Jurnal Teknologi Informasi dan Ilmu Komputer (JTIIK) DOI: 10.25126/jtiik.201961303
Vol. 6, No. 1, Februari 2019, hlm. 107-112 p-ISSN: 2355-7699
Akreditasi KEMENRISTEKDIKTI, No. 30/E/KPT/2018 e-ISSN: 2528-6579

ANALISIS PROTOKOL *CryptO-0N2* DENGAN MENGGUNAKAN *SCYTHER TOOL*

Esti Rahmawati Agustina[1], Magdalena Christine[2], Irma Fitriani[3]

[1,2,3]Badan Siber dan Sandi Negara
Email: [1]esti.rahmawati@bssn.go.id, [2]magdalena.christine@bssn.go.id, [3]irma.fitriani@bssn.go.id

(Naskah masuk: 28 November 2018, diterima untuk diterbitkan: 07 Januari 2019)

Abstrak

Pemilihan Umum (pemilu) di Indonesia merupakan peristiwa yang sangat penting mengingat Indonesia merupakan negara yang menganut paham demokrasi. Metode yang digunakan dalam pemilihan umum di Indonesia adalah dengan menggunakan kertas suara yang ditandai yaitu dicentang atau dicoblos. Banyak kelemahan dan kecurangan yang terjadi dalam sistem konvensional ini. Misalnya pemilih ganda, data pemilih tidak valid, surat suara rusak dan lain sebagainya. Salah satu solusi untuk menyelesaikan permasalahan pada sistem pemilu konvensional adalah dengan menerapkan *electronic voting* (*e-voting*). Berbagai penelitian dan pengembangan dilakukan dalam rangka membangun sistem *e-voting* yang aman. Salah satunya adalah dengan mengimplementasikan teknik kriptografi. Salah satu protokol *e-voting* yang menerapkan teknik kriptografi adalah protokol *CryptO-0N2*. Pada perkembangannya, protokol ini telah dianalisis dengan menggunakan verifikasi formal berbasis pendekatan logika yaitu *BAN Logic*. Verifikasi formal terhadap suatu protokol dapat dijamin obyektivitasnya dengan menggunakan *tools* tertentu. Pada paper ini disajikan analisis protokol *CryptO-0N2* dengan menggunakan *Scyther Tool*. *Tool* ini memeriksa klaim *secrecy* dan *authentication* dari protokol *CryptO-0N2*. Hasil menunjukkan dari 17 klaim (*secrecy* dan *authentication*) terdapat 10 klaim sukses dan 7 klaim gagal.

Kata kunci: *kriptografi, protokol CryptO-0N2, Scyther Tool*

CryptO-0N2 PROTOCOL ANALYSIS USING SCYTHER TOOL

Abstract

Elections in Indonesia is a very important event considering Indonesia is a democratic country. The method of the general election in Indonesia is use a marked ballot that is ticked or punched. Many weaknesses and frauds occur in this conventional system. For example multiple voters, invalid voter data, broken ballots and so forth. One solution to solve the problems in conventional electoral systems is to apply electronic voting (e-voting). Various research and development carried out in order to build a secure e-voting system. One of them is by implementing cryptographic techniques. One of the e-voting protocols employing cryptographic techniques is the CryptO-0N2 protocol. In its development, this protocol has been analyzed by using formal logic-approach based on logical verification that is BAN Logic. Formal verification of a protocol can be guaranteed objectivity by using certain tools. In this paper we present CryptO-0N2 protocol analysis using Scyther Tool. This tool examines the secrecy and authentication claims of the CryptO-0N2 protocol. The result shows from 17 claims (secrecy and authentication) there are 10 successful claims and 7 claims are failed.

Keywords: *cryptography, CryptO-0N2 protocol, Scyther Tool*

1. PENDAHULUAN

Pemilihan Umum (Pemilu) di Indonesia merupakan peristiwa yang sangat penting. Pemilu telah menjadi pesta rakyat. Hal ini dikarenakan Indonesia merupakan salah satu negara di dunia yang menganut paham demokrasi. Indonesia telah menyelenggarakan sebelas kali pemilihan umum yang berskala nasional yaitu pada tahun 1955, 1971, 1982, 1987, 1992, 1997, 1999, 2004, 2009, dan 2014 (www.kompas.com). Metode yang digunakan dalam pemilihan umum tersebut adalah dengan menggunakan kertas suara yang ditandai yaitu dicentang ataupun dicoblos. Berdasarkan artikel dari www.liputan6.com, metode pemilihan umum secara konvensional banyak memiliki kelemahan dan menimbulkan kecurangan. Misalnya pemilih ganda, data pemilih yang tidak valid, surat suara rusak dan lain sebagainya.

