مشاهداتی روی یک طرح احراز اصالت سبک‌وزن با قابلیت گمنامی و اعتماد در اینترنت اشیا

جواد علیزاده [۱] و نصور باقری [۲]

[۱] مرکز علم و فناوری فتح، دانشکده و پژوهشکده فناوری اطلاعات و ارتباطات، دانشگاه جامع امام حسین(ع)، تهران، ایران

[۲] دانشکده مهندسی برق، دانشگاه تربیت دبیر شهید رجائی، تهران، ایران

چکیده

امروزه اینترنت اشیا (IoT) مدام در حال پیشرفت و فراگیرشدن است. دو چالش اساسی در این فناوری، کارایی و امنیت اطلاعات و ارتباطات است. پروتکل‌های احراز اصالت و توافق کلید (AKA) نقش مهمی در امنیت اینترنت اشیا دارند. شبکه‌های حس‌گر بی‌سیم (WSN) یک مؤلفه مهم در برخی کاربردهای IoT هستند. در سال ۲۰۱۹، جانبابائی و همکاران یک پروتکل AKA سبک‌وزن برای WSN ارائه و ادعا کردند ویژگی‌های امنیتی مانند گمنامی و محرمانگی را تامین می‌کند. در این مقاله، چند آسیب‌پذیری مهم و غیر بدیهی از این طرح ارائه می‌شود. دقیق‌تر اینکه نشان داده می‌شود هنگام برقراری نشست با استفاده از این پروتکل، یک حس‌گر بدخواه می‌تواند پارامترهای محرمانه یک حس‌گر دیگر را به دست آورد. علاوه بر این نشان داده می‌شود یک مهاجم با داشتن تنها یک کلید نشست شناخته شده، می‌تواند هر کلید نشست دیگر توافق شده میان حس‌گر ها را به‌دست آورد. با توجه به این ضعف‌ها، حملاتی مانند حمله جعل گره حس‌گر و مردی در میانه روی پروتکل جانبابائی و همکاران عملی است و می‌توان نشان داد این طرح، برخلاف ادعای مؤلفان، نمی‌تواند ویژگی گم‌نامی گره‌های حس‌گر را تأمین کند. ضعف مهم این طرح مربوط به انتقال کلید نشست بدون استفاده از تابع چکیده‌ساز روی آن است که برای رفع آن یک پیشنهاد ساده ارائه می‌شود.

واژگان کلیدی: اینترنت اشیا، شبکه حس‌گر بی‌سیم، احراز اصالت و توافق کلید، گمنامی

Some observations on a lightweight authentication scheme with capabilities of anonymity and trust in Internet of Things (IoT)

Javad Alizadeh*[1] and Nasour Bagheri[2]

1 Fath Center, Faculty and Research Center of Communication and Information Technology, Imam Hossein University, Tehran, Iran

2 Electrical Engineering Department, Shahid Rajaee Teacher Training University, Tehran, Iran

Abstract

Over the last years, the concept of Internet of Things (IoT) leads to a revolution in the communications of humans and things. Security and efficiency could be the main challenges of that communication. On the other hand, authenticity and confidentiality are two important goals to provide desired security in an information system, including IoT-based applications. An Authentication and Key Agreement (AKA) protocol is a tool to achieve authenticity and agree on a secret key to reach confidentiality. Therefor using a secure AKA protocol, one can establish the mentioned security. In the last years, several articles have discussed AKA protocols in the WSN. For example, in 2014, Turkanovic et al. proposed a new

* نویسندهٔ عهده‌دار مکاتبات

* Corresponding author

• تاریخ ارسال مقاله: ۱۳۹۹/۶/۲۴ • تاریخ پذیرش: ۱۴۰۰/۹/۲۰ • تاریخ انتشار: ۱۴۰۱/۱۲/۲۹ • نوع مطالعه: پژوهشی

[DOI: 10.61186/jsdp.19.4.85]

AKA scheme for the heterogeneous ad-hoc WSN. In 2016, Sabzinejad et al. presented an improved one. In 2017, Jiang et al. introduced a secure AKA protocol. Some other AKA protocols have presented in the last three years. All the mentioned protocols are lightweight ones and need minimum resources and try to decrease the computation and communication costs in the WSN context.
In 2019, Janababaei et al. proposed an AKA scheme in the WSN for the IoT applications, in the journal of Signal and Data Processing (JSDP). In the context of efficiency, the protocol only uses a hash function, bitwise XOR, and concatenation operation. Hence, it can be considered as a lightweight protocol. The authors also discussed the security of their scheme and claimed that the proposed protocol has the capability to offer anonymity and trust and is secure against traceability, impersonation, reply and man in the middle attacks. However, despite their claims, this research highlights some vulnerabilities in that protocol, for the first time to the best of our knowledge. More precisely, we showe that a malicious sensor node can find the secret parameters of another sensor node when it establishes a session with the victimized sensor. Besides, an adversary can determine any session key of two sensor nodes, given only a known session key of them. We also show that the protocol could not satisfy the anonymity of the sensor nodes. Other attacks which influence the Janababaei et al.'s scheme, are impersonation attack on the sensor nodes and cluster heads and also the man in the middle attack.
In this paper we find that the main weaknesses of the Janababaei et al.'s protocol are related to computation of the session key, SK. We also propose a simple remedy to enhance the security of the Janababaei et al.'s protocol. An initial attempt to improve the protocol is using a hash function on the calculated key, SK. This suggestion is presented to enhance the security of the protocol against the observed weaknesses in this paper; but it does not mean that there are no other security issues in the protocol. Therefore, modification and improvement of the Janababaei et al.'s protocol such that it provides other security features can be considered in the future research of this paper. Besides, since in this paper we focus on the security of the protocol, then the efficiency of it was not discussed. Therefore one can consider the modification of the message structure of the protocol to reduce the computational and telecommunication costs of it as another future work in the context of this paper.

Keywords: Internet of Things, Wireless Sensor Network, Authentication and Key Agreement, Anonymity

۱- مقدمه

در سال‌های اخیر، مفهوم اینترنت اشیا (IoT) منجر به یک تحول اساسی در ارتباطات میان انسان‌ها و نیز اشیا شده است. به‌طوری‌که امروز می‌توان کاربردهای بسیاری پیدا کرد که مبتنی بر اینترنت اشیا هستند. برای مثال، اینترنت حس‌گرها[1] (IoS) یک نمونه از چنین کاربردهایی هستند که در زمینه شبکه‌های حس‌گر بی‌سیم (WSN) مهم می‌باشند [1,2]. چالش‌های مهم در حوزه اینترنت اشیا را می‌توان در دو دسته کلی امنیت و کارآیی در نظر گرفت. امنیت به‌صورت حفاظت اطلاعات و سامانه‌های اطلاعاتی از دسترسی‌ها، استفاده‌ها، کشف‌ها، تغییرات و اصلاحات غیر مجاز تعریف می‌شود [3]. همچنین کارآیی را می‌توان با توجه به هزینه‌های ارتباطی، حافظه و محاسباتی تعریف نمود.

