui isi nilai acak yang dikirim, sehingga membuat pihak tersebut yakin bahwa komunikasi dilakukan dengan pihak yang benar. Pada pemodelan protokol autentikasi *Single-tag* dan *Double-tag*, pemenuhan anonimitas *tag* digunakan untuk mencapai klaim autentikasi *aliveness*. Penggunaan fungsi *hash* sebagai pengganti $S-box$ digunakan untuk mengamankan nilai acak milik *tag* dan *reader*, akibatnya *tag* maupun *reader* dapat yakin bahwa mereka berkomunikasi dengan pihak yang benar jika menggunakan nilai *hash* yang mereka ketahui. Adanya anonimitas *tag* membuat protokol terhindar dari *eavesdropping* karena jika ada musuh yang mencuri informasi pada proses komunikasi, maka musuh tersebut tidak akan mendapat informasi apapun kecuali dapat mengetahui nilai $S-box$ nya.

Klaim keamanan autentikasi *weak-agreement* berkaitan dengan persetujuan komunikasi. Pada saat *tag* memulai komunikasi dengan mengirim pesan permintaan nilai acak kepada *reader*, *reader* merespon dengan mengirim nilai acak miliknya. Akibatnya, *tag* dapat melanjutkan komunikasi dengan *reader* karena *reader* telah menyetujui permintaannya. Berdasarkan klaim *aliveness* yang terpenuhi, klaim *weak-agreement* juga terpenuhi karena adanya anonimitas *tag*. Persetujuan dapat terjadi apabila nilai yang dikirimkan merupakan nilai acak yang hanya diketahui antara *tag* dan *reader*.

Non-injective agreement dipenuhi dengan cara menentukan suatu nilai yang disepakati dalam melakukan komunikasi. Pemodelan protokol autentikasi *Single-tag* dan *Double-tag* pada Scyther memanfaatkan kunci simetrik $k(I,R)$ dan

$k(IA, R, IB)$ sebagai nilai kesepakatan yang dipertukarkan pada saat komunikasi berlangsung. Penggunaan kunci simetrik sekaligus memenuhi *forward security*. *Forward security* terpenuhi apabila musuh mendapatkan kunci simetrik, maka musuh tidak akan berpengaruh pada data lain karena telah diamankan melalui anonimitas *tag*.

Pembuktian terakhir klaim autentikasi pada protokol *Single-tag* dan *Double-tag* adalah *non-injective synchronization*. Klaim tersebut berhubungan dengan sinkronisasi dalam protokol. Sinkronisasi yang dimaksud adalah dimana semua nilai yang didespkripsikan pada protokol benar-benar nilai yang digunakan dalam komunikasi. Pada pemodelan protokol autentikasi *Single-tag* dan *Double-tag* mendeskripsikan nilai *fresh hi*, kemudian nilai tersebut digunakan sebagai *nonce* yang dikirimkan dalam komunikasi dan diterima oleh *reader* sebagai *variable hi.* Nilai-nilai lain pada pemodelan yang telah dideskripsikan sebagai *fresh/variable* merupakan nilai yang terjadi dalam proses pengiriman dan penerimaan nilai, sehingga klaim autentikasi *non-injective synchronization* terpenuhi. Klaim ini berhubungan dengan keamanan terhadap serangan *replay* dan *relay*. Kedua serangan tersebut termasuk serangan aktif, dimana musuh dapat memasuki komunikasi antara *tag* dan *reader* untuk mengubah atau menambahkan nilai yang akan dikirimkan. Apabila nilai yang diterima tidak sesuai deskripsi, maka dapat diindikasikan bahwa dalam protokol terjadi serangan dalam proses komunikasi sehingga menyebabkan klaim autentikasi gagal terpenuhi.