Seiring dengan pemanfaatan Teknologi Informasi dan Komunikasi (TIK) dalam kehidupan sehari-hari, banyak penelitian dilakukan untuk

mencari alternatif solusi dalam menjawab tantangan terkait potensi kecurangan pada sistem pemilihan umum konvensional. Salah satu alternatif solusi tersebut adalah dengan *electronic voting (e-voting*). Jika diterapkan dengan tepat, sistem e-voting dapat mengurangi penipuan data, mempercepat pengolahan hasil pemilihan, dan meningkatkan aksesibilitas dalam pemilihan umum (Wolf, 2011). Pertengahan tahun 2009 di Kabupaten Jembrana (Bali), sistem ini pertama kali diterapkan. Sejak saat itu telah berpuluh kali pemilihan kepala dusun dilakukan dengan *e-voting*. Tercatat Kabupaten Jembrana telah menghemat anggaran sebesar 60%.

Pada tanggal 30 Maret 2010, Mahkamah Konstitusi telah mengesahkan bahwa penerapan *e-voting* tidak melanggar konstitusi selama tidak melanggar asas Pemilu yaitu luber dan jurdil. Dengan demikian *e-voting* mungkin diterapkan pada skala yang lebih luas, misalnya Pemilihan Umum Kepala Daerah dan Wakil Kepala Daerah. Dengan pengesahan tersebut berbagai daerah telah menyambut positif dan menerapkan *e-voting* ini, misalnya Pemilihan Kades di Musi Rawas pada tahun 2014 (www.antaranews.com), Pemilihan Kepala Desa Benteng di Kapubaten Banyuasin pada tahun 2015 (www.tribunnews.com), Pemilihan Ketua RW 2 di Kelurahan Pulisen, Kecamatan Boyolali pada tahun 2017 (www.jatengpos.com), dan lain sebagainya.

Teknik kriptografi dapat diimplementasikan untuk membangun sistem e-voting yang aman (Scheiner, 1996). Menurut Menezes, kriptografi adalah ilmu yang mempelajari teknik-teknik matematika yang berhubungan dengan aspek-aspek keamanan informasi seperti kerahasiaan data, keabsahan data, integitas data, serta *authentication* data (Menezes, 1997). Pada tahun 2012, Prakasa memperkenalkan salah satu protokol *e-voting* yang menggunakan fitur-fitur kriptografi dengan nama protokol CryptO-0N2 (Prakasa, 2012). Protokol ini merupakan protokol *secure ubiquitus e-voting* yang menggunakan fitur-fitur kriptografi yaitu fungsi hash dan asimetrik (penggunaan kunci privat dan kunci publik) untuk menjamin keamanan transmisi data. Protokol ini mempunyai sifat ubiquitus. Sifat ini mendukung proses pemilihan dapat dilakukan dimana saja, kapan saja, dan bagaimana saja. Untuk mendukung karakteristik ini digunakan pengidentifikasi biometrik yang melekat pada setiap orang sebagai identitas yang unik.

Sebagai jaminan keamanan pada protokol CryptO-0N2, analisis telah dilakukan terhadap protokol ini. Analisis yang dilakukan adalah dengan menggunakan pendekatan verifikasi formal yaitu BAN *Logic Approach* (Prakasa, 2012). Analisis dilakukan untuk membuktikan bahwa pihak – pihak yang berkomunikasi saling percaya dengan apa yang dipertukarkan untuk menjamin keamanan dalam komunikasi tersebut. Analisis dengan menggunakan BAN Logic ini mempunyai kelemahan terkait dengan subyektivitas verifikator karena bergantung pada kemahiran verifikator untuk memodelkan logika, Pada tahun 2006, Cas Cremers melakukan publikasi terhadap penelitiannya terkait dengan suatu metodologi untuk analisis dan verifikasi formal terhadap suatu *security protocol* dalam suatu bentuk *tool* analisis yang dikenal dengan nama *Scyther Tool* (Cremers, 2006). *Tool* analisis ini dapat menganalisis jaminan *secrecy* dan *authentication* pada suatu protokol.