برای تأمین امنیت در یک سامانه امنیتی، لازم است تا افراد استفاده‌کننده از سامانه یادشده، یکدیگر را احراز اصالت و یک کلید محرمانه میان خود برقرار کنند تا بتوانند با استفاده از آن در کنار الگوریتم‌های رمزنگاری از اطلاعات محرمانه خود در یک کانال ارتباطی محافظت کنند. برای رسیدن به هدف احراز اصالت می‌توان از یک پروتکل احراز اصالت و نیز برای رسیدن به هدف برقراری کلید محرمانه، می‌توان از یک پروتکل توافق کلید استفاده کرد. طرح‌هایی وجود دارند که می‌توانند دو هدف یادشده را هم‌زمان تأمین کنند. این طرح‌ها، پروتکل‌های احراز اصالت و توافق کلید (AKA) نامیده می‌شوند.

در سال‌های اخیر، فعالیت‌های متعددی در رابطه با مطالعه و بررسی پروتکل‌های AKA در WSN صورت گرفته است. برای مثال در سال ۲۰۱۴، *تورکانوویچ*[2] و همکاران [4] یک طرح AKA جدید برای WSNهای خاص‌منظوره ناهمگن[3] ارائه کردند که چون تنها از تابع چکیده‌ساز و عمل‌گر *xor* استفاده می‌کرد، یک طرح سبک‌وزن بود. در سال ۲۰۱۶، *سبزی‌نژاد* و همکاران [5] برخی ضعف‌های امنیتی برای پروتکل یادشده گزارش کردند. آنها همچنین یک پروتکل AKA جدید و بهبودیافته ارائه کردند، به‌طوری‌که ضعف‌های پروتکل تورکانوویچ و همکاران را نداشت. *امین*[4] و *بیسواس*[5] [6] در یک کار دیگر، ضعف‌های امنیتی دیگری از طرح ارائه‌شده در [4] به‌دست آورند و یک پروتکل AKA جدید دیگر معرفی کردند. پروتکل سبزی‌نژاد و همکاران در [7] تحلیل

[1] Internet of Sensors (IoS)

[2] Turkanovic
[3] Heterogeneous ad hoc WSN
[4] Amin
[5] Biswas

[DOI: 10.61186/jsdp.19.4.85]

و ارزیابی و بهبود داده شد. به‌طور مشابه پروتکل امین و همکاران نیز در [8] مورد تحلیل قرار گرفت و بهبود داده شد. در سال ۲۰۱۷، *جیانگ*[1] و همکاران [9] نشان دادند که پروتکل معرفی‌شده در [8] در برابر برخی حملات شناخته شده مقاوم نیست؛ سپس آنها یک پروتکل AKA جدید معرفی کردند.

علاوه‌بر طرح‌های یادشده، برخی پروتکل‌های AKA جدید نیز وجود دارند که در سال‌های اخیر ارائه شده‌اند. برای نمونه می‌توان به پروتکل‌های AKA در [10-14] اشاره کرد. تمام پروتکل‌های یادشده، پروتکل‌های AKA سبک‌وزن هستند که منابع کمی برای پیاده‌سازی لازم دارند و تلاش می‌کنند تا هزینه ارتباطات و محاسبات در WSN را کاهش دهند.

در سال ۲۰۱۹، *جانبابائی* و همکاران [1] یک طرح AKA برای شبکه‌های حس‌گر بی‌سیم در کاربردهای اینترنت اشیا ارائه کردند. از نقطه نظر کارآیی، این طرح تنها از تابع چکیده‌ساز و عمل‌گر xor استفاده می‌کند؛ بنابراین می‌توان آن را به عنوان یک پروتکل AKA سبک‌وزن در نظر گرفت. طراحان همچنین امنیت طرح خود را مورد مطالعه و بررسی قرار دادند و ادعا کردند که این طرح قابلیت تامین گمنامی گره‌های حس‌گر و اعتماد را دارد و در برابر حملات شناخته شده مانند ردیابی، جعل، بازخوردی و مردی در میانه امن است. برخلاف ادعای ایشان، می‌توان برخی آسیب‌پذیری‌های جدی از این پروتکل به دست آورد و نشان داد که آن یک طرح ناامن است. برای مثال در استفاده از پروتکل جانبابائی و همکاران، یک گره حس‌گر بدخواه می‌تواند پارامترهای محرمانه یگ گره دیگر را که با او در ارتباط است، به‌دست آورد. یا اگر یک مهاجم بتواند تنها یک کلید نشست میان دو گره حس‌گر را به‌دست آورد، او می‌تواند با استفاده از این کلید، کلیدهای نشست‌های بعدی یا قبلی را به‌دست آورد. با داشتن اطلاعات محرمانه یک گره حس‌گر به‌ترتیبی که یاد شد، یک مهاجم می‌تواند این گره حس‌گر را جعل یا او را ردیابی کند؛ بنابراین ادعای مؤلفان در مورد امنیت پروتکل ایشان، به‌خصوص ویژگی تأمین گم‌نامی آن، نقض می‌شود.

بقیه مطالب این مقاله به‌صورت زیر سازمان‌دهی شده است. طرح AKA جانبابائی و همکاران در بخش ۲ مرور می‌شود. در بخش ۳، آسیب‌پذیری‌های این طرح و ضعف‌های امنیتی آن در برابر حملاتی مانند کشف کلید و اطلاعات محرمانه، ردیابی، جعل و مردی در میانه تشریح می‌شود. در بخش ۴ پیشنهادهایی جهت بهبود پروتکل جانبابائی و همکاران و افزایش امنیت آن در برابر حملات بحث شده در بخش ۳ این مقاله ارائه می‌شود. درنهایت جمع‌بندی و نتیجه‌گیری مقاله در بخش ۵ بیان می‌شود.

