On Security of Anonymous Invitation-Based System

Naoto Yanai(✉) and Jason Paul Cruz

Osaka University, 1-5 Yamadaoka, Suita, Osaka 565–0871, Japan
yanai@ist.osaka-u.ac.jp

Abstract. In an anonymous invitation-based system, a user can join a group by receiving invitations sent by current members, i.e., inviters, to a server anonymously. This kind of system is suitable for social networks, and a formal framework with the anonymity of inviters and the unforgeability of an invitation letter was proposed in DPM 2017. The main concept of this previous system is elegant, but the formal security definitions are insufficient and weak in a realistic application scenario. In this paper, we revise formal security definitions as attacks representing a realistic scenario. In addition, we define a new aspect of the security wherein an adversary maliciously generates an invitation letter, i.e., *invitation opacity*, and the security for guaranteeing that an invitee with a valid invitation letter can always join the system, i.e., *invitation extractability*. A secure and useful construction can be expected by satisfying the security definitions described above.

Keywords: Anonymous invitation-based system · Anonymity
Unforgeability · Opacity · Extractability · Formal definition
Social networks

1 Introduction

Backgrounds. An invitation-based system consists of a *server*, a group of members, i.e., *inviters*, and a new member called *invitee*, who can join the system by receiving invitations from a certain number of inviters. Invitation-based systems provide many advantages; for example, a limited number of server resources can cover an unlimited number of users and such systems can often resist registration of fake users. Invitation-based systems have many historical examples, such as Gmail and Google Wave, and have been discussed in some recent works [3–5], but these do not consider the *anonymity* of users. To the best of our knowledge, only the work of Boschrooyeh et al. [2] considered anonymity of users and introduced an *anonymous invitation-based system*.

An anonymous invitation-based system mainly aims to preserve the privacy of users. In general, when an invitee wants to join a system, he/she may ask an invitation letter from a known current member of a system, e.g., friends or family. However, in such scenario, the invitations may contain privacy risks that can leak

J. Garcia-Alfaro et al. (Eds.): DPM 2018/CBT 2018, LNCS 11025, pp. 415–421, 2018.
https://doi.org/10.1007/978-3-030-00305-0_29

affiliations and locations according to common features. To avoid this kind of risks, Boschrooyeh et al. [2] formalized the security of an anonymous invitation-based system and proposed a concrete construction called *Inonymous*. Their main concept is elegant, but their security definitions are not defined well and are far from realistic threats. Well-defined security definitions including realistic threats are important because they can support in understanding the security of our proposed protocol and of subsequent works.

In this paper, we reconsider the work of Boschrooyeh et al. and formalize stronger security containing realistic attacks. We believe that our security definitions can help create realistic and effective constructions of anonymous invitation-based systems.

Contribution. In this paper, we formally define the security of an anonymous invitation-based system that captures realistic threats by revisiting the security of a previous anonymous invitation-based system. The previous definition includes the anonymity of inviters, i.e., *inviter anonymity*, and the unforgeability of invitation letters sent to a server, i.e., *invitation unforgeability*. In our definitions, we give generation of secret information for each user and oracle access for an adversary with respect to invitation letters. These definitions were not captured in the previous system, and we are able to discuss the security even against an adversary who obtains knowledge of invitation letters via our definitions. Moreover, we define *two new security definitions with respect to an invitee*. The first definition considers whether only an invitee can receive a valid invitation letter, i.e., *invitation opacity*, allowing the discussion on security against an adversary who maliciously generates invitation letters. The second definition considers that an invitee can always join the system as long as he/she can obtain a certain number of invitation letters, i.e., *invitation extractability*, allowing an invitee to join the system correctly. This work is ongoing, and we leave the construction of a concrete scheme following our definitions as an open problem and future work.

2 Preliminaries

Threshold Secret Sharing. We recall a definition of a threshold secret sharing scheme [6]. Let participants in this scheme be a set of n players. A set of values $(s_1, \cdots, s_n)$ is said to be a (t, n)-*threshold secret sharing* of the value s if the following conditions hold: any subset with $k(< t)$ values does not reveal any information about s; and there exists an efficient algorithm which takes any t values from the set and outputs s.