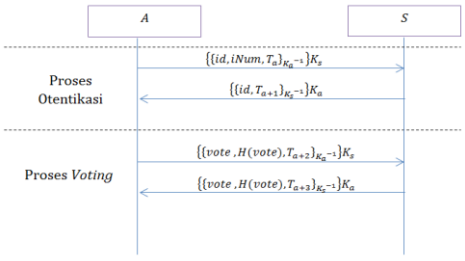
Verifikasi formal terhadap suatu protokol dapat dijamin obyektivitasnya dengan munculnya *Scyther Tool*. Pada makalah ini akan disajikan analisis terhadap protokol CryptO-0N2 dengan *Scyther Tool* serta analisis terhadap hasil verifikasinya terkait dengan jaminan *secrecy* dan *authentication* pada protokol tersebut.

2. LANDASAN TEORI

2.1 Protokol CryptO-0N2

Protokol CryptO-0N2 merupakan suatu protokol *secure ubiquitus e-voting*, yaitu protokol yang didesain untuk penggunaan *biometric identifier* sehingga dapat mendukung sistem yang bersifat *ubiquitus* serta fitur-fitur kriptografi untuk menjamin keamanan informasi selama *e-voting* berlangsung. (Prakasa, 2012). *Ubiquitus* merupakan suatu sistem yang memungkinkan manusia berinteraksi dengan komputer secara kontinyu, dimana saja, kapan saja dan bagaimana saja.

Protokol "CryptO-0N2" terdiri dari 2 (dua) proses, yaitu proses otentikasi dan voting. Proses otentikasi merupakan proses verifikasi untuk memeriksa apakah pemilih merupakan pemilih yang sah, sedangkan proses *voting* adalah proses merekapitulasi suara pilihan pemilih. Skema protokol ini adalah *client – server* sebagaimana yang terlihat pada Gambar 1 dibawah ini.



Gambar 1. Skema Protokol CryptO-0N2

Berikut adalah tabel yang menjelaskan properti yang digunakan dalam protokol CryptO-0N2.

Deskripsi protokol CryptO-0N2 adalah sebagai berikut:

a. *Client* dalam hal ini TPS akan mengirimkan pesan ke server berupa gabungan identitas pemilih, *identification number*, dan *timestamp*. Ketiga properti ini ditandatangani dengan menggunakan kunci privat TPS dan dienkripsi menggunakan kunci publik server.

Tabel 1. Properti Protokol CryptO-0N2

simbol properti	arti
id	Identitas pemilih
$iNum$	*Identification number*
$T_a, T_{a+1}, T_{a+2}, T_{a+3}$	*Timestamps*
K_a	Kunci publik Tempat Pemungutan Suara (TPS)
K_s	Kunci Publik Server Otentikasi
$K_a{}^{-1}$	Kunci privat Tempat Pemungutan Suara (TPS)
$K_s{}^{-1}$	Kunci Privat Server Otentikasi
$vote$	Pilihan dari pemilih
$H(vote)$	Nilai *hash* dari pilihan dari pemilih

b. Setelah server menerima pesan pertama, server ini akan mendekripsi pesan tersebut dengan menggunakan kunci privatnya. Kemudian hasil dekripsi tersebut akan didekripsi lagi mengggunakan kunci publik TPS. Dalam proses ini akan diverifikasi apakah pesan pertama ini benar-benar datang dari TPS. Selanjutnya ketiga properti didapatkan. Verifikasi juga dilakukan oleh server dengan cara memeriksa apakah identitas pemilih terdapat dalam data base pemilih, jika iya maka akan dilakukan pembangkitan *identification number* dengan inputan identitas pemilih tersebut. Nilai yang dibangkitkan akan dibandingkan dengan nilai *identification number* yang dikirimkan oleh TPS. Jika kedua nilai tersebut sama maka menyatakan pemilih merupakan pemilih yang sah. Jika demikian maka server otentikasi akan mengirimkan pesan ke TPS yang mengandung identitas pemilih dan *timestamp* plus 1. Kedua properti ini akan ditandatangani dengan menggunakan kunci privat server otentikasi dan dienkripsi dengan menggunakan kunci publik TPS.

