IJMTST is UGC Approved Journal with Journal Id: 43137

International Journal for Modern Trends in Science and Technology, 8(S08): 173-178, 2022

ISSN: 2455-3778 online
DOI: https://doi.org/10.46501/IJMTST08S0830
Available online at: https://www.ijmtst.com/vol8si08.html

Time-Bound Anonymous Authentication for Roaming Networks

Gorantla Sai Pujitha | K. Madhu Kiran

Computer Science and Engineering, Chalapathi Institute of Engineering and Technology, Guntur, AP, India

To Cite this Article
Gorantla Sai Pujitha and K. Madhu Kiran. Time-Bound Anonymous Authentication for Roaming Networks. International Journal for Modern Trends in Science and Technology 2022, 8(S08), pp. 173-178. https://doi.org/10.46501/IJMTST08S0830

Article Info
Received: 26 May 2022; Accepted: 24 June 2022; Published: 28 June 2022.

ABSTRACT

We propose an anonymous authentication protocol that supports time-bound credentials for efficient revocation. It isespecially suitable for large scale network in roaming scenario. With our newly designed group signature scheme as a building block, a timestamp can be embedded to user secret key. No expired key can be used to authenticate, and hence naturally revoked users (e.g., due to contract expiration) are not required to be put into the revocation list. This makes our protocol much faster than previous roaming protocols in terms of revocation checking, which is a main part in verification. Index Terms—accountable privacy, anonymous roaming, applied cryptography, authentication, privacy,revocation.

KEYWORDS: ROAMING NETWORKS, TIME BOUND ANONYMOUS, NETWORKS

1. INTRODUCTION

In mobile communications, roaming means a device going from its home location to a different location where it will connect to a foreign network for services. It allows mobile users to have connectivity careless of their geographic allocation. Users can make or receive phone call and SMS, or even access to the Internet from their mobile devices in any place on the Earth, provided that it is under a registered network coverage. Prior to connecting to a new foreign network, authenticationmust be made to protect the user and the network service provider. Figure 1 depicts the authentication model.

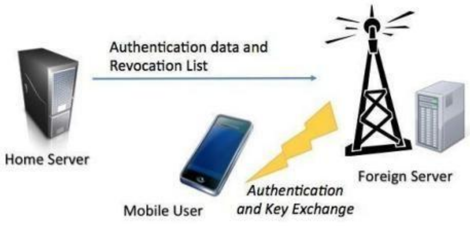

Fig:1 Authentication of Roaming Networks

2. ENTITIES RELATED TO ROAMING SERVICE

At the first glance, N will certainly increase as more users are revoked. This explains why most schemes take the first approach to design more efficient revocation check, and the second draws little attention from researchers. One approach to ―reduce‖ N is to make the system operates in epochs. However, this could boost up the size of the system parameter to be linear in the number of epochs. To have a closer look, verification phase in the authentication process consists of two steps:

(1) ―Validity Check‖ verifies if the authentication token is produced by a valid user; and (2) ―Revocation Check‖ verifies if the user has been revoked. Existing approaches often make a decision regarding the validity checking based on a single dimension, without considering time-sensitive checking, i.e., a token_s validity may depend on the current time. Generally speaking, either

be—natural‖or—premature‖. A user is—naturally revoked‖ when his access rights has expired, or a user can be prematurely revoked before the expiry time, say due to the compromise of the credential.

We believe natural revocation accounts for most user revocation in practice and prematurely revoked users are only a small fraction. A better approach is to deal with natural revocation‖ in the stage for validity check instead. This paper investigates an efficient approach for such validity check. A. Our Contribution As argued, there are unique requirements for anonymous authentication in roaming, and the investigation of an efficient approach in this setting is of interest to both academia and industry. We propose an anonymous authentication roaming protocol, with the following distinctive features:

1) Each user secret key issued by the home server is bounded to an expiry time. It is infeasible to use an expired user secret key for successful For revocation of existing anonymous authentication protocols we observe that the designs involve a boolean function RCheck() within the verification algorithm which takes an authentication token _ and a revocation list. RL =