Inonymous. We briefly recall the work of Boschrooyeh et al. [2] called Inonymous, which includes three entities, namely, a server, an invitee, and inviters. When an invitee wants to join the system, he/she first requests a token for user invitation from a server. The invitee then sends the token to inviters, who generate an invitation individually. If the invitee can receive invitations more than

some threshold value specified in advance, then he/she can get an invitation letter to join the system. Boschrooyeh et al. also defined the anonymity of inviters and the unforgeability of an invitation letter. Our definitions are continuation of the framework described above.

3 Anonymous Invitation-Based System

In this section, we formalize a syntax and new security notions. Boschrooyeh et al. [2] did not provide a syntax although they proposed security definitions of inviter anonymity and invitation unforgeability. Therefore, we first formally define our syntax and then define new security definitions.

3.1 Our Syntax

An anonymous invitation-based system is defined as follows:

Setup Given a security parameter 1^k as input, output a public parameter $para$.
ServerKeyGen Given $para$ as input, output a pair (msk, mpk) of a master secret key and a master public key.
UserKeyGen Given $para$ and two security parameters $(t, n) \in \mathbb{Z}$ such that $t \leq n$ as input, output shares s_i for (t, n)-secret sharing and its corresponding user public key upk, where a secret recovered from at least t output shares s_i is identical to a user secret key usk.
TokenGen Given $para, msk, mpk$, and an index j as input, output a token $token$.
InKeyGen Given $para$ as input, output a pair (isk, ipk) of an inviter secret key and an inviter public key.
InvGen Given $para, token$, a share s_i for the ith user, mpk, upk, and ipk as input, output an individual invitation Inv_i for the ith user or an error symbol $\perp$.
InvColl Given $para, token, mpk, upk, isk$, and t invitations $\{Inv_i\}_{i=1}^n$ as input, output an invitation letter $InvLet$ or an error symbol $\perp$.
InvVer Given $para, token, mpk, upk$, and $InvLet$, output $\top$ or $\perp$.

The correctness of the scheme is defined as follows: for any security parameters $(1^k, t, n)$, $para \leftarrow \mathbf{Setup}(1^k)$, $(\{s_i\}_{i=1}^n, upk) \leftarrow \mathbf{UserKeyGen}(para)$, $token \leftarrow \mathbf{TokenGen}(para, msk, mpk)$, and $(isk, ipk) \leftarrow \mathbf{InKeyGen}(para, isk, ipk)$, we say the scheme is correct if the following equation holds:

$$\top = \mathbf{InvVer}\left(para, token, mpk, upk, \mathbf{InvColl}\left(\begin{array}{r} para, token, mpk, upk, isk \\ \{\mathbf{InvGen}\,(para, token, s_i, mpk, upk, ipk)\}_{i \in U} \end{array}\right)\right),$$

where U is any subset of $[1, n]$ such that $|U| \geq t$.

3.2 Inviter Anonymity

In Inonymous, inviter anonymity does not allow an adversary to access oracles for invitations, i.e., invitation generation and invitation collection. The adversary may obtain knowledge about inviters, and thus considering access to such oracles is necessary. In addition, Inonymous' inviter anonymity does not include generation of shares while a pair of a master secret key and a master public key is generated by an adversary. These information should be generated by a challenger to prove the security clearly. We define a new security definition by introducing these points. Our definition is a game-based definition between a challenger $\mathcal{C}$ and an adversary $\mathcal{A}$ as follows:

Initial Phase Given security parameters 1^k and $(t, n) \in \mathbb{Z}$ as input, a challenger $\mathcal{C}$ generates a pair (msk, mpk) of a master secret key and a master public key via the setup algorithm, a pair of shares $\{s_i\}_{i=1}^n$ and a user public key upk via the user key generation algorithm, and a pair of an inviter secret key isk and an inviter public key ipk via the inviter key generation algorithm. Then, $\mathcal{C}$ sets a set U of indexes such that $|U| = n$ and initializes a list $Corr = \emptyset$ for corrupted inviters. $\mathcal{C}$ then runs an adversary $\mathcal{A}$ with $(para, msk, mpk, upk, ipk)$ as input.