c. Setelah TPS menerima pesan kedua, maka TPS akan mendekripsi pesan tersebut dengan menggunakan kunci privatnya, selanjutnya hasil dekripsi tersebut akan didekripsi lagi dengan menggunakan kunci publik server. Selanjutnya didapatkan identitas pemilih dan *timestamp* plus 1. Kedua properti ini sebagai tanda bahwa pemilih dapat melakukan proses pemilihan. Selanjutnya pemilih akan melakukan proses pilihan. Kemudian, TPS akan mengirimkan pesan ketiga ke server yang berisi pilihan pemilih, nilai *hash* dari pilihan pemilih dan timestamp plus 2. Ketiga properti ini akan ditandatangani dengan menggunakan kunci privat TPS dan kemudian dienkripsi dengan menggunakan kunci publik server.

d. Setelah server menerima pesan ketiga, maka server akan mendekripsi pesan tersebut dengan menggunakan kunci privatnya, selanjutnya hasil dekripsi tersebut akan didekripsi lagi dengan menggunakan kunci publik TPS. Selanjutnya ketiga properti ini didapatkan. Server akan melakukan verifikasi terhadap nilai hash dari pilihan yang dikirimkan. Server akan melakukan penghitungan nilai *hash* dari pilihan yang dikirimkan oleh TPS. Jika nilai tersebut sama maka dapat diyakini bahwa tidak ada perubahan pilihan selama komunikasi berlangsung. Kemudian server memasukkan piilihan tersebut dalam *database* tabulasi suara. Sebagai notifikasi bahwa hasil pemilihan telah masuk ke dalam data base tabulasi suara maka server akan mengirimkan pesan kepada TPS yang berisi pilihan, nilai hash dari pilihan dan timestamp plus 3. Pesan ini kemudian ditandatangani dengan menggunakan kunci privat server dan kemudian dienkripsi menggunakan kunci publik TPS.

e. Setelah TPS menerima pesan keempat maka TPS akan melakukan dekripsi dengan menggunakan kunci privatnya. Kemudian hasil dekripsi tersebut akan didekripsi lagi menggunakan kunci publik TPS. Selanjutnya ketiga properti didapatkan, sebagai notifikasi dari server bahwa pilihan pemilih sudah masuk dalam *database* tabulasi suara.

2.2 Scyther Tool Analyzer

Scyther merupakan *tool* untuk melakukan analisis terhadap *security protocol* dengan *perfect cryptography assumption*, yaitu mengasumsikan bahwa keseluruhan fungsi kriptografi adalah *perfect* dalam artian pihak yang tidak berwenang tidak dapat mempelajari apapun dari sebuah *ciphertext* pesan kecuali pihak tersebut mengetahui kunci untuk mendekripsi pesan tersebut (Cremers, 2006). *Tool* ini dapat digunakan untuk menemukan permasalahan yang mungkin muncul dalam rangka mendesain suatu protokol kriptografi, salah satunya dengan mengetahui berbagai serangan yang mungkin muncul. *Tool* ini juga dilengkapi dengan *Graphical User Interface* (GUI). Input dari *Scyther Tool* adalah deskripsi dari suatu *security protocol* yang meliputi propertinya sehingga dapat diperoleh klaim dari penggunaan properti tersebut dan mengevaluasinya. Evaluasi dilakukan terhadap klaim *secrecy* dan *authentication.* Klaim *secrecy* untuk memastikan kerahasiaan dari suatu pesan, sedangkan klaim *authentication* memastikan keberadaan dari mitra komunikasi (*aliveness*), mitra komunikasi tersebut aktif dan mengeksekusi setiap langkah protokol serta pihak pengirim (*synchronisation*) dan penerima menyelesaikan seluruh langkah protokol juga pihak penerima dan pengirim setuju dengan pesan yang dipertukarkan (*agreement*).

