CRYPTANALYSIS OF PASARGAD, A DISTANCE BOUNDING PROTOCOL BASED ON RFID SYSTEM

Mahdi Azizi[1], Nasour Bagheri[2] Abdolrasol Mirgadri[3]

[1]Faculty of Communication and Information Technology,
IHU University, Tehran, Iran,
mmazizi2006@gmail.com
[2]Electrical Engineering Department,
Shahid Rajaee Teacher Training University, Tehran, Iran,
NBagheri@srttu.edu
[3]Faculty of Communication and Information Technology,
IHU University, Tehran, Iran,
amrghdri@ihu.ac.ir

ABSTRACT

In this paper we analyze an authentication protocol so-called Pasargad which proposed by Arjemand et al. [1]. The Pasargad protocol is a distance bounding protocol which has been designed for RFID-based electronic voting systems. The designers have claimed that this protocol is more secure than Preneel and Single protocol [2], against relay attacks. However, in this paper, we present some efficient attacks against it. Our attacks include conditional impersonation attack and recovery key attack. Moreover, we show that this protocol has some structural flaw which may prevent to execution the protocol.

KEYWORDS

Distance bounding protocol, RFID, Electronic voting system, Pasargad protocol, relay attack.

1. INTRODUCTION

Radio frequency identification (RFID) system is composed of a Transponder (tag), reader and backend server. This technology can be used to link a user with a machine for authentication. First time, the RFID systems are used by the British Army during the Second World War, for identification friends/foes military aircraft. Now day, RFID systems have many applications like supply chain, access controlling, collecting road tolls, tracking animals, Passports, military and etc.

One of the applications of RFID technology is electronic voting (e-voting) which is important to governments for elections. In e-voting systems, the voter must have a smart card or memory card instead of the paper bolts in a traditional voting system. This card can be an RFID tag. An e-voting system should satisfy the following criteria

- **Authentication:** only authorized voters can vote.
- **Uniqueness :** voter cannot vote more than once.

- **Accuracy :** voting system should record the votes.
- **Integrity :** no one can modify the votes.
- **Verification :** voters should be able to verify whether votes have been counted correctly.
- **Reliability :** voting system should work robustly.
- **Secrecy :** in voting system no one should be able to determine how an individual voted.
- **Flexibility :** voting system should uses equipments that can handle verity of ballot questions formats.
- **User friendly :** voter should be able to vote with minimal equipments.

For more information on e-voting systems based on distance bounding protocols, we suggest the interested reader to read [3, 4, 5, 6, 7, 8, 9, 10]. So, we prefer reader to the significant number of publications analysis distance bounding protocols, for instance [10, 11, 12, 13].

Firstly, distance bounding protocols are introduced by Brands and Chaum [14]. Distance bounding protocols were used in some e-voting systems. So far, several protocols have been proposed based on distance bounding for e-voting systems [5, 14, 15, 16, 17]. Recently, Arjemand et al. [1] have proposed a new distance bounding protocol as called Pasargad protocol. Designers of Pasargad protocol claimed that this protocol is resistance against of known attacks.

In this paper, we show Pasargad Protocol is actually insecure by presenting conditional impersonation attack and key recovery attack.

The rest of the paper is organized by introducing the structure of Pasargad protocol which has been mentioned in section 2. After that, we analyze Pasargad protocol in section 3. Finally, the conclusion is given in section 4.

2. Description of the Pasargad Protocol

We introduce the following notations in order to proceed with developing our work.

2.1 Notations

Str_{Alice} , Str_{Bob} : String generated by Alice and Bob, respectively.

$Huff$: Huffman coding.

U_{Huff} : Decoding with Huffman algorithm.

St_1, St_2 : Source coding or output from Huffman algorithm.

P_{Pars} : Protocol Pars operation.

$U_{P_{Pars}}$: Protocol Pars decoding operation.

s_1, s_2, s_3 : Bit Strings with length $|K|$.

A_i, B_i : i-th character of Str_{Alice} and Str_{Bob}, respectively outputs Pars algorithm with 16-bits length.

a_i, b_i : i-th character of Str_{Alice} and Str_{Bob}, respectively.

K : Secret key shared between the tag and the reader.