Corrupt Oracle $\mathcal{A}$ queries an index $i \in U$, and $\mathcal{C}$ sets $Corr = Corr \cup \{i\}$. $\mathcal{C}$ then returns s_i as a share for the inviter corresponding to i.

InvGen Oracle $\mathcal{A}$ generates a token $token$ and chooses an index $i \in U$ as an inviter. $\mathcal{C}$ then generates an individual invitation inv_i for the inviter corresponding to i.

InvColl Oracle $\mathcal{A}$ generates a token $token$ and a set $\{inv_i\}_{i \in U' \subseteq U}$ of indexes, and then $\mathcal{C}$ returns an invitation letter $InvLet$.

Random Oracle $\mathcal{A}$ queries any input x to a hash function, and $\mathcal{C}$ returns the response y of the hash function.

Challenge $\mathcal{A}$ generates $token^*$, a set $\{in_i\}_{i \in U^* \subset U}$ where $|U^*| = t - 1$, and chooses two indexes $(u_0, u_1) \in U \backslash (U^* \cup Corr)$ as a challenge. Then, $\mathcal{C}$ chooses $b \in \{0, 1\}$ and generates an individual invitation inv_{u_b} via the invitation generation algorithm with $token^*$ and s_{u_b}. $\mathcal{C}$ then generates $InvLet^*$ via the invitation collection algorithm with $token^*$, isk, and $\{inv_i\}_{i \in U^*} \cup \{inv_{u_b}\}$ and returns $InvLet^*$.

Guess $\mathcal{A}$ outputs a guess $b' \in \{0, 1\}$ indicating which of the users u_0^*, u_1^* is used as the inviter. $\mathcal{A}$ wins the game if $b = b'$ holds. Otherwise, $\mathcal{C}$ wins the game.

Definition 1. We say that an anonymous invitation-based system satisfies $(q_r, q_i, q_c, q_h, t, n, \epsilon)$-inviter anonymity if there is no probabilistic polynomial-time adversary $\mathcal{A}$ who wins the game described above with a probability greater than ϵ. Here, $\mathcal{A}$ can access the corrupt oracle at most q_r times, the invitation generation oracle at most q_i times, the invitation collection oracle at most q_c times, and a random oracle at most q_h times, t is a threshold value, and n indicates the number of users.

3.3 Invitation Existential Unforgeability

We define invitation existential unforgeability below. In the invitation unforgeability of Inonymous, an adversary is not allowed to receive individual invitations and verification results of invitation letters. Moreover, a token utilized by an adversary for forging an invitation letter has to be designated in advance. Our definition removes these restrictions for an adversary. Our definition is a game-based definition between a challenger $\mathcal{C}$ and an adversary $\mathcal{A}$ as follows:

Initial Phase $\mathcal{C}$ generates $(msk, mpk, \{s_i\}_{i=1}^n, upk, isk, ipk)$ and a set U of indexes in a similar manner as in the game described in the previous section. However, $\mathcal{C}$ owns two additional lists, $\mathcal{T}$ and $\mathcal{IT}$, and initializes $\mathcal{T} = \emptyset$ and $\mathcal{IT} = \emptyset$. $\mathcal{C}$ then runs $\mathcal{A}$ with $(para, mpk, upk, isk, ipk)$ as input.

Corrupt Oracle This step is the same as in the game in the previous section.

TokenGen Oracle $\mathcal{A}$ chooses an index $i \in U$ as user information, and $\mathcal{C}$ generates a token $token_i$ via the token generation algorithm with msk and i. Then, $\mathcal{C}$ sets $\mathcal{T} = \mathcal{T} \cup \{token_i\}$ and then returns $token_i$.

InvGen Oracle $\mathcal{A}$ generates a token $token$ and chooses an index $i \in U$. Then, $\mathcal{C}$ sets $\mathcal{IT} = \mathcal{IT} \cup \{token\}$ and generates an individual invitation inv_i via the invitation generation algorithm with $token$ and s_i. $\mathcal{C}$ then returns Inv_i for i.