۲- مرور پروتکل جانبابائی و همکاران

در این بخش پروتکل AKA جانبابائی و همکاران به‌اختصار معرفی می‌شود. همان‌طورکه گفته شد، این پروتکل یک پروتکل سبک‌وزن برای شبکه‌های حس‌گر بی‌سیم است که می‌تواند به صورت هم‌زمان احراز اصالت گره‌های حس‌گر را انجام داده و یک کلید محرمانه میان آنها به اشتراک گذارد. جانبابائی و همکاران برای ارائه طرح خود از یک معماری سلسله‌مراتبی استفاده کردند که در آن گره‌های حس‌گر داخل خوشه‌هایی[2] دسته‌بندی می‌شوند. گره‌های حس‌گر که با SN نمایش داده می‌شوند، سرخوشه‌ها که با CH نشان داده می‌شوند و نیز ایستگاه‌های پایه[3] که با BS نمایش داده می‌شوند، مؤلفه‌های اصلی در معماری سلسله‌مراتبی استفاده شده در [1] هستند.

در معماری مورد نظر، اگر گره‌های حس‌گر در یک خوشه یکسان قرار داشته باشند، می‌توانند به‌طور مستقیم و بدون واسطه با یکدیگر ارتباط برقرار کنند؛ در غیر این صورت، یعنی اگر آنها در دو خوشه متفاوت قرار داشته باشند، به‌کمک سرخوشه‌ها ارتباط برقرار می‌کنند. در ادامه این مقاله حالتی که در آن دو گره حس‌گر در یک خوشه یکسان قرار دارند، در نظر گرفته می‌شود. حالت دیگر که تفاوت جزئی با حالت یادشده دارد در [1] توصیف شده است.

پروتکل جانبابائی و همکاران دو مرحله اصلی دارد که شامل مرحله ثبت نام و مرحله احراز اصالت و توافق کلید است. برای توصیف این مراحل از نمادگذاری معرفی شده در جدول ۱ استفاده می‌شود.

(جدول ۱): نمادگذاری استفاده شده در این مقاله
(Table 1): List of notations used in this paper

توضیحات	نماد
گره حس‌گر	SN_i
سرخوشه	CH_i
شناسه حس‌گر / سرخوشه	ID_i / ID'_i
تک‌شمار تصادفی محرمانه گره حس‌گر	N_i

[1] Jiang

[2] Clusters
[3] Base Stations

[DOI: 10.61186/jsdp.19.4.85]

کلید محرمانه مشترک میان گره حس‌گر و سرخوشه	K_i
شناسه مستعار گره حس‌گر / سرخوشه	AID_i / AID'_i
یک عدد تصادفی	Tr_i
کلید نشست به اشتراک گذاشته شده میان دو گره حس‌گر	SK
تابع چکیده‌ساز	$h(.)$
عملگر xor	$\oplus$
عملگر الحاق دو رشته بیت	$\|\|$
مقایسه مقدار U با V	$U =? V$

۱-۲-مرحله ثبت نام

این مرحله از پروتکل شامل دو نوع ثبت نام است. نوع نخست ثبت نام گره‌های حس‌گر توسط سرخوشه‌ها و نوع دوم ثبت نام سرخوشه‌ها به‌وسیلهٔ گره‌های درگاه[1] است. در نوع نخست، یک گره حس‌گر ، هویت واقعی خود یعنی ID_i را برای یک سرخوشه، تحت یک کانال ارتباطی امن ارسال، سپس سرخوشه یک کلید محرمانه مثل K_i و یک عدد تصادفی مثل Tr_i تولید کرده و مقدار $AID_i = h(ID_i||Tr_i)$ را به‌عنوان یک هویت مستعار[2] برای گره حس‌گر حساب می‌کند. درنهایت سرخوشه مقادیر AID_i, ID'_i, Tr_i, K_i را برای گره حس‌گر ، تحت همان کانال ارتباطی قبلی، ارسال می‌کند. روند ثبت نام توضیح داده‌شده در شکل (۱) خلاصه شده است.

Sensor Node	Cluster Head
$ID_i \rightarrow$	
	Generate Tr_i, K_i Compute $AID_i = h(ID_i\|\|Tr_i)$ $\leftarrow (AID_i, ID'_i, Tr_i, K_i)$

(شکل ۱): روند ثبت نام یک گره حس‌گر توسط یک سرخوشه

(Figure 1): The registration procedure of a sensor node with a cluster head

ثبت نام یک سرخوشه توسط یک گره درگاه نیز با روال مشابه با آنچه که توضیح داده شد، انجام می‌شود. از آنجا که این بخش نقشی در تحلیل‌های ارائه‌شده در این مقاله ندارد، از توضیح آن صرف نظر شده است.

۲-۲- مرحله احراز اصالت و توافق کلید

در مرحله احراز اصالت و توافق کلید پروتکل جانبابائی و همکاران، دو گره حس‌گر به‌صورت دوطرفه، هویت یکدیگر را احراز کرده و یک کلید نشست محرمانه مانند SK باهم به اشتراک می‌گذارند. بسته به موقعیت گره‌ها و اینکه آنها در یک خوشه یکسان هستند یا در خوشه‌های متفاوت قرار گرفته‌اند، این مرحله دو روال متفاوت خواهد داشت. در ادامه این مقاله حالتی توصیف می‌شود که در آن دو گره حس‌گر یک خوشه یکسان قرار دارند. برای حالت دیگر، لازم است تا سرخوشه‌ها نیز احراز اصالت شوند [1].

یک سرخوشه نقش مهمی در احراز اصالت و توافق کلید میان دو گره حس‌گر بازی می‌کند؛ درواقع، این سرخوشه یک نهاد سوم مورد اطمینان برای دو گره حس‌گر است که با هر کدام از گره‌ها یک کلید مشترک محرمانه اختصاصی به اشتراک گذاشته است. سرخوشه همچنین کلید نشست میان دو گره SN_i و SN_j را به صورت

$$SK = N_i \oplus N_j$$

تولید می‌کند که در آن N_i و N_j به‌ترتیب یک پارامتر محرمانه برای SN_i و SN_j است. بعد از اینکه سرخوشه مقدار کلید نشست را محاسبه کرد، آن را میان دو گره حس‌گر توزیع می‌کند. فرآیند مربوط به احراز اصالت و توافق کلید میان دو گره حس‌گر با کمک یک سرخوشه در شکل (۲) نشان داده شده است و به‌صورت زیر توصیف می‌شود.