{R1;R2; _ _ _ ;RN} as the input, where Ri is an element representing the revoked user i and N is the number of revoked users. An anonymous authentication token is not generated by a revoked user if and only if for every element In this case, Total Cost of Revocation Check = C _ N where C is the computation cost of RCheck() which is a design-specific constant. Equation

(1) suggests two approaches to speed up the revocation check:

1) a smaller C, i.e., more efficient designs of RCheck();
2) or a smaller N, i.e., less revoked users authentication.
3) Our design leads to a shorter revocation list which only contains the information on prematurely revoked identities but not those expired naturally.

The overall computation time and cost in the authentication will be greatly reduced since most of the portion is consumed by revocation check in situations where expired keys contribute most to user revocation. Our scheme requires additional computation overhead in the verification, due to the checking of the expiry information embedded in the secret key. However, the additional cost is a constant and independent of the number of revoked users. Furthermore, when there are large number of users revoked due to expiration, the efficiency savings in our design, i.e., short revocation messages and fast revocation check, will far outweigh the efficiency loss incurred by the overhead. Our estimated performance shows that our protocol is several times faster (in the server side) than previous protocols that support revocation while the overhead induced in the user side is just a few seconds. We argue that in some large scale networks (e.g. mobile network in China that contains billion users), our protocol can be hundred times faster than previous protocols. B. Enhancement over Our Conference Version Compared with the conference version[17]which merely provides a cryptographic treatment of verifier local revocation group signatures with time-bound keys, we not only

identify a few distinctive requirements of authentication in roaming protocol and how this notion can be helpful, but also significantly extend the primitive into practical roaming protocol and further improve its efficiency via a new design. Directly using our old scheme [17] for authentication in our roaming protocol results in a communication transcript of size O(`), where ` is the bit-length for representing a time period. Here we further propose a new design with O(1)-size transcript by using the accumulator system[29],and simplify the encoding method of time period. Intuitions of the design behind our cryptographic construction will be given shortly.

3. HOME NETWORK AND VISITED NETWORK

A. Intuition In our protocol, we require a group signature scheme as a primitive for both anonymous authentication and premature revocation. Group signature, introduced by Chaum and van Hey st [12], allows a member of a group to sign messages on behalf of the group without leaking his identity. But there is a group manager who has some trapdoor information which allows him to recover the identity of the signer from any valid group signature. A normal group signature cannot achieve our goal since the signature itself does not bear with any time-related information. Traditional revocation approach is inefficient since it either checks every entry in the revocation list or requires the manager to open all signatures. Our main contribution is an efficient technique in realizing time-bound key which is integrable with our underlying newly designed group signature scheme. In the authentication phase, the foreign server gives a user UI a challenge message m and a current time t. If UI can generate a valid signature on m and prove that t.

4. INBOUND AND OUTBOUND ROAMING

We first describe the new group signature scheme, which is followed by the complete description of our roaming protocol. A. Our Newly Designed Primitive We design a group signature scheme which allows users to authenticate themselves with both constant size transcripts and full anonymity. It was unknown [17] how to achieve both simultaneously. The old approach [17] is that either full anonymity is achieved at the cost of transcript size logarithmic in the total number of supported time periods, or constant-size communication with weakened anonymity guarantee. A group signature scheme consists of a tupple of probabilistic polynomial-time algorithms. During the group manager generates a master public key gpk and a master secret key msk. Gpk is published while msk is kept secret. During Gp Join, the group manager uses msk to generate a user secret key gski for user Ui and are vocation token grti which is used to trace user Ui . During Gp. Sign, a user Ui uses his secret key gski to generate a signature for a message mattimet. Gp.Ver takes mpk; t; m; and returns valid or invalid