Random Oracle This step is the same as in the game in the previous section.

Output $\mathcal{A}$ outputs a token $token^*$ and an invitation letter $InvLet^*$ as a forgery. $\mathcal{A}$ wins the game if the following conditions hold: the invitation verification algorithm with $token^*$ and $InvLet^*$ outputs $\top$; $token^* \in \mathcal{T}$, $|\{token \in \mathcal{T} | token = token^*\}| + |Corr| \leq t - 1$; $|Corr| \leq t - 1$; and $token^* \in \mathcal{IT}$, $|\{token \in \mathcal{IT} | token = token^*\}| + |Corr| \leq t - 1$. Otherwise, $\mathcal{C}$ wins the game.

Definition 2. We say that an anonymous invitation-based system satisfies $(q_r, q_t, q_i, q_h, t, n, \epsilon)$-invitation existential unforgeability if there is no probabilistic polynomial-time adversary $\mathcal{A}$ who wins the game described above with a probability greater than ϵ. Here, $\mathcal{A}$ can access the corrupt oracle at most q_r times, the token generation oracle at most q_t times, the invitation generation oracle at most q_i times, and a random oracle at most q_h times, t is a threshold value, and n indicates the number of users.

3.4 Invitation Opacity

We define invitation opacity below. This definition guarantees that only an invitee can generate an invitation letter from individual invitations generated by inviters. In this setting, an adversary can corrupt several inviters and obtain tokens and invitation letters. In this definition, an adversary can be a malicious inviter or an external attacker and its goal is to generate an invitation letter for an invitee. Our definition is a game-based definition between a challenger $\mathcal{C}$ and an adversary $\mathcal{A}$ as follows:

Initial Phase $\mathcal{C}$ generates $(msk, mpk, \{s_i\}_{i=1}^n, upk, isk, ipk)$, a set U of indexes, and two lists, $\mathcal{T}$ and $\mathcal{IT}$, similar to the game in the previous section. However, $\mathcal{C}$ then runs $\mathcal{A}$ with $(para, mpk, \{s_i\}_{i=1}^n, upk, ipk)$ as input.
TokenGen Oracle This step is the same as in the game in the previous section.
InvColl Oracle $\mathcal{A}$ generates a token $token$ and a set $\{inv_i\}_{i \in U' \subseteq U}$ of indexes, and $\mathcal{C}$ generates an invitation letter $InvLet$. Then, $\mathcal{C}$ sets $\mathcal{T} = \mathcal{T} \cup \{token\}$ and $\mathcal{IT} = \mathcal{IT} \cup \{InvLet\}$ and then returns $InvLet$.
Random Oracle This step is the same as in the game in the previous section.
Output $\mathcal{A}$ outputs a token $token^*$ and an invitation letter $InvLet^*$ as a forgery. $\mathcal{A}$ wins the game if the following conditions hold: the invitation verification algorithm with $token^*$ and $InvLet^*$ outputs $\top$; $token^* \notin \mathcal{T}$; and $InvLet^* \notin \mathcal{IT}$. Otherwise, $\mathcal{C}$ wins the game.

Definition 3. We say that an anonymous invitation-based system satisfies $(q_t, q_c, q_h, t, n, \epsilon)$-invitation opacity if there is no probabilistic polynomial-time adversary $\mathcal{A}$ who wins the game described above with a probability greater than ϵ. Here, $\mathcal{A}$ can access the token generation oracle at most q_t times, the invitation collection oracle at most q_c times, and a random oracle at most q_h times, t is a threshold value, and n indicates the number of users that generate an invitation letter.