گره حس‌گر SN_i یک تک‌شمار[3] تصادفی محرمانه مانند N_i تولید کرده و محاسبات زیر را انجام می‌دهد.

$$A = N_i \oplus K_i,$$
$$V_i = h(AID_i \parallel AID_j \parallel N_i \parallel Tr_i).$$

سپس، پیام:

$$M_1 = \{AID_i, AID_j, Tr_i, V_i, A\}$$

را برای سرخوشه، تحت یک کانال ناامن ارسال می‌کند. سرخوشه بعد از دریافت پیام M_1، مقدار Tr_i را در پایگاه داده خود جستجو می‌کند و براساس آن مقدار K_i را به‌دست آورده و $N_i = A \oplus K_i$ را حساب می‌کند؛ سپس آن

$$V'_i = h(AID_i \parallel AID_j \parallel N_i \parallel Tr_i)$$

را حساب کرده و مقدار آن را با V_i دریافتی مقایسه می‌کند تا اصالت گره حس‌گر SN_i را احراز کند. اگر احراز اصالت گره حس‌گر موفقیت‌آمیز بود، سرخوشه پیام $M_2 = \{AutReq\}$ را برای گره حس‌گر با هویت مستعار AID_j، یعنی گره حس‌گر SN_j ارسال کرده و از آن می‌خواهد تا هویت خود را برای سرخوشه اثبات کند. گره حس‌گر SN_j نیز یک تک‌شمار تصادفی محرمانه مانند N_j تولید کرده و محاسبات زیر را انجام می‌دهد.

$$B = N_j \oplus K_j,$$
$$V_j = h(AID_j \parallel N_j \parallel Tr_j).$$

[1] Gateway Nodes
[2] Alias Identity
[3] Nonce

[DOI: 10.61186/jsdp.19.4.85]

سپس، پیام:

$M_3 = \{AID_j, Tr_j, V_j, B\}$

را برای سرخوشه، تحت یک کانال ناامن ارسال می‌کند. سرخوشه نیز تلاش می‌کند تا هویت گره حس‌گر SN_j را مشابه با روالی که در مورد گره حس‌گر SN_i توصیف شد، احراز کند. بعد از این کار، سرخوشه مقادیر Tr_i، Tr_j، AID_i و AID_j را به‌روز می‌کند؛ درنهایت آن یک کلید محرمانه نشست میان گره‌های حس‌گر SN_i و SN_j به‌صورت $SK = N_i \oplus N_j$ تولید می‌کند. برای توزیع این کلید میان گره‌های حس‌گر ، سرخوشه محاسبات زیر را انجام می‌دهد:

$SK_i = (Tr_i \parallel SK) \oplus h(K_i \parallel AID_j),$
$SK_j = (Tr_j \parallel SK) \oplus h(K_j \parallel AID_i),$
$V_1 = h(AID_i \parallel SK_j \parallel N_j \parallel Tr_j),$
$V_2 = h(AID_j \parallel SK_i \parallel N_i \parallel Tr_i).$

در ادامه سرخوشه پیام‌های M_4 و M_5 را به ترتیب برای گره‌های حس‌گر SN_i و SN_j ارسال می‌کند که در آن

$M_4 = \{SK_i, V_2, Tr_i\},$
$M_5 = \{AID_i, SK_j, V_1, Tr_j\}.$

در انتهای این مرحله، گره حس‌گر SN_i (و نیز SN_j) مقادیر SK_i و V_2 (و نیز SK_j و V_1) را به کار می‌برد تا کلید نشست را محاسبه و بررسی و همچنین مقدار Tr_i (و نیز Tr_j) را به روز نماید. دو گره حس‌گر SN_i و SN_j همچنین هویت‌های مستعار خود را به‌صورت زیر به‌روز می‌کنند.

$AID_i = h(ID_i || N_1),$
$AID_j = h(ID_j || N_2).$

که در آن N_1 و N_2 مقادیر تصادفی هستند که به‌وسیلهٔ سرخوشه تولید می‌شوند.

Sensor Node i	Cluster Head	Sensor Node j
$N_i,$ $A = N_i \oplus K_i,$ $V_i = h(AID_i \parallel AID_j \parallel N_i \parallel Tr_i),$ $M_1 = \{AID_i, AID_j, Tr_i, V_i, A\} \rightarrow$		
	Check $Tr_i,$ Get $N_i,$ Verify $V_i,$ $M_2 = \{AutReq\} \rightarrow$	
		$N_j,$ $B = N_j \oplus K_j,$ $V_j = h(AID_j \parallel N_j \parallel Tr_j),$ $\leftarrow M_3 = \{AID_j, Tr_j, V_j, B\}$
	Check $Tr_j,$ Get $N_j,$ Verify $V_j,$ Generate $N_1, N_2,$ Update $AIDnew_i = h(ID_i \parallel N_1),$ Update $AIDnew_j = h(ID_j \parallel N_2),$ $Tr_i = N_1, \quad Tr_j = N_2,$ Compute $SK = N_i \oplus N_j,$ $SK_i = (Tr_i \parallel SK) \oplus h(K_i \parallel AID_j),$ $SK_j = (Tr_j \parallel SK) \oplus h(K_j \parallel AID_i),$ $V_1 = h(AID_i \parallel SK_j \parallel N_j \parallel Tr_j),$ $V_2 = h(AID_j \parallel SK_i \parallel N_i \parallel Tr_i),$ $AID_i = AIDnew_i,$ $AID_j = AIDnew_j,$ $\leftarrow M_4 = \{SK_i, V_2, Tr_i\},$ $M_5 = \{AID_i, SK_j, V_1, Tr_j\} \rightarrow$	
Get $SK, Tr_i,$ Verify $V_2,$ Compute $AIDnew_i,$ Update $AID_i = AIDnew_i.$		Get $SK, Tr_j,$ Verify $V_1,$ Compute $AIDnew_j,$ Update $AID_j = AIDnew_j.$

[DOI: 10.61186/jsdp.19.4.85]

جانبابائی و همکاران در [1] ادعا کردند طرح آنها که در بالا توصیف شد، یک طرح امن بوده و از کارآیی لازم نیز برخوردار است. ایشان امنیت طرح خود را با فرض مورد اطمینان‌بودن سرخوشه‌ها بررسی کردند. در بخش بعدی این مقاله نشان داده می‌شود که پروتکل AKA توصیف‌شده، برخی ضعف‌های امنیتی اساسی دارد که باعث می‌شود به‌طورکامل ناامن باشد.