2) Exculpability: A group signature scheme is exculpable if no polynomial-time adversary can forge a signature that is attributed to an honest member such that the member cannot dispute. Here we assume the group manager is honest. Consider the game between an adversary A and a challenger C as follows. Setup. C performs the initialization phase and obtains the result(gpk;msk).Its ends Ampk. C prepares an empty revocation list RL. Query. A issues the following queries to C. Join: A requests for creating a new group member with a designated key expiry time. C per-forms the user joining phase locally and gets gski for a new index i. A gets grti . Corrupt(i): C returns the secret key gski of user i to A and adds i to RL. Sign(i; t; m): C returns the signature signed by gski at time t, on message m or _?_. Revoke(i): C returns the revocation token grti of user i and adds i to RL. Forge. A outputs a message m, a signature and a revocation token grt, which must be one of those introduced in the Join process. We say that A wins the game if 1) is a valid signature on m with the revocation list RL. 2) is not obtained from the signing queries with expiry time t on m. 3) is traced to a user with the revocation token grt. 3). Efficiency Analysis We compare the performance of our scheme with two existing roaming protocols that support user revocation by Yang [31] et al. and He [20] et al. We first consider authentication at the user side. Public key operations are counted as follows: for signature scheme we deploy ECDSA [1] which takes 1 G1 exponentiation operation for signing, and 1 for verification. We analyze the efficiency from an estimation based on the benchmark from jPBC5 on the timing of various mathematical operations required in the system implementation. We count the number of basic operations required in various protocol and provide an estimation based on the benchmark of the jPBC library for the following devices: Our estimation assumes that there are 1,000,000 users being revoked per year, including those naturally revoked(contract expired)and prematurely revoked (e.g. key compromised, phone stolen). We argue that this figure is reasonable in some large scale networks. For example, as of September 2013, there are more than 1.2 billion mobile users in China6 . However, there are only 3 operators in the whole country. Each operator contains hundreds of million subscribers.

5. ROAMING SCENARIO'S

A. Automatic UI exploration Dynamic analysis of applications to understand their runtime properties requires that we be able to run them in some way with adequate code coverage. Android Applications are primarily GUI based and so recent work has taken the approach of automatically exercising the application_s graphic user interface in black-box or white-box manner. Apps Playground [6] implemented a general framework for exercising Android applications to check privacy leakages in applications as well as the presence of certain malicious behaviors. We have followed Apps Playground_s approach to UI exploration in this paper. Azim et al. proposed A 3 E [21] with targeted and depth-first exploration as the high points of their solution while Choi et al. [22] developed an active-learning based solution that minimizes application restarts. Numerous other application-oriented works have also used automated UI exploration. Liu et al. [23] automatically explore the application UI of Windows applications to identify cases of ad fraud. Sounthiraraj et al. [24] use automatic UI exploration as part of their methodology to verify the presence SSL/TLS certificate validation vulnerabilities in Android applications. Ravindranath et al. [25] automatically explore applications to detect faults and crashes in them. Bhoraskar et al. [26] perform a preliminary static analysis to prune away irrelevant code and instrument the applications so that automatic exploration can easily reach the right parts in the instrumented applications. Haoet al. [27] propose a one-stop UI exploration framework that is customizable to meet the requirements of different applications. The above techniques are mostly black box when it comes to exploring the UI (not considering the preliminary static analysis involved in some works above). Other works have also used white box approaches to improve application code coverage. App Intent [28] develops techniques to effectively use symbolic execution on Android applications for analyzing privacy leakages. Xia et al. [29] also use a white box dynamic analysis to accurately identify privacy leakages in Android applications. Our work uses automatic exploration as a technique to accomplish triggering of app-web interfaces and thus use any of the above works or any future advancements in this area to improve the triggering of app-web interfaces. B. Advertisement Security and Privacy Mobile advertisements have been studied in the past from multiple security and privacy perspectives such as ad fraud and security and privacy implications of using ad-supported applications. Liu et al. [23] study a kind of ad fraud in which the developer places ads and the main application widgets in such a way that it becomes easy for the user to mistakenly click on ads. Crussell et al. [30] study ad fraud in mobile applications from a network perspective. They identify repackaged applications 12 with the purpose to direct ad revenue away from the original developers and to the persons who repackaged the applications and study the prevalence and implications of this kind of ad fraud. Our main concern in this paper is not adfraud but the propagation of malicious content through advertisements and web links embedded in applications

