Modeling and Analysis of Remote Attestation Components for Android Applications in IoT

Vasily A. Desnitsky[1], Igor V. Kotenko[2]
St. Petersburg Institute for Informatics and Automation of the Russian Academy of Sciences (SPIIRAS)
St. Petersburg, Russia
[1]desnitsky@comsec.spb.ru, [2]ivkote@comsec.spb.ru

***Abstract*— In the paper the mechanisms for protection of Android applications from the threats of integrity violation of the software and of critical data on the base of remote attestation principles are modeled. The attestation is based on checking the control flow and checksums of the specified data structures. In particular program markers are placed at the source code level of a specific application. After that a regular expression is formed, which determines the correct control flow of the protected program, and then a graph to verify the attested data is formed on the attestation side. Analytical and experimental evaluations of the implemented protection components and the protocol of their interaction taking into account limitations on the computing and communication resources of the target device are performed.**

Keywords*— *remote attestation; integrity; mobile application; Internet of Things (IoT)

I. Introduction

A problem of protecting IoT software from unauthorized modification threats is becoming increasingly important and is caused by the susceptibility of software platforms of mobile and embedded devices such as Android, Raspberry Pi and others to threats of integrity and authenticity violation of the code and data used.

In general application of algorithms controlling immutability of the software, which are built directly into the program they protect, can increase the protection level. However the local nature of the protection and limiting its persistence as well as situating the program in an environment being non-trusted and uncontrolled by the software developer or the owner of the digital rights leads to the fact that such protection mechanisms can be neutralized by an intruder if there are sufficient tools and resources.

The remote validation mechanism, investigated in the paper, is based on the use of a client-server approach to protection and allows increasing security of the software under resource limitations of the mobile platform as well as limitations of the communication channel bandwidth.

The work is performed in SPIIRAS and partially supported by RFBR (projects #16-29-09482 and 18-07-01488), the Council for Grants of the President of Russia (project # MK-5848.2018.9) and the budget theme (project #AAAA-A16-116033110102-5).

In the paper modeling and analysis of particular protection algorithms are performed within the framework of an integrated approach to the implementation of software protection components, implementing remote attestation with the use of Android platform.

The distinguishing features of the results achieved in the paper include, in particular, experimental data obtained during the modeling of protection components under mobile operating system limitations.

The paper is organized as follows. Section 2 provides an overview of existing works in the subject field. Section 3 reveals features of the approach to remote attestation of mobile applications. Section 4 describes results of modeling of specific remote attestation based algorithms that implement. Section 5 presents results of the experimental studies, whereas Section 6 concludes the paper.

II. Related work

Brasser et al. [1] and C. Preschern et al. [2] consider remote attestation as means of protection against malicious software intrusion attacks on embedded devices [3]. The features and methods that allow implementation of the attestation with minimal additional costs are demonstrated. At that C. Preschern et al. [2] propose adaptation of software methods for remote attestation to solve tasks of protecting critical systems with minimal forced revision of established procedures for safety properties certification.

J. Ho et al. [4] show that in sensor networks the remote certification is used for detection of self-propagating network worms by sequentially infecting nodes using traffic detection methods.

Srinivasan et al. [5] investigate remote software-based attestation to ensure the integrity of the operating system kernel and user applications. In particular the authors propose a technique that allows determining whether the already certified application was substituted by an intruder or not.

M. Santra, et al. [6] propose the use of remote attestation and the three-phase protocol constructed on it, using a SELinux module for providing secure interaction in distributed information systems. The paper also substantiates

978-1-5386-5824-6/18/$31.00 ©2018 IEEE

effectiveness of the proposed approach, using methods of formal analysis and ProVerif verifier.

T. AbuHmed, et al. [7] propose software techniques for remote attestation of wireless sensor networks against tampering attacks into their work. These techniques are not based on the use of the accuracy factor of the measured runtime execution, thereby improving previously proposed integrity monitoring methods in wireless sensor networks [8].

D. Fu and X. Peng [9] analyze security of the mechanisms of one- and multi-hop attestation in wireless sensor networks.

Tan et al. [10] propose a multi-level remote attestation protocol to monitor integrity of IoT-systems, taking into account their inherent computational limitations and device's power limitations.

In [11], [12] the authors propose a reference architecture and partial models for mechanisms of IoT remote attestation, using cloud solutions to improve the targets of the remote attestation process.