3. METODE PENELITIAN

Penelitian ini dilakukan dengan langkah – langkah sebagai berikut:

a. Mengkodekan deskripsi protokol CryptO-0N2 sesuai dengan format yang ditentukan oleh *Scyther Tool.* Tahap ini bertujuan untuk menkonversi deskripsi protocol sesuai dengan Bahasa yang dimengerti oleh *Scyther Tool.*
b. Melakukan verifikasi terhadap kode pada poin a dengan menggunakan *Scyther Tool.* Tujuan dari langkah ini adalah untuk menjalankan verifikasi terhadap kode yang telah dibuat sesuai dengan deskripsi protocol.
c. Melakukan observasi hasil verifikasi protokol CryptO-0N2 terhadap klaim sukses dan gagal. Tujuan dari langkah ini adalah untuk memeriksa hasil verifikasi dari protokol. Suatu properti pada protokol di klaim sukses jika tidak ada serangan yang muncul terhadap klaim tertentu. Hal tersebut ditandai dengan status OK pada hasil *Scyther tool*.
d. Melakukan analisa terhadap dengan klaim yang gagal sesuai dengan grafik hasil verifikasi *Scyther Tool. Tool* ini akan menampilkan grafik serangan yang mugkin terjadi jika satu atau banyak klaim dinyatakan gagal.

4. PEMBAHASAN DAN ANALISIS

Analisis terhadap protokol CryptO-0N2 adalah dengan melakukan evaluasi terhadap klaim *secret* dan *authentication* pada protokol. Input dari *Scyther Tool* adalah deskripsi protokol CrptO-0N2disesuikan dengan aturan pengkodean protokol pada *tool* ini. Berikut adalah tabel yang mendeskripsikan protokol CryptO-0N2 yang telah disesuaikan dengan aturan pengkodean *Scyther Tool.*

Tabel 2. Deskripsi Protokol CryptO-0N2 yang Telah Dikodekan

jenis input	***syntax***	**deskripsi**
Deklarasi Global	usertype Timestamp;	Penggunaan *timestamo* dan fungsi *hash*
Role I (inisiator)	fresh T1: Timestamp; fresh A: Ticket; fresh B:Ticket; fresh V: Ticket; fresh T3: Timestamp;	Inisiator membangkitan T1,T3 A, B, dan V secara *fresh.* *A* adalah id; B adalah iNum; V adalah *vote;*
	macro m1=h(V);	*Macro* adalah penggunaan suatu formulasi, m1 adalah output fungsi *hash* dari V.
	var A: Ticket; var T2: Timestamp; var T4: Timestamp; var V: Ticket; var m4: Ticket;	Inisiator menerima pesan berisi A, V, m4, T4 dan T2. A adalah id; V adalah *vote;* m4 adalah nilai *hash* dari *vote* yang dihitung oleh Responder.
	send_1(I,R,{{A,B, T1} sk(I)}pk(R));	Inisiator mengirimkan pesan pertama kepada Responder
	send_3(I,R,{{V,m1,	Inisiator
	T3}sk(I)}pk(R));	mengirimkan pesan ketiga kepada Responder
	recv_2(R,I,{{A,T2} sk(R)}pk(I));	Inisiator menerima pesan kedua dari Reponder
	recv_4(R,I,{{V,m4,T4} sk(R)}pk(I));	Inisiator menerima pesan keempat dari Reponder

	claim_i1(I, *Secret*, A);
	claim_i2(I, *Secret*, B);
	claim_i3(I, *Secret*, T1);
	claim_i4(I, *Secret*, V);
	claim_i5(I, *Secret*, m1);
	claim_i6(I, *Secret*, T3);
		Klaim Inisiator terkait dengan *secrecy*

	claim_i7(I, Alive);
	claim_i8(I, Nisynch);
	claim_i9(I, Niagree);
		Klaim Inisiator terkait dengan otentikasi

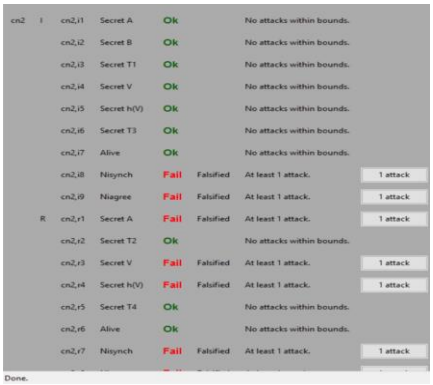
Role R (responder)	fresh T2: Timestamp; fresh T4: Timestamp;	Responder membangkitan T2 dan T4 secara *fresh.*
	macro m3=h(V);	*Macro* adalah penggunaan suatu formulasi, m3 adalah output fungsi *hash* dari V.
	var T1: Timestamp; var A: Ticket; var B: Ticket; var V: Ticket; var m2: Ticket; var T3: Timestamp;	Responder menerima pesan yang berisi A, B, V, m2, T3, dan T1 A adalah id; B adalah iNum; V adalah *vote;* m2 nilai hash dari vote.
	recv_1(I, R, {{A, B, T1}sk(I)}pk(R));	Responder menerima pesan pertama dari inisiator
	recv_3(I, R, {{V, m2, T3}sk(I)}pk(R));	Responder menerima pesan ketiga dari inisiator
	send_2(R,I,{{A,T2} sk(R)}pk(I));	Responder mengirimkan pesan kedua kepada inisiator
	send_4(R,I,{{V,m3,T4} sk(R)}pk(I));	Responder mengirimkan pesan keempat kepada inisiator