۳- ارزیابی امنیتی پروتکل جانبابائی و همکاران

در این بخش برخی آسیب‌پذیری‌های امنیتی پروتکل جانبابائی و همکاران تشریح می‌شود. برای تحلیل و ارزیابی این پروتکل لازم است تا ابتدا توانمندی مهاجم و مدل حمله مشخص شود. مهاجمی که در این مقاله برای حمله به پروتکل ذکر شده در نظر گرفته شده است، از مدل ارائه شده در [5,7,9] پیروی می‌کند. دقیق‌تر اینکه در این مقاله مهاجم $\mathcal{A}$ یک نوع مهاجم فعال است که می‌تواند پیام‌های تبادل‌شده در یک کانال عمومی میان گره‌های حس‌گر و سرخوشه‌ها را شنود کند و یا تغییر دهد. برای مثال $\mathcal{A}$ می‌تواند پیام‌های نشست‌های قبلی پروتکل را ثبت و ضبط نماید و از آنها برای تحت تاثیر قراردادن نشست کنونی یا نشست‌های آینده استفاده کند. مهاجم $\mathcal{A}$ همچنین می‌تواند یکی از اعضای شرکت‌کننده در پروتکل (یعنی یک گره حس‌گر یا یک سرخوشه) نیز باشد. چنین مهاجمی، یک نهاد بدخواه از پروتکل در نظر گرفته می‌شود. علاوه‌بر این توضیحات مهاجم $\mathcal{A}$ همچنین می‌داند که چگونه پروتکل اجرا می‌شود. گفتنی است، اگر مهاجم به‌عنوان یک سرخوشه بدخواه در نظر گرفته شود، فرض مربوط به مورد اطمینان‌بودن آن که در پروتکل جانبابائی و همکاران در نظر گرفته شده بود، نقض می‌شود؛ ولی باید توجه داشت که این موضوع می‌تواند درعمل اتفاق بیافتد.

در ادامه این مقاله با درنظرگرفتن قابلیت‌هایی که برای مهاجم $\mathcal{A}$ شمرده شد، نشان داده می‌شود پروتکل جانبابائی و همکاران در برابر چنین مهاجمی دارای ضعف‌های اساسی است. برای مثال مهاجم $\mathcal{A}$ می‌تواند کلیدهای نشست گره‌های حس‌گر را به‌دست آورد یا پارامترهای محرمانه این گره‌ها را کشف کند. او همچنین می‌تواند یک هویت یک گره حس‌گر را موقع استفاده از این پروتکل جعل نماید. یکی از مهم‌ترین موفقیت‌های مهاجم در برخورد با پروتکل جانبابائی و همکاران این است که او می‌تواند یک گره حس‌گر را ردیابی و گم‌نامی آن را نقض کند.

۳-۱- کشف کلید نشست میان گره‌های حس‌گر

در این بخش نشان داده می‌شود که یک مهاجم می‌تواند کلیدهای نشست گره‌های حس‌گر را در مدل کلید نشست معلوم[1] به‌دست آورد. در مدل کلید نشست معلوم فرض می‌شود که مهاجم مقادیر برخی کلیدهای نشست قبلی یا آینده را می‌داند و سعی می‌کند تا کلید نشست کنونی یا برخی کلیدهای نشست دیگر را به‌دست آورد.

با توجه به پروتکل جانبابائی و همکاران فرض کنید که مهاجم $\mathcal{A}$ یک نشست قبلی میان دو گره حس‌گر را شنود کرده باشد و کلید این نشست را به‌دست آورده باشد (مدل کلید نشست معلوم). نشست مورد نظر را با $S^{(1)}$ و کلید نشست یادشده را با $SK^{(1)}$ در نظر بگیرید. مطابق توضیحات پروتکل مهاجم $\mathcal{A}$ می‌تواند به مقادیر تبادل‌شده طی نشست $S^{(1)}$ که شامل مقدارهای A و B هستند، دسترسی داشته باشد. این مقادیر به‌ترتیب با $A^{(1)}$ و $B^{(1)}$ نشان داده می‌شوند. با توجه به محاسبات پروتکل می‌توان معادلات زیر را نوشت:

$$A^{(1)} = N_i^{(1)} \oplus K_i,$$
$$B^{(1)} = N_j^{(1)} \oplus K_j,$$
$$SK^{(1)} = N_i^{(1)} \oplus N_j^{(1)} = A^{(1)} \oplus K_i \oplus B^{(1)} \oplus K_j$$
$$= A^{(1)} \oplus B^{(1)} \oplus K_i \oplus K_j.$$

بنابراین می‌توان مقدار $K_i \oplus K_j$ را به صورت زیر به‌دست آورد:

$$K_i \oplus K_j = SK^{(1)} \oplus A^{(1)} \oplus B^{(1)}.$$

یادآوری می‌شود که مهاجم $\mathcal{A}$، مقدار $SK^{(1)}$ را در مدل کلید نششت معلوم می‌داند و مقادیر $A^{(1)}$ و $B^{(1)}$ را نیز با مشاهده پیام‌های $M_1^{(1)}$ و $M_3^{(1)}$ به‌دست آورده است. حال مهاجم با داشتن مقدار $K_i \oplus K_j$، می‌تواند کلید محرمانه هر نشست دیگر میان دو گره حس‌گر SN_i و SN_j را به

[1] Known Session Key Model

[DOI: 10.61186/jsdp.19.4.85]

دست آورد. فرض کنید، هدف مهاجم به‌دست‌آوردن کلید نشست $SK^{(t)}$، کلید محرمانه مربوط به نشست $S^{(t)}$ باشد. او می‌تواند این کار را به‌صورت زیر انجام دهد:

- مهاجم $\mathcal{A}$ مقادیر $A^{(t)}$ و $B^{(t)}$ را به‌ترتیب از پیام‌های $M_1^{(t)}$ و $M_3^{(t)}$ به‌دست می‌آورد.
- مهاجم $\mathcal{A}$، کلید نشست $SK^{(t)}$ را به‌صورت زیر محاسبه می‌کند:

$$SK^{(t)} = A^{(t)} \oplus B^{(t)} \oplus K_i \oplus K_j.$$

۳-۲- کشف پارامترهای محرمانه گره‌های حس‌گر

در استفاده از پروتکل جانبابائی و همکاران، گره‌های حس‌گر یک سری پارامترهای محرمانه دارند که نباید برای مهاجم یا حتی یک گره حس‌گر دیگر مشخص باشد. در این بخش نشان داده می‌شود که اگر مهاجم در نقش یک گره حس‌گر بدخواه در این پروتکل شرکت کند، می‌تواند پارامترهای محرمانه گره‌های حس‌گر را به‌دست آورد. برای مثال، فرض کنید گره حس‌گر SN_i یک گره حس‌گر بدخواه باشد که می‌خواهد اطلاعات محرمانه گره حس‌گر SN_j را به‌دست آورد. آن یک ارتباط با SN_j برقرار کرده و کلید نشست میان خود و SN_j را به‌صورت SK محاسبه می‌کند. با توجه به پروتکل، $SK = N_i \oplus N_j$ است و N_i یک پارامتر محرمانه است که در اختیار SN_i قرار دارد. حال گره حس‌گر SN_i می‌تواند پارامتر محرمانه مربوط به گره حس‌گر SN_j یعنی N_j را به‌صورت زیر حساب کند:

$$N_j = SK \oplus N_i.$$

درنهایت، آن می‌تواند مقدار B را از پیام M_3 به‌دست آورد و کلید محرمانه مشترک میان گره حس‌گر SN_j و سرخوشه، یعنی K_j را به‌صورت زیر حساب کند:

$$K_j = B \oplus N_j.$$

فرض کنید مهاجم $\mathcal{A}$ به ترتیبی که توضیح داده شد، توانسته باشد کلیدهای محرمانه گره‌های حس‌گر SN_j و SN_l، یعنی به‌ترتیب K_j و K_l را به‌دست آورده باشد. حال او می‌تواند از پیام‌های تبادل شده در یک نشست میان گره‌های حس‌گر SN_j و SN_l استفاده کرده و کلید نشست میان آنها، در هر نشستی را، به‌دست آورد. دقیق‌تر اینکه مهاجم $\mathcal{A}$ از پارامترهای محرمانه گره‌های حس‌گر SN_j و SN_l، یعنی به K_j و K_l و نیز A و B استفاده کرده و کلید نشست مورد نظر را به‌صورت زیر حساب می‌کند:

$$SK = N_j \oplus N_l = A \oplus B \oplus K_j \oplus K_l.$$

۳-۳- جعل گره حس‌گر

در حمله جعل گره حس‌گر ، مهاجم $\mathcal{A}$ تلاش می‌کند تا یک گره حس‌گر ، برای مثال گره SN_i را جعل کند. فرض کنید مهاجم با توجه به توضیحات بخش ۳-۲، توانسته باشد K_i، کلید محرمانه میان گره SN_i و سرخوشه را به‌دست آورد. او می‌تواند Tr_i را از پیام M_4 در نشست جاری، یعنی $S^{(t)}$ شنود کند. او همچنین آخرین مقدار به‌روزشده AID_i (هویت مستعار SN_i) را از روی اطلاعات عمومی گره حس‌گر SN_i به‌دست می‌آورد. حال مهاجم $\mathcal{A}$، می‌تواند گره حس‌گر SN_i را در نشست بعدی، یعنی $S^{(t+1)}$ جعل کند. برای این کار مهاجم به‌صورت زیر عمل می‌کند:

- مهاجم $\mathcal{A}$ یک تک‌شمار تصادفی مانند N'_i تولید می‌کند.
- مهاجم $\mathcal{A}$ مقادیر زیر را حساب می‌کند.

$A' = N'_i \oplus K_i,$
$V'_i = h(AID_i \parallel AID_j \parallel N'_i \parallel Tr_i)$

- مهاجم پیام

$M'_1 = \{AID_i, AID_j, Tr_i, V'_i, A'\}$

را برای سرخوشه ارسال می‌کند.

- سرخوشه بعد از بررسی Tr_i، N'_i را به‌دست می‌آورد؛ سپس آن V'_i را بررسی و تأیید می‌کند. به عبارت دیگر سرخوشه، مهاجم $\mathcal{A}$ را به جای گره حس‌گر SN_i می‌پذیرد.

۳-۴- حمله مردی در میانه

مهاجم $\mathcal{A}$ که قبلاً فرض شده است یک مهاجم فعال است، می‌تواند در پروتکل جانبابائی و همکاران، پیام‌های M_4 و M_5 را طوری تغییر دهد که گره‌های حس‌گر شرکت‌کننده در این پروتکل پیام‌های تغییر یافته را تأیید کرده و بپذیرند. به این ترتیب مهاجم می‌تواند یک کلید جعلی مانند SK' تولید کرده و آن را به‌عنوان کلید محرمانه نشست میان دو گره SN_i و SN_j قرار دهد و نشست آنها را کنترل کند. توجه شود که در این حمله فرض می‌شود، مهاجم پارامترهای محرمانه گره‌ها را با توجه به حمله بخش ۳-۲ به‌دست آورده است. با این فرض

[DOI: 10.61186/jsdp.19.4.85]

حمله مردی در میانه روی پروتکل جانبابائی و همکاران به‌صورت زیر انجام می‌شود:

- مهاجم $\mathcal{A}$ یک مقدار تصادفی مانند SK' تولید می‌کند و مقادیر SK'_i و SK'_j را با جایگزین کردن SK با SK' به‌ترتیب در محاسبات مربوط به SK_i و SK_j به‌دست می‌آورد.
- مهاجم $\mathcal{A}$ مقادیر جدید V'_1 و V'_2 را به ترتیب با جایگزین‌کردن SK_j و SK_i با SK'_j و SK'_i در محاسبات مربوط به V_1 و V_2 به‌دست می‌آورد.
- برای تغییر پیام‌های M_4 و M_5 که از طرف سرخوشه تولید و به ترتیب برای گره‌های SN_i و SN_j ارسال می‌شود، مهاجم مقادیر SK_i، SK_j، V_2 و V_1 را به ترتیب با مقادیر SK'_i، SK'_j، V'_2 و V'_1 جایگزین می‌کند.
- هر دو گره حس‌گر SN_i و SN_j به ترتیب پیام‌های M_4 و M_5 تغییر یافته توسط مهاجم $\mathcal{A}$ را تأیید کرده و تصور می‌کنند که به‌کمک سرخوشه و با استفاده از کلید نشست SK' با یکدیگر ارتباط امن دارند، درحالی‌که نشست و ارتباط آنها توسط مهاجم $\mathcal{A}$ کنترل می‌شود.

۳-۵- جعل سرخوشه

مشابه حمله جعل گره حس‌گر ، در حمله جعل سرخوشه، مهاجم تلاش می‌کند تا یک سرخوشه را جعل کند. در این بخش نشان داده می‌شود که پروتکل جانبابائی و همکاران در برابر حمله جعل سرخوشه نیز آسیب‌پذیر است. برای انجام این حمله مهاجم $\mathcal{A}$ از روش جعل M_4 و M_5 در حمله مردی در میانه که در بخش ۳-۴ تشریح شد استفاده کرده و یک سرخوشه را به‌صورت زیر جعل می‌کند:

- بعد از دریافت پیام M_1 از گره حس‌گر SN_i، مهاجم $\mathcal{A}$ پیام M_2 شامل درخواست احراز اصالت را برای گره حس‌گر SN_j ارسال می‌کند.
- بعد از اینکه مهاجم $\mathcal{A}$ پیام M_3 را از گره حس‌گر SN_j دریافت کرد، M_4 و M_5 را با توجه به توضیحات بخش ۳-۴ محاسبه کرده و به ترتیب برای گره‌های حس‌گر SN_i و SN_j ارسال می‌کند.
- هر دو گره حس‌گر SN_i و SN_j، به ترتیب پیام‌های M_4 و M_5 جعلی را تأیید و مهاجم $\mathcal{A}$ را به عنوان سرخوشه مجاز احراز اصالت می‌کنند.