K. Ramachandran and H. Lutfiyya [13] prove the importance of remote attestation of software updates, using cloud computing and procedures for verifying its correctness [14].

Y. Zhang et al. [15] extend remote attestation application to ensuring the confidentiality by a modified Extended Hash Algorithm [16]. It allows increasing level of the confidentiality with comparable performance characteristics during the execution.

T. Syed et al. [17] propose effective solutions for increasing scalability of mechanisms for remote attestation of device sets by using Big Data technology, including using multiprocessor systems [18], property based authentication mechanisms [19] and characteristics of these properties [20].

Increasing the efficiency of the server part of the authentication mechanism with a large number of instances of attested programs is also achieved by reorganizing and reducing the chain of trust used in the attestation process [21].

H. Li et al. [22] propose models of remote attestation based on the paradigm of attack graphs to tackle tasks of monitoring and attestation of software components [23].

III. Approach to Remote Attestation of Mobile Applications

Remote attestation of a mobile application includes software local and remote components that are located within a non-trusted and trusted environment, respectively, as well as a secure protocol for their network interaction.

The interaction between the components is based on roles of a client - the attested entity, and the server - the attesting one. The protocol assumes implementation of the protection functions of the protocol itself from possible interception and modification of packets at the transport level. The payload of the protocol includes program identifiers and numeric values that characterize the current state of elements of the program code and critical data of the application.

Specific algorithms used within the framework of the protection mechanism on the base of remote attestation principles assume, first, introduction of specific constructions emplaced into the objective code at the stage of forming the syntactic tree of the application and, second, isolation of basic blocks and particular instructions in the code.

Security constructions do not directly perform any code integrity checks and data locally, but send their snapshots to the side of the trusted server. This fact greatly complicates successful intruder's modification of the application, being not subsequently detected on the server side.

A typical scenario for applying remote attestation to the protection of mobile devices involves remote control by a mobile application store or content provider over multiple instances of client applications. In case of a violation it warns on the violation on a specific device and stops its further maintenance until the detected violation is rectified.

IV. Protection Algorithms

A control flow checking algorithm is based on a control flow graph of the protected program, built statically. The graph is used in dynamics for remote control of the correctness of the process of its execution. This algorithm allows ensuring the correctness of the execution of a sequence of commands, including branching structures, loops, handling of exceptional situations, etc.

The control flow checking algorithm includes two stages, namely static and dynamic. Statically one prepares and embeds the attesting module constructions in the program code. The initiating construction establishes a connection to the remote attesting module via HTTP sockets.

Program markers, which are operations *send (A)* of sending a specific identifier *A* to the server side, are situated in the program code on the boundaries of the base blocks. The attesting module function on the client side contains sending a sequence of identifiers of program markers during their passage in the execution process (dynamically).

On the server side of the connection one constructs a regular expression, which determines the correct chains of operation of program markers within the static stage. The regular expression is used to build a transition graph, which nodes determine program markers and arcs denote permissible transitions between them.

At the dynamic stage when receiving the identifiers of program markers from the client side the process of traversing the graph is performed and its correctness is checked in accordance with the structure of the graph.

As an example Fig. 1 schematically shows a fragment of the code of the protected program with built-in functions for sending program markers.

The regular expression in the infix form constructed for the given fragment on the server side is *A(B(C|D)E)*F*. The graph of transitions with the final vertex F corresponding to this regular expression is shown in Fig. 2.

```
send("A");
while(t<=250){
    send("B");
    if(checkLicenseCode()){
        setDependMode();
        send("C");
    } else {
        // business code
        send("D");
    }
    runEmerging();
    t++;
    send("E");
}
send("F");
```

Fig. 1. Code fragment of the protected program

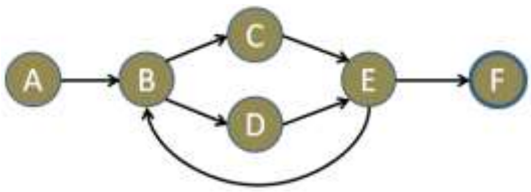

Fig. 2. Transition graph for the remote attestation algorithm

The checksum algorithm requires the existence of invariant data structures that are critically important in relation to the task of ensuring the integrity of the protected code and data. Cryptographic algorithm MD5 is used as the basis.

The sending to the server side of the connection and checking the received token is done by using the *send (md5(criticulStructure))* and *verify(getNextToken())* functions, respectively.

