Security limitations of an authorized anonymous ID-based scheme for mobile communication

This item was submitted to Loughborough University's Institutional Repository by the/an author.

Citation: PHAN, R.C.-W., 2005. Security limitations of an authorized anonymous ID-based scheme for mobile communication. in: IEEE Communications Magazine, 43(5), pp.149 - 153

Additional Information:

- This journal article is published in IEEE Communications Magazine[© IEEE]. It is also available at: http://ieeexplore.ieee.org/ Personal use of this material is permitted. However, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution to servers or lists, or to reuse any copyrighted component of this work in other works must be obtained from the IEEE.

Metadata Record: https://dspace.lboro.ac.uk/2134/5677

Version: Published

Publisher: © IEEE

Please cite the published version.
This item was submitted to Loughborough’s Institutional Repository (https://dspace.lboro.ac.uk/) by the author and is made available under the following Creative Commons Licence conditions.

For the full text of this licence, please go to: http://creativecommons.org/licenses/by-nc-nd/2.5/
Security Limitations of an Authorized Anonymous ID-Based Scheme for Mobile Communication

Raphael C.-W. Phan, Swinburne University of Technology

ABSTRACT

In this article we discuss the security limitations of a recently proposed authorized anonymous ID-based scheme for mobile communications due to He et al. We present three example attacks an attacker could mount on the scheme, point out the weaknesses we exploited, and suggest how to counter them. Our attacks are variants of the replay attack to which any security scheme should be resistant. Such attacks are easy to mount since they simply require replaying previous valid messages, and are often passive attacks and thus hard to detect. Therefore, our results are devastating since they show that the scheme has failed to achieve its main objective of establishing mutual authentication between legitimate parties.

INTRODUCTION

The issue of location privacy of mobile users is frequently addressed. This is because users are unwilling to have their locations tracked while they roam the mobile network since this would intrude on their privacy. One of the ways to guarantee the location privacy of users is to use anonymous IDs.

Recently, in the May 2004 issue of this magazine, He et al. [1] presented an authorized anonymous ID-based scheme that eliminates the need for a trusted third-party server, in contrast to previous work that depended on it. Their scheme exploited the concept of blind signatures [2] to generate authorized anonymous IDs for mobile users.

In this article we show the security limitations of He et al.’s scheme. In particular, we show that their scheme allows several attacks: impersonation attacks by any malicious mobile user against the administrator and by any malicious administrator against the user, and a nontrivial denial of service (DoS) attack by anyone against the user. All these attacks are variants of the replay attack, which is the most basic attack against which any security protocol must guard [3]. Such attacks are devastating since they are so easy to mount (they merely involve replaying previous messages and having them considered valid by legitimate parties) and mostly passive [4] in nature (and indeed this is the case for our impersonation attacks), and thus almost impossible to detect or trace. Being susceptible to such attacks is a critical failure of the scheme to achieve its main objective of establishing mutual authentication among legitimate parties.

We also discuss in detail which weaknesses of the scheme we are exploiting in our attacks and further suggest countermeasures to guard against them. Our countermeasures are very feasible since they require only slight changes to the existing scheme by introducing extra information (timestamps or information about the sender and receiver) in a few communicated messages, or simply including an extra response message.

THE SCHEME OF HE ET AL.

The authorized anonymous ID-based scheme of He et al. consists of the following parties (also illustrated in Fig. 1):

- **Administrator, A**: The administrator authenticates legitimate users and grants them access to the mobile communication network it is overseeing.
- **Mobile user, U**: Any user of a mobile device who wishes to gain access to the mobile communication network administered by A.
- **Connector, C**: An access point delegated by A to authenticate the Us; it controls connections between Us and the network.

Before we proceed to describe the two protocols, we define additional notation to be used throughout this article.

The scheme makes use of both symmetric-key cryptosystems and public-key cryptosystems [4]. A symmetric-key cryptosystem allows two parties wishing to communicate secretly to use a secret key shared between them to encrypt

---

1 The actual scheme makes use of agents to act on behalf of the parties, but since our attacks work regardless of this detail, we shall simply concentrate on the parties.
and decrypt messages for each other. Both encryption and decryption use the same secret (symmetric) key. In contrast, a public-key cryptosystem requires that each party own a pair of keys, a public key (broadcast to everyone) and a private key (known only to the party who owns it). Encryption with a public key can only be decrypted by its corresponding private key, and vice versa.

When party $A$ wishes to send a confidential message to party $B$, it encrypts the message with the public key, $E_B$, of $B$, thus ensuring that only $B$ can decrypt it with its private key, $D_B$. When party $A$ wishes to prove that a message is indeed generated by itself, it decrypts (signs) the message with its private key, $D_A$. Then anyone can verify that the message came from $A$ by encrypting (verifying) this with $A$’s public key, $E_A$.