5. KESIMPULAN

Berdasarkan hasil penelitian yang dilakukan, dapat diambil kesimpulan bahwa hasil analisis protokol *Single-tag* dan *Double-tag* adalah sebagai berikut:

a. Protokol autentikasi *Single-tag* dan *Double-tag* memenuhi klaim autentikasi *aliveness*, *weak-agreement*, *non-injective agreement*, dan *non-injective synchronization* pada Scyther.
b. Masing-masing klaim autentikasi yang dipenuhi memiliki hubungan dengan analisis secara teoritis yang dinyatakan oleh Kang. Hubungan analisis teoritis dengan analisis empiris adalah sebagai berikut:
    - Klaim *aliveness* terpenuhi karena adanya penggunaan *S-Box* yang dimodelkan sebagai fungsi *hash* dalam deklarasi Scyther sehingga mendukung pemenuhan anonimitas *tag* dan keamanan terhadap serangan *eavesdropping*.
    - Klaim *weak agreement* terpenuhi karena adanya persetujuan komunikasi antara *tag* dan *reader* dalam bentuk proses kirim-terima sehingga memenuhi anonimitas *tag*.
    - Klaim *non-injective agreement* terpenuhi karena adanya kunci simetris bersama yang dibagikan sebagai nilai kesepakatan antara *tag* dan *reader* sehingga memenuhi *forward security*.
    - Klaim *non-injective synchronization* terpenuhi karena setiap nilai yang dideklarasikan sesuai dengan nilai yang dipertukarkan dalam komunikasi dan adanya pencatatan waktu pada proses *challenge-response*, sehingga dapat menghindari serangan *replay* dan *relay*.

sasinya. Oleh karena itu, disarankan pada penelitian selanjutnya untuk menggunakan alat analisis lain yang menunjang klaim autentikasi secara lengkap.

REFERENSI

[1] Q. Xiao, T. Gibbons, and H. Lebru, "RFID Technology, Security Vulnerabilities, and Countermeasures," *Supply Chain W. to Flat Organ.*, no. January 2009, 2009, doi: 10.5772/6668.

[2] M. Habibi, M. Gardeshi, and M. R. Alaghband, "Practical Attacks on a RFID Authentication Protocol Conforming to EPC C-1 G-2 Standard," *Int. J. UbiComp*, vol. 2, no. 1, pp. 1–13, 2011, doi: 10.5121/iju.2011.2101.

[3] Z. Syafrilah, *Analisis Formal Protokol Autentikasi Gope Menggunakan Scyther Analisis Formal Protokol Autentikasi Gope Menggunakan Scyther*. Politeknik Siber dan Sandi Negara, 2020.

[4] A. Juels, "'Yoking-proofs' for RFID tags," *Proc. - Second IEEE Annu. Conf. Pervasive Comput. Commun. Work. PerCom*, pp. 138–143, 2004, doi: 10.1109/PERCOMW.2004.1276920.

[5] S. Piramuthu, "Protocols for RFID tag/reader authentication," *Decis. Support Syst.*, vol. 43, no. 3, pp. 897–914, 2007, doi: 10.1016/j.dss.2007.01.003.

[6] H. Y. Chien and C. H. Chen, "Mutual authentication protocol for RFID conforming to EPC Class 1 Generation 2 standards," *Comput. Stand. Interfaces*, vol. 29, no. 2, pp. 254–259, 2007, doi: 10.1016/j.csi.2006.04.004.

[7] P. Peris-Lopez, J. C. Hernandez-Castro, J. M. Estevez-Tapiador, and A. Ribagorda, "Cryptanalysis of a novel authentication protocol conforming to EPC-C1G2 standard," *Comput. Stand. Interfaces*, vol. 31, no. 2, pp. 372–380, 2009, doi: 10.1016/j.csi.2008.05.012.

[8] J. Kang, "Lightweight mutual authentication RFID protocol for secure multi-tag simultaneous authentication in ubiquitous environments," *J. Supercomput.*, vol. 75, no. 8, pp. 4529–4542, 2019, doi: 10.1007/s11227-016-1788-6.

[9] R. Air, I. Protocol, and M. Version, "EPC ™

Radio-Frequency Identity Protocols Generation-2 UHF RFID Specification for RFID Air Interface," pp. 1–152, 2015.

[10] R. Patel, B. Borisaniya, A. Patel, D. Patel, M. Rajarajan, and A. Zisman, "Comparative analysis of formal model checking tools for security protocol verification," *Commun. Comput. Inf. Sci.*, vol. 89 CCIS, pp. 152–163, 2010, doi: 10.1007/978-3-642-14478-3_16.

[11] E. S. Han and A. goleman, daniel; boyatzis, Richard; Mckee, *Operational Semantics and Verification of Security Protocols*, vol. 53, no. 9. 2019.

[12] G. Lowe, "A Hierarchy of Authentication Specifications," *IEEE*, pp. 31–33, 1997.

[13] C. Cremers, *Scyther User Manual*. 2014.