	claim_r1(R, *Secret*, A);
	claim_r2(R, *Secret*, T2);
	claim_r3(R, *Secret*, V);
	claim_r4(R, *Secret*, m3);
	claim_r5(R, *Secret*, T4);
		Klaim Responder terkait dengan *secrecy*

	claim_r6(R, Alive);
	claim_r7(R, Nisynch);
	claim_r8(R, Niagree);
		Klaim Responder terkait dengan otentikasi

*Kode sumber protokol CryptO-0N2 terdapat pada alamat berikut ini http://bit.ly/Scyther-CryptO-0N2.

Dari tabel 2 diatas dapat dilihat terdapat 17 klaim terkait dengan *secrecy* dan *authentication* untuk *Role* I maupun R. Langkah selanjutnya setelah mengkodekan protokol adalah melakukan verifikasi terhadap deskripsi protokol tersebut. Gambar 2 menunjukkan hasil verifikasi protokol CryptO-0N2. Dari 17 klaim, terdapat 10 klaim yang dinyatakan benar atau OK dan 7 klaim dinyatakan gagal atau *FAIL*.



Gambar 2. Hasil Verifikasi Protokol CryptO-0N

Berikut adalah penjelasannya:

a. Seluruh klaim *secret* dari *Role* I (*Inisiator*) adalah benar. Hal ini dapat diartikan bahwa parameter yang dikirimkan oleh *Inisiator* dijamin kerahasiaannya;
b. Tidak seluruh klaim *secret* dari *Role* R (*Responder)* adalah benar. Hal ini dapat diartikan bahwa tidak semua parameter yang dikirimkan oleh *Responder* dijamin kerahasiaannya. Parameter yang dinyatakan *secret* adalah T2 dan T4, sedangkan parameter yang gagal dinyatakan *secret* adalah A, V, dan h(V);
c. Seluruh klaim *authentication* tipe Aliveness adalah benar. Hal ini diartikan bahwa baik *Inisiator* juga *Responder* telah mengeksekusi semua *events;*
d. Seluruh klaim *authentication* tipe *Synchronisation* adalah tidak benar. Hal ini dapat diartikan bahwa terdapat setidaknya satu *attack* sehingga klaim tersebut tidak benar;
e. Seluruh klaim *authentication* tipe *Agreement* adalah tidak benar. Hal ini dapat diartikan bahwa terdapat setidaknya satu *attack* sehingga kaim tersebut tidak benar.

Setiap klaim yang dinyatakan *FAIL* dapat ditemukan informasi berupa skema serangan yang menyebabkan klaim tersebut gagal. Seluruh skema (gambar 3 sampai gambar 9) dari klaim – klaim tersebut terdapat pada http://bit.ly/attackonCryptO-0N2.

Berikut adalah penjelasan dari setiap klaim yang dinyatakan gagal:

a. Pada jenis klaim *authentication* tipe *synchronisation* oleh *Role* I (*Inisiator*)
   Klaim *authentication* dengan tipe *synchronisation* pada protokol Crypt0-ON2 tidak terpenuhi atau ditemukan kondisi yang tidak memenuhi untuk klaim tersebut. Klaim *authentication* tipe *synchronisation* merupakan klaim *authentication* yang paling kompleks dari pemodelan *authentication* dari verifikasi *Scyther*. Klaim ini mensyaratkan setiap *run* dari protokol yang diinisiasi oleh agen I harus terkorenspondensi satu-satu dengan pihak R. Pada Gambar 3 dapat dilihat bahwa terdapat *run* yang tidak unik yaitu recv_1 yang dapat dilakukan melalui *run*#2 dan *run* #3, sehingga terdapat dua *run* atas satu sesi protokol. Hal ini menyebabkan kondisi klaim *authentication* tipe *synchronisation* tidak terpenuhi.
b. Pada jenis *claim authentication* tipe *agreement* oleh *Role* I (*Inisiator*)
   Klaim *authentication* dengan tipe *agreement* pada protokol Crypt0-ON2 tidak terpenuhi atau ditemukan kondisi yang tidak memenuhi untuk klaim tersebut. Klaim *authentication* tipe *synchronisation* merupakan klaim *authentication* yang juga relatif kompleks dari pemodelan *authentication* dari verifikasi *Scyther*. Klaim ini mensyaratkan setiap *run* dari protokol yang diinisiasi oleh agen I harus sepakat dengan data yang dipertukarkan dengan pihak R. Pada Gambar 4 dapat dilihat bahwa terdapat *run* yang tidak unik yaitu send_1 yang dapat dilakukan melalui *run*#2 dan *run* #3, sehingga terdapat dua *run* atas satu sesi protokol. Hal ini menyebabkan kondisi klaim *authentication* tipe *agreement* tidak terpenuhi.
c. Pada jenis klaim *secret* dengan parameter A oleh *Role* R (*responder*)
   Klaim *secret* pada parameter A dapat digagalkan dengan skenario yaitu Eve melakukan impersonasi Bob, sehingga pesan send_1 yang harusnya dienkripsi dengan kunci publik Bob, tetapi dienkripsi dengan kunci publik Eve, sehingga Eve mampu melakukan dekripsi atas pesan yang terkandung di dalam send_1. Dengan demikian klaim *secret* pada parameter A menjadi gagal. Alur skenario serangan tersebut dapat dilihat pada Gambar 5.
d. Pada jenis klaim *secret* dengan parameter V oleh *Role* R (*responder*)
   Klaim *secret* pada parameter V dapat digagalkan dengan skenario yaitu Eve melakukan impersonasi Bob, sehingga pesan send_1 yang harusnya dienkripsi dengan kunci publik Bob, tetapi dienkripsi dengan kunci publik Eve, sehingga Eve mampu melakukan dekripsi atas pesan yang terkandung di dalam send_1. Dengan demikian klaim *secret* pada parameter V menjadi gagal. Alur skenario serangan tersebut dapat dilihat pada Gambar 6.
e. Pada jenis klaim *secret* dengan parameter h(V) oleh *Role* R (*responder*)
   Klaim *secret* dengan parameter h(V) dapat digagalkan dengan skenario Eve membangkitkan seluruh parameter pada pesan, sehingga Bob menerima pesan seolah-seolah dari Alice. Eve mempelajari format pesan untuk dikirimkan ke Alice sampai Bon mengirimkan pesan pada send_4 ke Eve, dengan demikian Eve mempe*Role*h pesan yang ditandatangani oleh Bob, pesan ini kemudian oleh Eve di kirim ke Alice dengan dienkripsi menggunakan kunci publik Alice. Alur skenario serangan tersebut dapat dilihat pada Gambar 7.

f. Pada jenis klaim *authentication* tipe *synchronisation* oleh *Role* R (responder)
Klaim *authentication* dengan tipe *synchronisation* pada protokol Crypt0-ON2 tidak terpenuhi atau ditemukan kondisi yang tidak memenuhi untuk klaim tersebut. Klaim *authentication* tipe *synchronisation* merupakan klaim *authentication* yang relatif kompleks dari pemodelan *authentication* dari verifikasi Scyther. Klaim ini mensyaratkan setiap run dari protokol yang diinisiasi oleh agen I harus berkorespondensi satu – satu. Pada Gambar 8 dapat dilihat bahwa terdapat run yang tidak unik yaitu send_1 yang dapat diterima pada recv_1 juga recv_3, sehingga terdapat dua penerimaan atas satu sesi protokol. Hal ini menyebabkan kondisi klaim *authentication* tipe *synchronisation* tidak terpenuhi