$\oplus$: Bitwise exclusive or.

A, B : Honest prover and verifier (principals).

$\bar{A}, \bar{B}$: Malicious prover and verifier or intruders.

$A \rightarrow B$: Assigning the value of A to B.

2.2 Distance Bounding

In distance bounding protocols, it is assumed that principals (A, B, …) can compute the time of sending and receiving of messages. The most accurate method of distance estimation is to use the time of flight signal. For measuring round trip time, two principal (A and B) perform a challenge-response in the protocol. One of the entities, for example A (reader), sends a challenge and starts a timer. After receiving the challenge, B (tag) does some computations to construct the response. The response is sent back to A and the timer is stopped. The propagation speed of the signal and round trip time is known. Therefore, the distance between the reader and the tag is easily calculated.

Desmetet et al. suggested a distance bounding protocol which is resistance to mafia attack [3, 4]. The first distance bounding protocol was designed by Brands and Chaum based on the idea of Desmetet et al. In this protocol, it has been used of round trip time for exchanged messages [5]. Generally, if the tag be in admissible neighborhood of the reader then distance bounding protocol operates correctly.

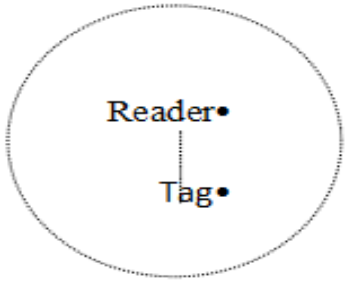

Figure 1: Tag in the admissible neighborhood of a reader [27]

Several papers have been published about distance bounding protocol in order to thwart the relay attacks [17-27]. Generally, we describe other scenario attacks for distance bounding protocol in figure 2.

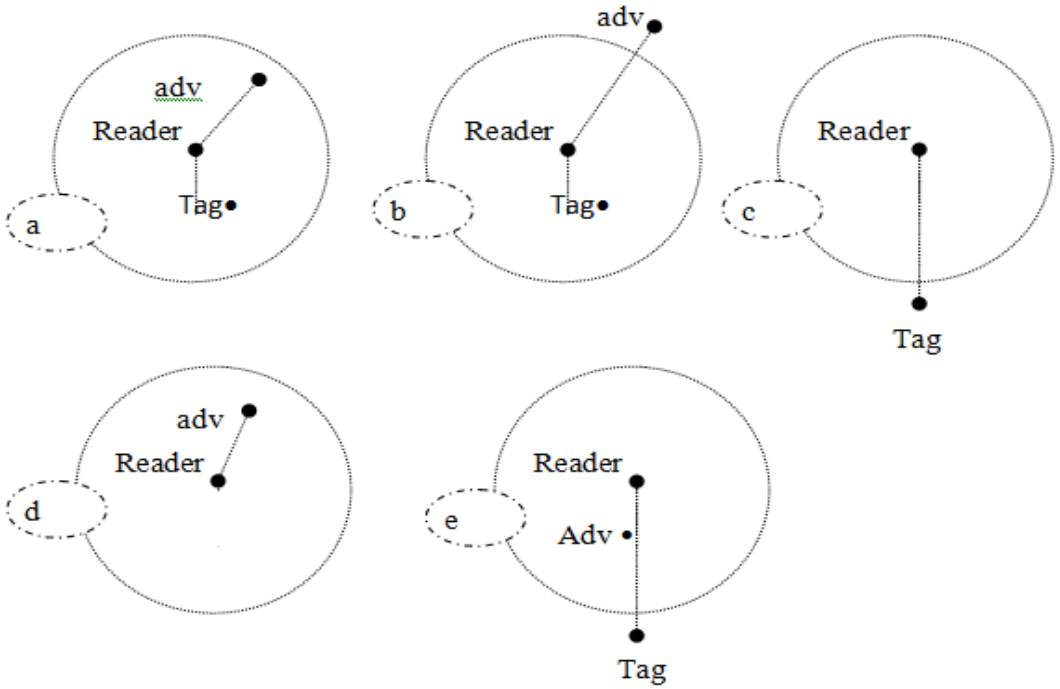

Figure 2: a) Man-in-the-middle(MIM) attack with an inside adversary, b) Man-in-the-middle(MIM) attack with an outside adversary, c) Distance fraud attack, d) Impersonation fraud attack, e) Mafia/Terrorist frauds attacks [27].