V. Experiments and Discussion

The evaluation of the solutions constructed in the work is done by defining and analyzing values of a number of indicators, namely, the indicators of efficiency, reliability and resource consumption [24].

The efficiency indicator is due to the hardware limitations of the Android platform and the limited bandwidth of the communication channel, which affect the stability and continuity of the application, as well as the usability of the end user.

The efficiency is calculated on a test scenario by using a system function *System.currentTimeMillis()* as an average value of the time delays that occur as a result of executing instructions for sending program markers and checksums.

The results of the conducted experiments showed that when the ratio of the number of built-in instructions of the evaluating module to the number of instructions of the target program not exceeding 20%, the average delay value did not exceed the established allowable limit of 200 ms.

Calculation of the resource consumption indicators is performed by using the jmap and jstat utilities. These ones allow estimating the increase in the consumption of the consumed RAM on the client side after adding the attestation functions to the code. Based on a series of measurements made on the test application, it was determined that the increase in the memory consumption did not exceed 21% in comparison with the unprotected version of the software application.

To evaluate the reliability indicator of the proposed security solution, fuzzy testing of the protected application was performed on pre-generated tests, including random and boundary values of the input data. Testing a series of 250 samples of input data revealed no false positive and false negative errors. Therefore it confirms the correctness of the proposed approach to the remote attestation and the operational capability of its software implementation.

Applicability of the proposed approach to protection of the integrity of Android applications is due to the achievable level of deployment automation of the proposed security solutions as well, including the choice of location and placement of attesting instructions in the code. This makes it possible to solve the problems of efficient selection and adaptation of existing software tools for processing Java code, both at the source code level and directly by using bytecode analysis tools.

VI. Conclusion

An approach to remote attestation of mobile applications, using control flow checking and checksum checking algorithms has been investigated.

A software implementation of the algorithms was performed by using Android platform as an example to serve as a basis for obtaining experimental characteristics of these algorithms.

As a direction for future research it is planned, first, to develop techniques to analyze security of mobile applications and, second, to increase their security, including at the source code level and object code one.

References

[1]. F. Brasser, K. B. Rasmussen, A. R. Sadeghi and G. Tsudik, "Remote attestation for low-end embedded devices: The prover's perspective," 2016 53nd ACM/EDAC/IEEE Design Automation Conference (DAC), Austin, TX, 2016, pp. 1-6.

[2]. C. Preschern, A.J. Hörmer, N. Kajtazovic and C. Kreiner, "Software-Based Remote Attestation for Safety-Critical Systems," 2013 IEEE Sixth International Conference on Software Testing, Verification and Validation Workshops, Luxembourg, 2013, pp. 8-12.

[3]. V. Desnitsky, A. Chechulin, I. Kotenko, D. Levshun and M. Kolomeec, "Application of a Technique for Secure Embedded Device Design Based on Combining Security Components for Creation of a Perimeter Protection System," 2016 24th Euromicro International Conference on Parallel, Distributed, and Network-Based Processing (PDP), Heraklion, 2016, pp. 609-616. DOI: 10.1109/PDP.2016.99.

[4]. J.W. Ho and M. Wright, "Distributed Detection of Sensor Worms Using Sequential Analysis and Remote Software Attestations," in IEEE Access, vol. 5, pp. 680-695, 2017.

[5]. R. Srinivasan, P. Dasgupta and T. Gohad, "Software Based Remote Attestation for OS Kernel and User Applications," 2011 IEEE Third International Conference on Privacy, Security, Risk and Trust and 2011 IEEE Third International Conference on Social Computing, Boston, MA, 2011, pp. 1048-1055.

[6]. M. Santra, S. K. Peddoju, A. K. Bhattacharjee and A. Khan, "Design and Analysis of a Modified Remote Attestation Protocol," 2017 IEEE Trustcom/BigDataSE/ICESS, Sydney, NSW, 2017, pp. 578-585.

[7]. T. AbuHmed, N. Nyamaa and D. Nyang, "Software-Based Remote Code Attestation in Wireless Sensor Network," GLOBECOM 2009. 2009 IEEE Global Telecommunications Conference, Honolulu, HI, 2009, pp. 1-8.

[8]. K. Xiangying and C. Yanhui, "Dynamic Remote Attestation Based on Concerns," 2015 8th International Symposium on Computational Intelligence and Design (ISCID), Hangzhou, 2015, pp. 76-80.