۳-۶- ردیابی گره حس‌گر

در این بخش نشان داده می‌شود که پروتکل جانبابائی و همکاران در برابر حمله ردیابی گره حس‌گر آسیب‌پذیر است و نمی‌تواند گم‌نامی گره حس‌گر را تأمین کند.

یک مهاجم برای ردیابی یک گره حس‌گر در این پروتکل، ابتدا با توجه به توضیحات بخش ۳-۲، کلیدهای محرمانه گره‌های حس‌گر را به‌دست می‌آورد. مهاجم این کلیدها را در پایگاه داده خود ذخیره می‌کند. حال فرض کنید مهاجم کلید K_i را که مربوط به SN_i است، در اختیار داشته باشد. هنگامی که این مهاجم یک نشست میان گره‌های حس‌گر ، برای مثال دو گره SN_i و SN_j را مشاهده کند، او می‌تواند پیام‌های تبادل شده، یعنی M_1، M_2، M_3 و M_4 را شنود کند؛ سپس او مقدار:

$$SK_i = (Tr_i \parallel SK) \oplus h(K_i \parallel AID_j).$$

را از پیام M_4 به‌دست می‌آورد. با فرض اینکه مهاجم $\mathcal{A}$، مقدار K_i را دارد، او می‌تواند محاسبه زیر را انجام دهد.

$$(Tr_i \parallel SK) = SK_i \oplus h(K_i \parallel AID_j).$$

سپس مهاجم بررسی می‌کند که آیا $A \oplus SK = B$ درست است یا نه. اگر این عبارت صحیح باشد آنگاه SN_i گرهی است که نشست را شروع کرده است؛ بنابراین مهاجم $\mathcal{A}$ می‌تواند گره حس‌گری که کلید او K_i است را ردیابی کند. روشن است که او حتی می‌تواند گره حس‌گر SN_i را برای حالتی که شروع‌کننده نشست نیز نباشد، ردیابی کند.

۴- پیشنهاد جهت بهبود پروتکل جانبابائی و همکاران

در بخش قبل، برخی ضعف‌های امنیتی پروتکل جانبابائی و همکاران، مانند کشف کلید نشست، کشف پارامترهای محرمانه گره‌های حس‌گر، جعل گره‌های حس‌گر و سرخوشه، حمله مردی در میانه روی آن و ردیابی گره‌های حس‌گر تشریح، در دو حمله ابتدایی، از یک ضعف خاص پروتکل برای یافتن کلیدهای نشست و پارامترهای محرمانه استفاده شد و در حملات دیگر از پارامترهای محرمانه گره‌های حس‌گر که بهدست آمده بود، استفاده شد تا اهداف مورد نظر حاصل شود، بنابراین شروع ضعف‌های پروتکل جانبابائی و همکاران مربوط به چگونگی کشف پارامترهای محرمانه گره‌های حس‌گر است. در این پروتکل یک مهاجم می‌تواند هر کلید نشست را در مدل

[DOI: 10.61186/jsdp.19.4.85]

حمله کلید نشست معلوم محاسبه کند، به این خاطر که کلیدهای نشست به‌صورت زیر محاسبه می‌شوند:

$$SK = N_i \oplus N_j.$$

به‌طور مشابه، یک مهاجم می‌تواند به خاطر نوع محاسبه SK اطلاعات محرمانه گره‌های حس‌گر را به‌دست آورد. یک تلاش اولیه برای بهبود امنیت پروتکل جانبابائی و همکاران، استفاده از یک تابع چکیده‌ساز برای محاسبه کلید نشست یعنی SK است. درواقع اگر کلید نشست در این پروتکل به‌صورت:

$$SK = h(N_i \oplus N_j)$$

محاسبه شود، آنگاه مهاجم $\mathcal{A}$ نمی‌تواند به اهداف خود در حملات یادشده دست یابد.

توجه شود، بهبود ارائه‌شده در این بخش مربوط به امن کردن پروتکل جانبابائی و همکاران در برابر حملات تشریح شده در این مقاله است. ممکن است، برخلاف این بهبود حملات دیگری (غیر از حملات یادشده) روی این پروتکل قابل بررسی باشد. برای مثال، اگر مهاجم جلوی ارسال پیام M_3 یا M_4 را قطع کند، آنگاه برای گره‌های حس‌گر (به‌ترتیب) SN_i یا SN_j یک ناهمزمانی ایجاده می‌شود. برای حل این مساله، طراحان پیشنهاد کرده‌اند تا از V_i و V_j به‌صورت:

$$V_i = h\big(AID_i \parallel AID_j \parallel N_i \parallel Tr_i\big),$$

$$V_j = h\big(AID_j \parallel N_j \parallel Tr_j\big)$$

استفاده شود؛ اما این روش نمی‌تواند مساله ناهم‌زمانی یادشده را حل کند. به این خاطر که گره‌های SN_i و SN_j به‌ترتیب AID_i و AID_j را به‌روز نمی‌کنند، درحالی‌که سرخوشه این کار را انجام می‌دهد. برای حل این مساله لازم است تا سرخوشه مقادیر قدیمی را ثبت کند.

۵- نتیجه‌گیری

در این مقاله، امنیت پروتکل احراز اصالت و توافق کلید سبک‌وزن جانبابائی و همکاران مورد مطالعه و بررسی قرار گرفت و نشان داده شد که در برابر برخی حملات ناامن است. به این ترتیب که اگر یک مهاجم اطلاعات یک نشست از پروتکل میان دو گره حس‌گر را داشته باشد، می‌تواند هر کلید نشست دیگر میان این گره‌ها را حساب کند. همچنین تشریح شد که یک مهاجم در نقش یک گره حس‌گر بدخواه می‌تواند اطلاعات محرمانه گره‌های حس‌گر دیگر را به‌دست آورد. با استفاده از این ضعف‌ها می‌توان برای پروتکل حملات جعل گره حس‌گر ، جعل سرخوشه، مردی در میانه و ردیابی ارائه کرد. موفق‌بودن حمله ردیابی روی پروتکل مورد نظر نشان می‌دهد که بر خلاف ادعای طراحان، این پروتکل نمی‌تواند گمنامی گره حس‌گر را تأمین کند.