The Fig. 2-a shows that adversary can set admissible distance between the tag and the reader, then he/she does man-in-the-middle (MIM) attack. The Fig. 2-b shows that adversary can set outside distance between the tag and the reader and then he/she does MIM attack. The Fig. 2-c shows that the tag set outside distance bound of the reader and then he/she does distance fraud attack. The Fig. 2-d shows that adversary can set admissible distance bound the tag and the reader, then he/she does impersonation attack and in Fig. 2-e can be seen if adversary can set admissible distance the tag and the reader, then he/she does Mafia/Terrorist attack.

2.3 Pasargad Protocol Overview

The authors of Pasargad protocol claim that this protocol is suitable for e-voting system. The Pasargad protocol is designed based on distance bounding that prevent voter's identity falsification and the voter cannot change itself vote by mafia attack [1]. So the authors claimed that the Pasargad protocol is resistance than Preneel and Single's protocol [2]. The success probability of attacker in Preneel and Singelee's protocol, is 1/2, but in the Pasargad protocol, success probability of attacker decreases to 2^{-16}.

The structure of the Pasargad protocol includes two algorithms: Pars and Huffman algorithms, which will be described in the next subsection. The Pasargad protocol has two phases of identity verification and distance bounding identification. The identity verification phase is depicted in Fig. 3.

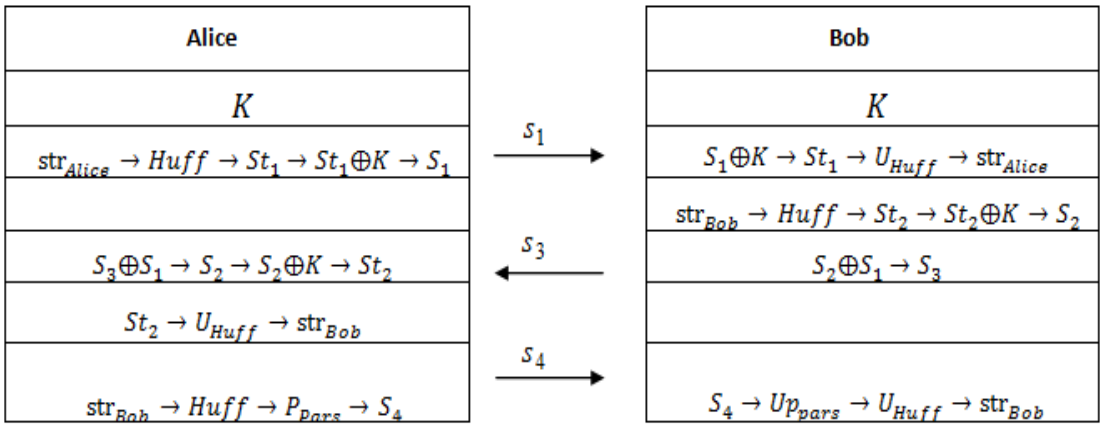

Figure 3: Identity verification [1]

2.4 Phase Detection Distance Bounding

The distance bounding identification is depicted in Fig. 4 as follows:

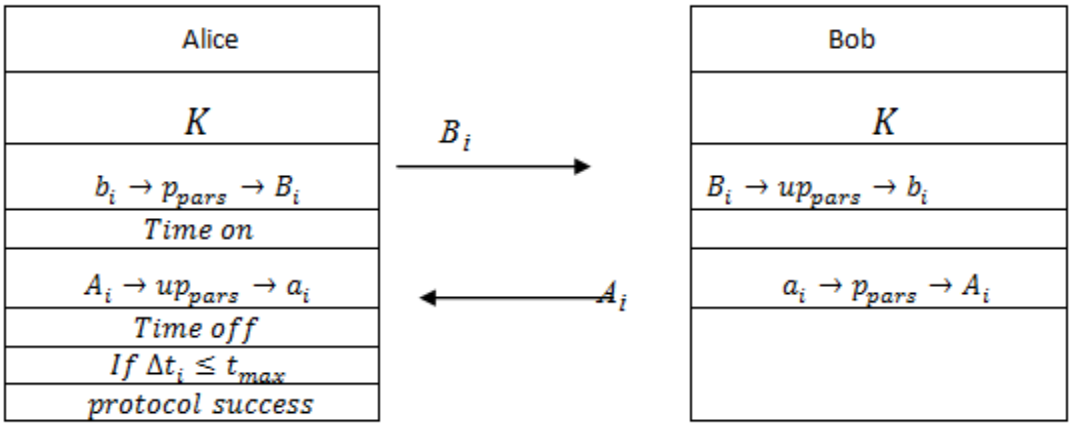

Figure 4: Distance bounding identification [1]

In this phase characters message send between Alice and Bob as follows:

- Alice runs Pars algorithm on the i^{th} character from string str_{Bob}, $(Pars(b_i)=B_i)$'
- Then Alice turns on its chronometer and sends B_i to Bob.
- Bob receives B_i and then computes b_i with pars Algorithm.
- Bob runs Pars algorithm on input a_i (the i^{th} character from string str_{Alice}) and output Pars $(a_i) = A_i$, then sends to Alice.
- Alice upon receipt A_i, decrypts it and obtains a_i then turns off its chronometer.
- Alice computes Δt, the time between send and receive message, if $\Delta t \leq t_{max}$ then protocol is run. Otherwise, the protocol has failed**.**

2.5 Pars algorithm

Now, we describe Pars algorithm, in this algorithm used ASCII code (any character in ASCII code equal one byte and we have 256 states for any byte). In Parse Algorithm encoded three characters (three byte) into four segments. For example we show encoding any byte with a symbol in table 1.

Table 1: encoding in Pars algorithm [1]

Byte 1	Byte 2	Byte 3
########	********	$$$$$$$$

These 24 bits are divided to four equal segments. Therefore, table 1 converted to table 2 as follows:

Table 2: Process of dividing three byte to four segments in pars algorithm [1]

segment 1	segment 2	Segment3	segment 4
######	## ****	**** $$	$$$$$$

Based on Table 2, instead of 256 states in ASCII code we have 64 states for any segment, which including 26 English capital letters (A,B,…,Z), 26 English lower case letters (a,b,…,z), numbers (0,1,…,9) and two characters "/" and "+". So this process makes a new encoding which is called base 64 in Table 3 as follows:

Table 3.Base 64 in Pars protocol [1]

Value	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
PPars	*A*	*B*	*C*	*D*	*E*	*F*	*G*	*H*	*I*	*J*	*K*	*L*	*M*	*N*	*O*	*P*
Value	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
PPars	*Q*	*R*	*S*	*T*	*U*	*V*	*W*	*X*	*Y*	*Z*	*a*	*b*	*c*	*d*	*e*	*f*
Value	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47
PPars	*g*	*h*	*i*	*j*	*k*	*l*	*m*	*n*	*o*	*p*	*q*	*r*	*s*	*t*	*u*	*v*
Value	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63
PPars	*w*	*x*	*y*	*z*	0	1	2	3	4	5	6	7	8	9	+	/

We illustrate encoding Pars algorithm with an example. Pick up word "Hello" according this algorithm, first "Hello" is divided into two segments "Hel" and "lo", since second part is incomplete (any segment which congruent in mod 3 is complete). If a word be incomplete then with concatenated the character "=" until word converts to complete as follows:

Table 4.ASCII code word "Hello"

Character	*H*	*e*	*l*	*l*	*o*
ASCII Value	72	101	108	108	111
Binary Value	01001000	01100101	01101100	01101100	01101111

"Hel" in ASCII code =01001000 01100101 01101100
"Hel" In Base 64= 010010 000110 010101 101100="SGVs"
"lo" in ASCII code=01101100 01101111
Since "lo" is incomplete by concatenated two "0" bits and one "=" to end of word convert to complete segment. Then run Pars algorithm as follows:
011011 000110 1111** ******
So, "lo" in Base 64 converts to "bg8=" and word "Hello" encodes to"SGVsbg8="[1].