[9]. D. Fu and X. Peng, "TPM-based remote attestation for Wireless Sensor Networks," in Tsinghua Science and Technology, vol. 21, no. 3, pp. 312-321.

[10]. H. Tan, G. Tsudik and S. Jha, "MTRA: Multiple-tier remote attestation in IoT networks," 2017 IEEE Conference on Communications and Network Security (CNS), Las Vegas, NV, 2017, pp. 1-9.

[11]. Houbing Song; Glenn A. Fink; Sabina Jeschke, "Secure Registration and Remote Attestation of IoT Devices Joining the Cloud: The Stack4Things Case of Study," in Security and Privacy in Cyber-Physical Systems:Foundations, Principles, and Applications, 1, Wiley-IEEE Press, 2017, pp.472.

[12]. S. Azadiabad, H. Pedram and M. R. Abbasy, "Scalable protocol for remote integrity attestation of cloud based distributed services," 2014 IEEE 8th International Conference on Application of Information and Communication Technologies (AICT), Astana, 2014, pp. 1-5.

[13]. K. Ramachandran and H. Lutfiyya, "A remote attestation infrastructure for verifying the application of software updates," 2017 IFIP/IEEE Symposium on Integrated Network and Service Management (IM), Lisbon, 2017, pp. 317-325.

[14]. V. Desnitsky, I. Kotenko, "Expert Knowledge based Design and Verification of Secure Systems with Embedded Devices," 4rd IFIP International Workshop on Security and Cognitive Informatics for Homeland Defense (SeCIHD 2014). Fribourg. Switzerland. Lecture Notes in Computer Science (LNCS). Vol. 8708. Springer-Verlag, 2014, pp.194-210.

[15]. Y. Zhang, L. Wang, Y. You and L. Yi, "A Remote-Attestation-Based Extended Hash Algorithm for Privacy Protection," 2017 International Conference on Computer Network, Electronic and Automation (ICCNEA), Xi'an, 2017, pp. 254-257.

[16]. K. Xiangying and C. Yanhui, "Left full binary hash tree for remote attestation," 2017 IEEE 2nd International Conference on Signal and Image Processing (ICSIP), Singapore, 2017, pp. 385-390.

[17]. T. A. Syed, S. Jan, S. Musa and J. Ali, "Providing efficient, scalable and privacy preserved verification mechanism in remote attestation," 2016 International Conference on Information and Communication Technology (ICICTM), Kuala Lumpur, 2016, pp. 236-245.

[18]. M. Kiperberg, A. Resh and N. J. Zaidenberg, "Remote Attestation of Software and Execution-Environment in Modern Machines," 2015 IEEE 2nd International Conference on Cyber Security and Cloud Computing, New York, NY, 2015, pp. 335-341.

[19]. Y. Liang, K. E. Guo and J. Li, "The remote attestation design based on the identity and attribute certificates," 2014 11th International Computer Conference on Wavelet Actiev Media Technology and Information Processing(ICCWAMTIP), Chengdu, 2014, pp. 325-330.

[20]. A. Francillon, Q. Nguyen, K.B. Rasmussen and G. Tsudik, "A minimalist approach to Remote Attestation," 2014 Design, Automation & Test in Europe Conference & Exhibition (DATE), Dresden, 2014, pp. 1-6.

[21]. W. Luo, W. Liu, Y. Luo, A. Ruan, Q. Shen and Z. Wu, "Partial Attestation: Towards Cost-Effective and Privacy-Preserving Remote Attestations," 2016 IEEE Trustcom/BigDataSE/ISPA, Tianjin, 2016, pp. 152-159.

[22]. H. Li and S. Wang, "An Efficient and Flexible Dynamic Remote Attestation Method," 2014 Ninth International Conference on Broadband and Wireless Computing, Communication and Applications, Guangdong, 2014, pp. 239-246.

[23]. C. Meng, Y. He and Q. Zhang, "Remote Attestation for Custom-built Software," 2009 International Conference on Networks Security, Wireless Communications and Trusted Computing, Wuhan, Hubei, 2009, pp. 374-377.

[24]. V. Desnitsky, I. Kotenko, "Modeling and Analysis of IoT Energy Resource Exhaustion Attacks," Intelligent Distributed Computing XI. Studies in Computational Intelligence. Springer-Verlag. Vol.737. Proceedings of 11th International Symposium on Intelligent Distributed Computing (IDC'2017). Belgrade. Serbia. Springer-Verlag, 2017, pp. 263-270.