3.5 Invitation Extractability

We define invitation extractability below. This definition guarantees that the invitation verification algorithm with an invitation letter always outputs $\top$ when the invitation collection algorithm outputs the invitation letter. In this setting, information that an adversary can obtain is the same as that in the game in the previous section. However, the goal of the adversary is to output individual invitations whose resultant invitation letter is rejected. Our definition is a game-based definition between a challenger $\mathcal{C}$ and an adversary $\mathcal{A}$ as follows:

Initial Phase $\mathcal{C}$ generates $(msk, mpk, \{s_i\}_{i=1}^n, upk, isk, ipk)$, a set U of indexes, and a list $\mathcal{T}$, but does not generate $\mathcal{IT}$. Then, $\mathcal{C}$ runs $\mathcal{A}$ with $(para, mpk, \{s_i\}_{i=1}^n, upk, ipk)$ as input.
TokenGen Oracle This step is the same as in the game in the previous section.
InvColl Oracle This step is almost the same in the game in the previous section, except that $\mathcal{C}$ sets only $\mathcal{T} = \mathcal{T} \cup \{token\}$.
Random Oracle This step is the same as in the game in the previous section.
Output $\mathcal{A}$ outputs a token $token^*$ and a set $\{inv_i\}_{i \in U^* \subseteq U}$ of individual invitations, where $|U^*| \geq t$. $\mathcal{A}$ wins the game if the following conditions hold: the invitation collection algorithm with $token^*$, $\{in_i\}_{i \in U^* \subseteq U}$, and upk outputs $InvLet^*$; and the invitation verification algorithm with $token^*$ and $InvLet^*$ outputs $\perp$. Otherwise, $\mathcal{C}$ wins the game.

Definition 4. We say that an anonymous invitation-based system satisfies $(q_t, q_c, q_h, t, n, \epsilon)$-invitation extractability if there is no probabilistic polynomial-time adversary $\mathcal{A}$ who wins the game described above with a probability greater

than ϵ. Here, $\mathcal{A}$ can access the token generation oracle at most q_t times, the invitation collection oracle at most q_c times, and a random oracle at most q_h times, t is a threshold value, and n indicates the number of users that generate an invitation letter.

4 Conclusion

In this paper, we presented new security definitions of an anonymous invitation-based system from four standpoints, namely, invitation anonymity, invitation unforgeability, invitation opacity, and invitation extractability. The first two definitions are presented in Inonymous [2] and we introduced oracle access related to invitations. The last two definitions are for invitee's security and have never been discussed in previous work. We believe that a scheme that satisfies these definitions can be used for many applications.

Although we did not discuss a specific construction in this paper, we consider that a scheme may be constructed by combining verifiably encrypted signatures (VES) [1] with Inonymous. VES are digital signatures wherein a signer encrypts its signatures under a public key of a trusted third party to confirm that the signer has truly signed a certain object. We consider that a trusted third party in VES is similar to an invitee in our proposed system. We will construct a scheme using such an approach and prove its security as future work.

References

1. Boneh, D., Gentry, C., Lynn, B., Shacham, H.: Aggregate and verifiably encrypted signatures from bilinear maps. In: Biham, E. (ed.) EUROCRYPT 2003. LNCS, vol. 2656, pp. 416–432. Springer, Heidelberg (2003). https://doi.org/10.1007/3-540-39200-9_26
2. Boshrooyeh, S.T., Küpçü, A.: Inonymous: anonymous invitation-based system. In: Garcia-Alfaro, J., Navarro-Arribas, G., Hartenstein, H., Herrera-Joancomartí, J. (eds.) ESORICS/DPM/CBT -2017. LNCS, vol. 10436, pp. 219–235. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-67816-0_13
3. Gong, N.Z., Wang, D.: On the security of trusted-based social authentications. IEEE Trans. Inf. Forensics Secur. **9**(8), 1251–1263 (2014)
4. Malar, G.P., Shyni, C.E.: Facebookfs trustee based social authentication. Int. J. Emerg. Technol. Comput. Sci. Electron. **12**(4), 224–230 (2015)
5. Parameswari, M., Sukumaran, S., Kalaiselvi, S.: Trustee based authentication. Int. J. Latest Res. Sci. Technol. **4**(5), 84–88 (2015)
6. Shamir, A.: How to share a secret. Commun. ACM **22**(11), 612–613 (1979)