2.6 Huffman coding

Pasargad protocol uses Huffman algorithm. In this method for two strings with equal length, if frequency be different for letters then length of coding will be different. This property creates a defect for Pasargad Protocol which we will use of this defect in structural analysis later. For clear this property we give following example:

The frequency of the letters in phrase "The Pasargad is good" is given in table 5.

Table 5.Frequancy letters

symbol	Frequency	symbol	Frequency
T	1	r	1
h	1	g	2
e	1	d	2
p	1	i	1
a	3	o	2
s	2	$space$	3

So, the converting of the phrase "The Pasargad is good" according to above table with using Huffman coding shows in table 6 as follows:

Table 6: Huffman code

symbol	T	h	e	p	a	s
code	0000	0001	0010	0011	100	1010
symbol	r	g	d	i	o	$space$
code	0100	1011	110	0101	111	011

00000001001001100111001010100010010111001100110101101001110111111111110
This code has 70 bits length [1].

If we want encode another string with the same length, but different letters, for example the phrase "Huffman code is good" is given in table 7 as follows.

Table 7: Frequency letters of "Huffman code is good"

symbol	Frequency	symbol	Frequency
H	1	i	1
u	1	s	1
m	1	g	1
a	1	f	2
n	1	d	2
c	1	o	3
e	1	$space$	3

Then, the output length of this phrase is 74 bits.

3. Cryptanalysis of Pasargad Protocol

We analyze the Pasargad protocol as several view point. Also, we argue structural analysis, conditional impersonation and key recovery attack.

3.1 Structural Analysis

In the first stage of phase identity verification, Alice chooses a String ($\text{Str}_{\text{Alice}}$) and computes S_1 by using Huffman algorithm as follow:

$$\text{str}_{Alice} \rightarrow Huff \rightarrow St_1 \rightarrow St_1 \oplus K \rightarrow S_1$$

As already mention in subsection 2-6, Huffman algorithm cannot generate two strings with same length for two messages with different frequency letters and the same length. So, in the first stage cannot be carried out and protocol fails. We suggest cancel Huffman algorithm from this protocol.

Another ambiguous in Pasargad protocol chooses 16 bits in distance bounding identification phase, which unadoptable with pars algorithm.

3.2 Conditional Impersonation attack

We analyze the security Pasargad protocol and demonstrate several weaknesses on this protocol.

As already mention in section 2 (Fig. 2), we can choose one of those scenario for identity verification phase and then attack as follows:

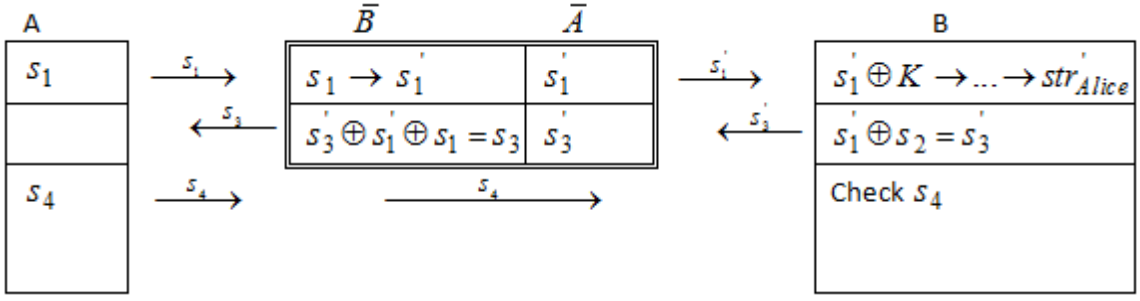

Figure 5: Conditional impersonate A if you can obtain key

In this attack adversary eavesdrops first message from A and converts to another message s_1' then sends to B. Next B receives message from adversary then protocol generates message s_3' and sends by channel. Only the adversary can convert this message to message s_3, in last step A generates message s_4 and sends to B without changing by the Adversary. Finally B accepted A (without received message from A) with received message from the adversary. If the adversary have had key then could be impersonated A and B (Tag and reader).