The authorized anonymous ID-based scheme proposed by He et al. [1] consists of two phases specified by two corresponding protocols: the registration protocol and the controlled connection protocol. The registration protocol is to authorize a mobile user, $U$, who applies for an authorized anonymous ID from the administrator, $A$. Meanwhile the controlled connection protocol is to control access of the network to only legitimate and authorized users. Here, the mobile user, $U$, presents his/her obtained authorized anonymous ID to an access point connector, $C$, in order to request connection. This ID is also used to authenticate the packets from the mobile user, $U$, to the access point connector, $C$, for access control purposes.

**REGISTRATION PROTOCOL**

A summary of the steps in the registration protocol is shown in Fig. 2.

The registration protocol assumes that an infrastructure such as a public key infrastructure (PKI) [4] is already in place and allows for the initial authentication of parties. Therefore, the mobile user, $U$, must digitally sign its request, $c_0$, using its private key, $D_U$, before sending it to the administrator, $A$. Upon reception of this request message ($c_0$, $D_U(c_0)$), $A$’s action of authenticating $U$ would be to verify $D_U(c_0)$ by encrypting it with $U$’s public key, $E_U$ and checking that the resulting value equals the received $c_0$.

At the end of this protocol, the mobile user, $U$, has obtained his authorized anonymous ID, and, $c_0 = D_A(H(r_1))$, which will be used in the next phase whenever he wishes to request access to the network at any access point connector, $C$.

**CONTROLLED CONNECTION PROTOCOL**

The steps in this protocol are summarized in Fig. 3.

Whenever the mobile user, $U$, wishes to access the network at an access point connector, $C$, it initiates the controlled connection protocol by sending an access request, $c_2$, to $C$, which it immediately forwards to the administrator, $A$. The administrator decrypts the access request message using its private key, $D_A$, and proceeds to verify the authenticity of this decrypted value. It then encrypts this with the secret key, $K_{CA}$, that it shares with the access point connector, $C$, and sends this back to $C$. Upon receipt, $C$ decrypts the message and also verifies its authenticity. Then it generates and sends the ack message to the mobile user, $U$, signifying that access has been granted to $U$. Thereafter, the access point connector, $C$, and $U$ can communicate with each other by sending authenticated packets between them.

**ATTACKS ON THE SCHEME**

In this section we describe three different attacks on He et al.’s authorized anonymous ID-based scheme.
IMPERSIONATION ATTACK BY MALICIOUS USER AGAINST ADMINISTRATOR

We first show an attack that can be mounted by a malicious mobile user $U$ against the administrator $A$ such that $U$ can gain access to any other network of which $A$ is a user.

In more detail, suppose that $U$ is a legitimate user on a network, $N_1$, of which $A_1$ is the administrator. Furthermore, suppose that $A_1$ is a legitimate user on another network, $N_2$, of which $A_2$ is the administrator. This is a typical scenario since diverse mobile and wireless networks are allowed to coexist in our current information age, and is depicted in Fig. 4.

The malicious mobile user $U$'s intention is to impersonate $A_1$ as a legitimate user to administrator $A_2$ in network $N_2$. In other words, $U$ would like to generate an id authorization request message, $c_0$ of its own, as a function of its own chosen random numbers, $r_0$ and $r_1$:

$$c_0 = E_{A_2}(r_0) \times H(r_1).$$  \hspace{1cm} (1)

However, in order for this to be authenticated by administrator $A_2$ and appear to have really come from $A_1$, note that $U$ also needs to obtain the value of $D_{A_1}(c_0)$ so that it can complete forming the id authorization message $(c_0, D_{A_1}(c_0))$. Surprisingly, this is very easy to achieve. Since $U$ is a legitimate user of network $N_1$, it simply forms the message $(c_0, D_{A_1}(c_0))$ to $A_1$ as an apparent id authorization request in the registration protocol session of network $N_1$. Since $A_1$ is administrator of network $N_1$, it will verify the authenticity of the message, which will be successful since $U$ is a legitimate user. The administrator $A_1$ then computes $c_1 = D_{A_1}(c_0)$ and returns this to the mobile user $U$, and therefore $U$ has successfully used the administrator $A_1$ as a decryption (signing) oracle to compute the value $D_{A_1}(c_0)$. $U$ is now able to form the complete message $(c_0, D_{A_1}(c_0))$ to administrator $A_2$ in the registration protocol session of network $N_2$, and would successfully be authenticated to $A_2$ as the legitimate $A_1$ when in fact $A_1$ did not initiate the protocol session at all. This is an impersonation attack by mobile user $U$ against administrator $A_1$, and allows $U$ to act as a legitimate user of network $N_2$.