در این مقاله توضیح داده شد که ضعف اصلی پروتکل مربوط به نحوه محاسبه کلید نشست است که تنها با استفاده از عمل‌گر xor و بدون کاربرد تابع چکیده‌ساز محاسبه می‌شود. درنهایت یک پیشنهاد ابتدایی برای بهبود پروتکل جانبابائی و همکاران در برابر حملات یادشده بیان شد. این پیشنهاد در حالت کلی با تمرکز بر رفع ضعف‌های امنیتی مشاهده‌شده ارائه شد؛ اما این امر به معنی عدم وجود مسائل امنیتی دیگر در پروتکل، مانند نقض محرمانگی پیشرو/ پسرو، نیست؛ بنابراین اصلاح و بهبود پروتکل جانبابائی و همکاران به‌طوری‌که سایر ویژگی‌های امنیتی را نیز فراهم کند، می‌تواند در ادامه کار این مقاله مد نظر قرار گیرد. علاوه‌براین، به دلیل تمرکز روی مباحث امنیتی، بحث بهینه‌سازی ساختار پیام‌های مبادله‌شده از نظر کاهش سربار محاسباتی/ مخابراتی در این مقاله مد نظر قرار نگرفته است؛ لذا یک مسیر پژوهشی برای کار آینده می‌تواند اصلاح ساختار پیام‌های مبادله‌شده با هدف کاهش هزینه محاسباتی و مخابراتی باشد.

۶- مراجع 6- References

[۱] جانبابائی شادی، قرائی حسین، محمد زاده ناصر. ارائه طرح احراز اصالت سبک با قابلیت گمنامی و اعتماد در اینترنت اشیا. *پردازش علائم و داده‌ها*. ۱۳۹۷; ۱۵ (۴) :۱۱۱-۱۲۲

[1] Sh. Janbabaei, H. Gharaee, and N. Mohammadzadeh, "The lightweight authentication scheme with capabilities of anonymity and trust in internet of things (IoT)," SIGNAL AND DATA PROCESSING, vol. 15, no. 4 (38), 2019, (In Persian).

[2] M. A. Ferrag, L. A. Maglaras, H. Janicke, J. Jiang, and L. Shu, "Authentication protocols for internet of things: a comprehensive survey," Security and Communication Networks, 2017.

[3] J. Andress, The basics of information security: understanding the fundamentals of InfoSec in theory and practice. Syngress, 2014.

[4] M. Turkanovi´c, B. Brumen, and M. H¨olbl, "A novel user authentication and key agreement scheme for heterogeneous ad hoc wireless sensor networks, based on the internet of things notion," Ad Hoc Networks, vol. 20, pp. 96–112, 2014.

[DOI: 10.61186/jsdp.19.4.85]

[5] M. S. Farash, M. Turkanovi´c, S. Kumari, and M. H¨olbl, “An efficient user authentication and key agreement scheme for heterogeneous wireless sensor network tailored for the internet of things environment,” Ad Hoc Networks, vol. 36, pp. 152–176, 2016.

[6] R. Amin and G. Biswas, “A secure light weight scheme for user authentication and key agreement in multigateway based wireless sensor networks,” Ad Hoc Networks, vol. 36, pp. 58–80, 2016.

[7] R. Amin, S. H. Islam, G. Biswas, M. K. Khan, L. Leng, and N. Kumar, “Design of an anonymity-preserving three-factor authenticated key exchange protocol for wireless sensor networks,” Computer Networks, vol. 101, pp. 42–62, 2016.

[8]Y. Lu, L. Li, H. Peng, and Y. Yang, “An energy efficient mutual authentication and key agreement scheme preserving anonymity for wireless sensor networks,” Sensors, vol. 16, no. 6, p. 837, 2016.

[9]Q. Jiang, S. Zeadally, J. Ma, and D. He, “Lightweight three-factor authentication and key agreement protocol for internet-integrated wireless sensor networks,” IEEE Access, vol. 5, pp. 3376–3392, 2017.

[10]R. Ali, A. K. Pal, S. Kumari, M. Karuppiah, and M. Conti, “A secure user authentication and keyagreement scheme using wireless sensor networks for agriculture monitoring,” Future Generation Computer Systems, vol. 84, pp. 200–215, 2018.

[11]Y. Lu, G. Xu, L. Li, and Y. Yang, “Anonymous threefactor authenticated key agreement for wireless sensor networks,” Wireless Networks, vol. 25, no. 4, pp. 1461–1475, 2019.

[12]S. Athmani, A. Bilami, and D. E. Boubiche, “Edak: An efficient dynamic authentication and key management mechanism for heterogeneous wsns,” Future Generation Computer Systems, vol. 92, pp. 789–799, 2019.

[13]M. Nikravan and A. Reza, “A multi-factor user authentication and key agreement protocol based on bilinear pairing for the internet of things,” Wireless Personal Communications, vol. 111, no. 1, pp. 463–494, 2020.

[14]Y. Yu, L. Hu, and J. Chu, “A secure authentication and key agreement scheme for iot-based cloud computing environment,” Symmetry, vol. 12, no. 1, p. 150, 2020.

جواد علیزاده، دکترای خود را در رشته رمزنگاری از دانشگاه جامع امام حسین (ع)، در سال ۱۳۹۵ تحت راهنمایی آقای دکتر محمدرضا عارف و مشاوره آقای دکتر نصور باقری دریافت کردند. در حال حاضر ایشان استادیار دانشکده فناوری اطلاعات و ارتباطات دانشگاه جامع امام حسین (ع) هستند. علاقه‌مندی‌های پژوهشی وی شامل طراحی و تحلیل پروتکل‌های رمزنگاری و الگوریتم-های رمزنگاری متقارن است.

نصور باقری، کارشناسی ارشد و دکترای خود را در رشته مهندسی الکترونیک از دانشگاه علم و صنعت ایران، به‌ترتیب در سال‌های ۱۳۸۱ و ۱۳۸۹ دریافت کرده است. در حال حاضر ایشان دانشیار دانشکده مهندسی برق دانشگاه تربیت دبیر شهید رجایی هستند. وی مؤلف بیش از صد مقاله در حوزه امنیت اطلاعات، پروتکل‌های رمز و رمزشناسی هستند. علاقه‌مندی‌های پژوهشی ایشان شامل طراحی و تحلیل پروتکل‌های رمزنگاری مناسب برای IoT یا RFID و همچنین طراحی و تحلیل رمزهای متقارن است.

[DOI: 10.61186/jsdp.19.4.85]