6. BENFITS OF TECHNOLOGY INROAMING

To check whether a user is in the revocation list, it is required to check against all users in the list. The time thus grows linearly. In Figure 3, we plot the time consumed for each user revocation checking (the time required to check whether a user is in the revocation list)as the Y-axis(0-200),against the number of years the system has been used as the X-axis (0-5). Again, we use some existing results from jPBC. It is worth noting that the server in reality may be more powerful and support parallel processing, e.g., a octacore processor, and the running time can be shortened significantly. Another way is to divide the revocation list into n pieces and giveit to n computers for checking independently. However,the ratios of running time between our scheme and other schemes remain the same. Moreover, we believe in most of the cases, the majority in the revocation list (say, at least 60%) are naturally revoked. From Figure 3 we can see that if the server uses a 2.4 GHz processor, our scheme takes about 6 minutes (after the system has been running for 5 years) if 60% of the list are naturally revoked users. Yang_s scheme takes 180 minutes while He_s scheme takes 90 minutes. Even if there are 10 computers for parallel processing, Yang_s scheme still takes 18 minutes. While He_s scheme takes 9 minutes. Our scheme only requires 0.6 minute and 0.3 minute for 60% and 80% users who are naturally revoked, respectively. From the result, we can see that although our protocol requires more computation on the user side during the authentication process (about 4 seconds in our simulation), the time is still acceptable even for a 1GHz processor device (the overall running time is still around 6 seconds). On the other hand, the roaming server requires much less computation time for the revocation checking. The time should be practical enough to be deployed in real life scenarios. By having an efficient revocation checking, our protocol actually improves the performance of the whole roaming authentication protocol.

We proposed an anonymous authentication roaming protocol that supports efficient revocation of naturally expired credentials. It relies on the underlying newly designed group signature scheme which can bind the expiry time to the secret key of every user. With this new feature, expired keys are no longer needed to be included in there vocation list since the authentication token generated by those keys will be invalid.

A. Automatic UI exploration Dynamic analysis of applications to understand their runtime properties requires that we be able to run them in some way with adequate code coverage. Android Applications are primarily GUI based and so recent work has taken the approach of automatically exercising the application_s graphic user interface in black-box or white-box manner. Apps Playground [6] implemented a general framework for exercising Android applications to check privacy leakages in applications as well asthe presence of certain malicious behaviors. We have followed Apps Playground_s approach to UI exploration in this paper. Azim et al. proposed A 3 E [21] with targeted and depth-first exploration as the high points of their solution while Choi et al. [22] developed an active-learning based solution that minimizes application restarts. Numerous other application-oriented works have also used automated UI exploration. Liu et al. [23] automatically explore the application UI of Windows applications to identify cases of ad fraud. Sounthiraraj et al. [24] use automatic UI exploration as part of their methodology to verify the presence SSL/TLS certificate validation vulnerabilities in Android applications. Ravindranath et al. [25] automatically explore applications to detect faults and crashes in them. Bhoraskar et al. [26] perform a preliminary static analysis to prune away irrelevant code and instrument the applications so that automatic exploration can easily reach the right parts in the instrumented applications. Hao et al. [27] propose a one-stop UI exploration framework that is customizable to meet the requirements of different applications. The above techniques are mostly black box when it comes to exploring the UI (not considering the preliminary static analysis involved in some works above). Other works have also used white box approaches to improve application code coverage. App Intent [28] develops techniques to effectively use symbolic execution on Android applications for analyzing privacy leakages. Xia et al. [29] also use a white box dynamic analysis to accurately identify privacy leakages in Android applications. Our work uses automatic exploration as a technique to accomplish triggering of app-web interfaces and thus use any of the above works or any future advancements in this area to improve the triggering of app-web interfaces. B. Advertisement Security and Privacy Mobile advertisements have been studied in the past from multiple security and privacy perspectives such as ad fraud and security and privacy implications of using ad-supported applications. Liu et al. [23] study a kind of ad fraud in which the developer places ads and the main application widgets in such a way that it becomes easy for the user to mistakenly click on ads. Crussell et al. [30] study ad fraud in mobile applications from a network perspective. They identify repackaged applications 12 with the purpose to direct ad revenue away from the original developers and to the persons who repackaged the applications and study the prevalence and implications of this kind of ad fraud. Our main concern in this paper is not adfraud but the propagation of malicious content through advertisements and web links embedded in applications. In order to curb malware and scam attacks on mobile platforms it is important to understand how they reach the user. In this paper, we explored the app-web interface, wherein 13 a user may go from an applicationto a Web destination via advertisements or web links embedded in the application.