IMPERSIONATION ATTACK BY MALICIOUS ADMINISTRATOR AGAINST USER

Suppose that $U$ is a legitimate user of two networks, $N_1$ and $N_2$, where $A_1$ and $A_2$ are the administrators, respectively, and $A_1$ is a legitimate user of network $N_2$. This scenario is depicted in Fig. 5.

At the end of the registration protocol in $N_1$, the legitimate user $U$ would have successfully used its chosen random numbers, $r_0$ and $r_1$, to obtain its authorized id.

We emphasize the critical information derived by administrator $A_1$ from both the registration and controlled connection protocols with user $U$ in network $N_1$. After the registration protocol, $A_1$ knows $r_0 \times D_{A_1}(H(r_1))$. After the controlled connection protocol, $A_1$ further knows the value of $r_1$ and $D_{A_1}(H(r_1))$. From these, $A_1$ can compute

$$r_1 = \frac{D_{A_1}(H(r_1))}{D_{A_1}(H(r_1))}.$$  \hspace{1cm} (2)

With these $r_0$ and $r_1$ values, $A_1$ can now impersonate user $U$ in network $N_2$ by pretending to be $U$ to initiate a registration protocol in network $N_2$ with administrator $A_2$ and replaying the id authorization request message, $(c_0, D_{A_1}(c_0))$ previously used by user $U$ in $N_1$'s registration protocol with $A_1$. This will be correctly authenticated by administrator $A_2$ as having come from user $U$ since it was indeed signed by $U$. $A_2$ then returns $c_1 = D_{A_2}(c_0) = D_{A_2}(E_{A_1}(r_0) \times H(r_1)) = D_{A_2}(E_{A_1}(r_0)) \times D_{A_2}(H(r_1))$. In order for $A_1$ to remove the blind factor $D_{A_2}(E_{A_1}(r_0))$ to obtain the anonymous authorization ID, $\text{id} = D_{A_2}(H(r_1))$. $A_1$ legitimately (as itself) initiates a registration protocol in network $N_2$ with admin-
Consider that legitimate user $U$ initiates a normal registration protocol session with administrator $A$ by sending the digitally signed message $(c_0, D_A(c_0))$. However, unauthorized party $I$ intercepts this and multiplies the first component with $E_U(\alpha)$, while the second component is multiplied with $\alpha$, where $\alpha$ is any arbitrary value. $I$ then forwards the resultant products, $(c_0 \times E_U(\alpha), D_A(c_0) \times \alpha)$, to administrator $A$. Upon receipt of this, $A$ verifies the authenticity of the message by using $U$’s public key, $E_U$, to encrypt the second component, $D_A(c_0) \times \alpha$, to obtain $c_0 \times E_U(\alpha)$ and therefore successfully gets a match with the received first component. Hence, as far as $A$ is concerned, legitimate user $U$ has successfully authenticated itself to $A$ and the message received indeed came from $U$. Then administrator $A$ computes $c_1' = D_A(c_0 \times E_U(\alpha)) = D_A(E_A(r_0) \times H(r_1) \times E_U(\alpha)) = (r_0 \times D_A(H(r_1) \times E_U(\alpha)))$ and sends this back to legitimate user $U$. However, $U$ is unable to verify the authenticity of this message and therefore aborts. Administrator $A$, on the other hand, thinks that the registration protocol session has successfully been completed. This shows that the registration protocol fails to achieve its objective of fully authenticating both parties (legitimate user $U$ and administrator $A$).

**Countermeasures**

The main problem we exploited in mounting our two impersonation attacks is that during the registration protocol the network administrator can be used as an oracle since the administrator would decrypt (sign) and return any authentic $c_0$ it receives. Furthermore, since the authenticated $c_0$ is merely signed by the user but does not contain any timestamp, previously valid messages can be replayed by an attacker for future communications. Note that while He et al. did suggest using timestamps, it was only for ensuring the freshness of the authorized id, not for the messages communicated between the parties in both registration and controlled connection protocols. Therefore, the first countermeasure is that all messages should be timestamped in order to prevent such replay attacks.