3.3 Key Recovery

In the Pasargad protocol used of two algorithms; Pars and Huffman which only encode data. So, we show how an adversary is able to recover key with following theorem:

Theorem In the pasargad protocol, a passive attacker, after eavesdropping one authentication session between the tag and the reader, can recovery key from $St_2 \oplus K = S_2 \rightarrow K$.

Proof: An adversary can eavesdrop some information after eavesdropping a session of the pasargad protocol.

- First, adversary can eavesdrop S_1, S_3 and S_4 from channel.
- Second, adversary can compute Str_{Bob} from inverse of following relation

- Third, adversary can compute St_2, and S_2 from Huffman coding and Pasargad protocol respectively.
- Finally, adversary can find key by computing :

$$St_2 \oplus K = S_2 \rightarrow K \; .\blacksquare$$

Thus, we can impersonate tag with recover the key completely.

4. Conclusion

In this paper, we have analyzed the structure and the security of a distance bounding protocol called Pasargad. First, we analyzed the structure of Pasargad protocol and showed that Huffman algorithm cannot generate two strings with same length for two messages with different frequency. So, the protocol cannot be carried out with Huffman coding and must be remove Huffman algorithm. Second, we presented an attack against the Pasargad protocol named Conditional Impersonation attack. Finally, we prove that an adversary can recover key of the Pasargad protocol. Hence, we conclude that the Pasargad protocol is not only unsecure but also is not suitable for e-voting systems.

REFERENCES

[1] M. Arjemand, M. Gardashi, R. Taheri zohur and M. Kazemi. Providing a Distance Bounding Protocol Named Pasargad In order to Defend Against Relay Attacks on RFID-Based Electronic Voting system. In International Journal of Ubicomp (IJU), vol.2, No.3, July 2011.

[2] D. Singelee, B. Preneel, Distance Bounding in Noisy Environments. In Proceedings of the 4th European conference on Security and privacy in ad-hoc and sensor networks, ESAS'07, pages 101–115, Berlin, Heidelberg, 2007. Springer-Verlag

[3] S. Bengio, G. Brassard, Y. Desmedt, C. Goutier, and J.-J. Quisquater. Secure Implementation of Identification Systems. Journal of Cryptology, 4(3):175-183, 1991.

[4] T. Beth and Y. Desmedt. Identification Tokens for: Solving the Chess Grandmaster Problem. In CRYPTO, volume 537 of Lecture Notes in Computer Science, pages 169-177, Santa Barbara, California, USA, August 1990. Springer-Verlag.

[5] S. Brands and D. Chaum. Distance-Bounding Protocols. In Advances in Cryptology { EU-ROCRYPT'93, volume 765 of Lecture Notes in Computer Science, pages 344-359, Lofthus, Norway, May 1993. Springer-Verlag.

[6] L. Bussard and W. Bagga. Distance-Bounding Proof of Knowledge to Avoid Real-time Attacks. IFIP International Federation for Information Processing, pages 223-238, Chiba, Japan, May-June 2005. Springer-Verlag.

[7] T. Kohno, A. Stubbefield, A. D. Rubin and D. S. Wallach. Analysis of an Electronic Voting System. IEEE Computer Society Press, 2004.

[8] E.Shan. End-to-End Verifiability for Optical Scan Voting Systems. Master thesis, MIT, Department of Electrical Engineering and Computer Science, 2008

[9] B. Adida. Advance in Cryptographic Voting Systems. PhD thesis, MIT, Department of Electrical Engineering and Computer Science, 2006

[10] C. H. Kim and G. Avoine. RFID Distance Bounding Protocols with Mixed Challenges. IEEE Trans. Wireless Commune., vol. 10, no. 5, pp. 1618–1626, 2011

[11] C. Hee Kim. Security Analysis of YKHL Distance Bounding Protocol with Adjustable False Acceptance Rate. IEEE COMMUNICATIONS LETTERS, VOL. 15, NO. 10, 2011

[12] G. Avoine, M. Bingol, S. Kardas, C. Lauradoux, and B. Martin. A Framework for Analyzing RFID Distance Bounding Protocols. J.Computer Security, vol. 19, no. 2, pp. 289–317, 2010

[13] Y. Oren and A. Wool. Attacks on RFID-Based Electronic Voting Systems. ePrint Archive, 2009

[14] S. Capkun and J.-P. Hubaux. Secure Positioning in Wireless Networks. IEEE Journal on Selected Areas in Communications, 24(2), February, 2006.