We used our implemented system for a period of two months to study over 600,000 applications in two continents and identified several malware and scam campaigns propagating through both advertisements and web links in applications. With the provenance gathered, it was possible to identify the responsible parties (such as ad networks and application developers). Our study shows that should such as system be deployed, the users can be offered better protection on the Androidecosystem by screening out offending applications that embed links leading to malicious content as well as by making ad networks more accountable for their ad content. We proposed an anonymous authentication roaming protocol that supports efficient revocation of naturally expired credentials. It relies on the underlying newly designed group signature scheme which can bind the expiry time to the secret key of every user. With this new feature, expired keys are no longer needed to be included in the revocation list since the authentication token generated by those keys will be invalid. This results in a significant efficiency improvement for revocation checking, due to the elimination of the expired keys in the revocation list. Moreover, compared with the conference version of this paper, we described the complete roaming protocol instead of just the group signature primitive. We further reduced the underlying group signature size from O(`) to a constant size, where ` is the bit-length for representing a time period, without losing any user anonymity. This makes our construction more practical in the roaming network environment.

Conflict of interest statement

Authors declare that they do not have any conflict of interest.

REFERENCES

[1] ANSI X9.62. Public Key Cryptography for the Financial Services Industry: The Elliptic Curve Digital Signature Algorithm (ECDSA), 1999. Masayuki Abe, Sherman S. M. Chow, Kristiyan Haralambiev, and Miyako Ohkubo. Double Trapdoor Anonymous Tags for Traceable Signatures. Int. J. Inf. Sec., 12(1):19–31,2013.

[2] Tolga Acar, Sherman S. M. Chow, and Lan Nguyen. Accumulators and U-Prove Revocation. In Financial Cryptography and Data Security, pages 189–196,2013.

[3] Man Ho Au, Apu Kapadia, and Willy Susilo. BLACR: TTP-Free Blacklistable Anonymous Credentials with Reputation. In NDSS,2012.

[4] ManHoAu,JosephK.Liu,WillySusilo,andTsz Hon Yuen. Secure ID-based Linkable and Revocableiff-Linked Ring Signature with Constant-size Construction. Theor. Comput. Sci., 469:1– 14,2013.

[5] ManHoAu,WillySusilo,YiMu,andSherman S. M. Chow. Constant-Size Dynamic k-Times Anonymous Authentication. IEEE Systems Journal, 7(2):249–261, 2013. Mihir Bellare, Daniele Micciancio, and Bogdan Warinschi. Foundations of Group Signatures: Formal Definitions, Simplified Requirements, and a Construction Based on General Assumptions. In EuroCrypt, volume 2656 of LNCS, pages 614– 629. Springer, 2003.

[6] Dan Boneh and Xavier Boyen. Short Signatures Without Random Oracles. In EuroCrypt, volume 3027 of LNCS, pages 56–73. Springer,2004.

[7] Dan Boneh and Hovav Shacham. Group Signatures with Verifier-Local Revocation. In CCS, pages 168–177. ACM,2004.

[8] Ernie Brickell and Jiangtao Li. Enhanced Privacy ID: A Direct Anonymous Attestation Scheme with Enhanced Revocation Capabilities.

[9] IEEE Trans. Dependable Sec. Computer, 9(3):345–360,2012. [10] Julien Bringer and Alain Patey. VLR Group Signatures - How to Achieve Both Backward.