Another countermeasure to complicate replay attacks is to explicitly include both the sender’s and receiver’s identities into a message that can only be signed (i.e., decrypted using the private key of) the sender. For instance, if the id authorization request message $(c_0, D_A(c_0))$ sent during the registration protocol from user $U$ to administrator $A$ is changed to $(c_0, D_A(c_0 \| U \| A))$, and similarly the administrator’s reply, $c_1$, to the user is changed to $c_1 = D_A(c_0 \times A \| U)$, where $\|$ denotes concatenation, our attacks would no longer work since a malicious attacker would no longer be able to replay previously valid messages as being sent from or to incorrect parties.

To prevent the nontrivial DoS attack described earlier, we suggest using a challenge-response mechanism [4] toward the end of the registration protocol to allow administrator $A$ to verify that user $U$ really received the correct $c_1$. 

**Nontrivial denial of service attack by any unauthorized party against user**

We next show a nontrivial DoS attack by any unauthorized party, $I$, who does not need to be a legitimate user of any network. This attack causes a legitimate user, $U$, to be denied access to a network to which it is supposed to have access, and is nontrivial in the sense that the administrator, $A$, is unable to detect anything wrong with the registration protocol. This scenario is illustrated in Fig. 6.
This can be done, for example, by treating $c_1$ as a challenge to $U$; then $U$ is further required to sign this and return it as the response message $D_A(c_1)$, so $A$ is assured that $c_1$ is received correctly. A further weakness we exploited is that any knowledge of $(r_1, D_A(H(r_1)))$ can be used to gain access to the system during the controlled connection protocol. No further authentication is required of the user. This therefore highlights that previous such pairs, although outdated and no longer used, should not be disclosed to anyone to prevent so-called garbage-man-in-the-middle [5] attacks. Similarly, the shared session key, $K_{CA}$, between access point connector $C$ and administrator $A$ during the controlled connection protocol should never be disclosed even though outdated, since this would allow an attacker to read previous $c_1$ messages, obtain a previously valid $(r_1, D_A(H(r_1)))$ pair, and hence impersonate a legitimate user. In particular, the lifetime of $K_{CA}$ must at least be longer than that of $(r_1, D_A(H(r_1)))$.

CONCLUDING REMARKS

We have discussed security limitations of the authorized anonymous ID-based scheme proposed by He et al. and given suggestions on how to modify the scheme to counter such attacks.

We have one final concern relating to He et al.’s suggestions to improve [1] their scheme by the access authorization revocation and reconfusion methods: The access authorization revocation method revokes a certain outdated authorized anonymous ID. Meanwhile, the reconfusion method allows the user to request that the administrator generate a new authorized anonymous ID to replace the old one. Specifically, user $U$ sends a reconfusion request to administrator $A$ that is merely a function of the old authorized anonymous ID of $U$, two random values chosen by $U$, and a secret key, $K_{UA}$, also chosen by $U$ for symmetric encryption between $U$ and $A$. From this, the administrator generates the new authorized anonymous ID, encrypts this with $K_{UA}$, and sends the encrypted value back to $U$.

This means that through the reconfusion method, a malicious user or any attacker with knowledge of a user’s old authorized anonymous ID could obtain multiple new anonymous authorized IDs, which he could use even if the old authorized anonymous ID has been revoked via the access authorization revocation method. Therefore, we suggest that the reconfusion protocol should also check for freshness of the reconfusion request (via timestamps or having this protocol work in collaboration with the access authorization revocation method) so that new anonymous authorized IDs will not be generated if the reconfusion request contains a revoked authorized anonymous ID.

ACKNOWLEDGMENTS

We are grateful to the anonymous reviewers whose many comments helped to improve the presentation of this article. We also thank Dennis M. L. Wong for his pointers on preparing eps graphics with Latex.

REFERENCES


BIOGRAPHY

RAFAEL PHAN [M] (rphan@swinburne.edu.my) received a B.Eng. (Hons) electronics degree majoring in computer engineering from Multimedia University (MMU), Cyberjaya, Malaysia, in 1999 and an M.Eng.Sc. degree on “Cryptanalysis of the Advanced Encryption Standard & Skipjack” in 2001. He was also a tutor with the Faculty of Engineering, MMU, and a researcher at the Center for Smart Systems and Innovation, MMU from June 1999 to June 2001. He is currently director of the Information Security Research (CURES) Laboratory and lecturer with the School of Engineering, Swinburne University of Technology (Sarawak Campus), Kuching, Malaysia. He serves as a reviewer for IEEE Transactions on Computers, IEEE Communications Letters, IEEE Signal Processing Letters, IEEE Transactions on Image Processing, IEEE Transactions on Image Processing, Information Processing Letters, Cryptologia, and IEEE Spectrum. His research interests include cryptography, cryptanalysis, block ciphers, authentication and key exchange protocols, side-channel attacks, digital watermarking, and smart card security.