g. Pada jenis klaim *authentication* tipe *agreement* oleh *Role* R (responder)
Klaim *authentication* dengan tipe *agreement* pada protokol *Crypt0-ON2* tidak terpenuhi atau ditemukan kondisi yang tidak memenuhi untuk klaim tersebut. Klaim *authentication* tipe *synchronisation* merupakan klaim *authentication* yang juga relatif kompleks dari pemodelan *authentication* dari verifikasi *Scyther*. Klaim ini mensyaratkan setiap *run* dari protokol yang diinisiasi oleh agen I harus sepakat dengan data yang dipertukarkan dengan pihak R. Pada Gambar 9 dapat dilihat bahwa terdapat run yang tidak unik yaitu send_1 yang dapat diterima pada recv_1 dan recv_3, sehingga terdapat dua penerimaan atas satu pengiriman. Hal ini menyebabkan kondisi klaim *authentication* tipe *agreement* tidak terpenuhi.

5. KESIMPULAN DAN SARAN

Telah dilakukan analisis terhadap protokol CryptO-0N2 dengan menggunakan *Scyther* Tool. Dari 17 klaim terkait *secrecy* dan *authentication* terdapat 10 klaim yang sukses dan 7 klaim yang gagal dengan rincian sebagai berikut:

a. Klaim Sukses (OK)
   1) Klaim *secret* A, B, T1, V, h(V), T3, T2, T4 oleh *Role* I;
   2) Klaim *authentication* berjenis alive oleh *Role* I
   3) Klaim *authentication* berjenis *alive* oleh *Role* R
b. Klaim Gagal (FAIL)
   1) Klaim *authentication* berjenis Nisynch dan Niagree oleh *Role* I;
   2) Klaim *secret* A, V, h(V) oleh *Role* R;
   3) Klaim Klaim *authentication* berjenis Nisynch dan Niagree oleh *Role* R;

Setidaknya terdapat satu *attack* terhadap protokol jika klaim *secrecy* dan *authentication* dinyatakan gagal. Skema serangan dapat dipelajari untuk proses perbaikan protokol CryptO-0N2 agar diperoleh hasil sukses untuk seluruh klaim.

DAFTAR PUSTAKA

ANTARANEWS. Pilkades Sistem e-voting Diterapkan di Musirawas [online] Tersedia di: < https://www.antaranews.com/berita/405457/pilkades-sistem-e-voting-diterapkan-di-musirawas> [Diakses 15 Oktober 2018]

CREMES, CAS, 2006. Scyther – Semantics and Verification of Security Protocols. Eindhoven: Technische Universtait Eindhoven.

CREMES, CAS, 2006. The Scyther Tool, [online] Tersedia di: < https://www.cs.ox.ac.uk/people/cas.cremers/scyther/> [Diakses 26 Mei 2017]

JATENGPOS. Tak Hanya Pilkades, Pemilihan Ketua RW di Boyolali pun pakai e-voting [online] Tersedia di: < http://www.jatengpos.com/2017/01/tak-hanya-pilkades-pemilihan-ketua-rw-di-boyolali-pun-pakai-e-voting-782695> [Diakses 15 Oktober 2018]

KOMPAS. Rekam Jejak Pemilu dari Masa ke Masa, [online] Tersedia di: < https://nasional.kompas.com/read/2018/08/06/15380041/rekam-jejak-pemilu-dari-masa-ke-masa> [Diakses 9 Oktober 2018]

LIPUTAN6. Kecurangan Pemilu, [online] Tersedia di: < http://www.liputan6.com/tag/kecuranganpe-milu> [Diakses 24 Mei 2017]

MENEZES, A, dkk., 1997. Handbook of Applied Cryptography. Florida: CRC Press Inc.

PRAKASA, P.Y., 2012. Rancang Bangun Aplikasi Secure E – Voting dengan Implementasi Protokol CryptO-0N2 untuk Pemilihan Umum Elektronis. Jakarta: Universitas Gunadharma

SCHENEIER, BRUCE, 1996. Applied Cryptography, Second Edition: Protocols, Algorithms, and Source Code in C. New Jersey: John Willey and Son Inc.

TRIBUNNEWS. Pemilihan Kepala Desa di Banyuasin Ini Menggunakan e-voting [online] Tersedia di: < http://www.tribunnews.com/regional/2015/11/11/pemilihan-kepala-desa-di-banyuasin-ini-menggunakan-e-voting> [Diakses 15 Oktober 2018]

WOLF, PETER, 2011. Introducing Electronic Voting. Swedia: International IDEA Publication Office Bulls Graphic.