[15] S. Brands and D. Chaum. Distance-Bounding Protocols. In Workshop on the theory and application of cryptographic techniques on Advances in cryptology, pages 344–359. Springer-Verlag New York, Inc., 1994.

[16] G.P. Hancke and M.G. Kuhn. An RFID Distance Bounding Protocol. Proceedings of the IEEE/Create-Net Secure Comm, 67- 73, 2005.

[17] C. Meadows, R. Poovendran, D. Pavlovic, L.W. Chang, and P. Syverson. Distance Bounding Protocols: Authentication Logic Analysis and Collusion Attacks. Secure Localization and Time Synchronization for Wireless Sensor and Ad Hoc Networks, 279- 298, Springer-Verlag, 2007.

[18] J. Reid, J.M.G. Nieto, T. Tang, and B. Senadji. Detecting Relay Attacks with Timing-Based Protocols. Proceedings of the 2nd ACM Symposium on Information, Computer, and Communications Security, 204-213, 2007.

[19] G. Hancke. A Practical Relay Attack on ISO 14443 Proximity Cards. Manuscript, February 2005.

[20] S. Drimer and S. J. Murdoch. Keep your enemies close: Distance Bounding against Smartcard Relay Attacks. USENIX Security Symposium, pp 87-102, August 2007.

[21] G. Hancke and M. Kuhn. An RFID Distance Bounding Protocol. In Conference on Security and Privacy for Emerging Areas in Communication Networks - SecureComm 2005, Athens, Greece, September 2005.

[22] M. Hlavác and T. Rosa. A Note on the Relay Attacks on e-Passports: The Case of Czech e-Passports. Cryptology ePrint Archive, Report 2007/244, 2007.

[23] Y.-J. Tu and S. Piramuthu. RFID Distance Bounding Protocols. In First International EURASIP Workshop on RFID Technology, Vienna, Austria, September 2007.

[24] G. Kapoor, W. Zhou, and S. Piramuthu. Distance Bounding Protocol for Multiple RFID Tag Authentication. In EUC '08: Proceedings of the 2008 IEEE/IFIP International Confer- ence on Embedded and Ubiquitous Computing, pages 115-120, Shanghai, China, December 2008.

[25] C. H. Kim and G. Avoine. RFID Distance Bounding Protocol with Mixed Challenges to Prevent Relay Attacks. 8th International Conference on Cryptology And Network Security, CANS'09, volume 5888 of Lecture Notes in Computer Science, pages 119-133, Kanazawa, Ishikawa, Japan, December 2009.

[26] G. Avoine and A. Tchamkerten. An Efficient Distance Bounding RFID Authentication Protocol: Balancing False-Acceptance Rate and Memory Requirement. In Information 21 Security Conference - ISC'09, volume 5735 of Lecture Notes in Computer Science, pages 250-261, Pisa, Italy, September 2009.

[27] G. Avoine1, M. A.Bingol, S. KardasC. Lauradoux, B. Martin. A Framework for Analyzing RFID Distance Bounding Protocols. Journal of Computer Security, August 2010.

Authors

Mahdi Azizi received his M.S. degree in Communications, Cryptology &Information Security in 2005. Currently, he is a Ph.D. candidate at the Department of Information and Communication Technology I.H University, Tehran, Iran. His research interests include RFID security, authentication protocols and Cryptanalysis.

Nasour Bagheri is a lecturer at Electrical Engineering Department, Shahid Rajaee Teacher Training University, Tehran, Iran. He is the author several articles in information security and cryptology. Homepage of the author is available at: http://n-bagheri.srttu.ir/

Abdolrasoul Mirghadri received the B.Sc., M.Sc. and Ph.D degrees in Mathematical Statistics, from the faculty of Science, Shiraz Universityin 1986, 1989 and 2001, respectively. He is an associate professor at the faculty and research center of communication and information technology, IHU, Tehran, Iran since 1989. His research interest includes: Cryptography, Statistics and Stochastic Processes.He is a member of ISC, ISS and IMS