Flexible Fair and Collusion Resistant Pseudonym Providing System

Belal Amro[1], Albert Levi[2], and Yucel Saygin[2]
[1]College of IT, Hebron University, Palestine
[2]Faculty of Engineering and Natural Sciences, Sabanci University, Turkey

Abstract: *In service providing systems, user authentication is required for different purposes such as billing, restricting unauthorized access, etc., to protect the privacy of users, their real identities should not be linked to the services that they use during authentication. A good solution is to use pseudonyms as temporary identities. On the other hand, it may also be required to have a backdoor in pseudonym systems for identity revealing that can be used by law enforcement agencies for legal reasons. Existing systems that retain a backdoor are either punitive (full user anonymity is revealed), or they are restrictive by revealing only current pseudonym identity of. In addition to that, existing systems are designed for a particular service and may not fit into others. In this paper, we address this gap and we propose a novel pseudonym providing and management system. Our system is flexible and can be tuned to fit into services for different service providers. The system is privacy-preserving and guarantees a level of anonymity for a particular number of users. Trust in our system is distributed among all system entities instead of centralizing it into a single trusted third party. More importantly, our system is highly resistant to collusions among the trusted entities. Our system also has the ability to reveal user identity fairly in case of a request by law enforcement. Analytical and simulation based performance evaluation showed that Collusion Resistant Pseudonym Providing System (CoRPPS) provides high level of anonymity with strong resistance against collusion attacks.*

Keywords: *Security, privacy, pseudonym, anonymity, access control.*

Received August 7, 2015; accepted October 24, 2016

1. Introduction

As discussed in [14, 18], the lack of privacy is the main hindrance for the success of a service providing system that requires user authentication. This encouraged service providers to develop a privacy preserving system that protects users' privacy. Most of these systems depend on the usage of temporary identities instead of real identities. These temporary identities are called pseudonyms [10].

In this paper, we propose a novel pseudonym providing system, called Collusion Resistant Pseudonym Providing System (CoRPPS). CoRPPS distributes trust among all system parties and resists against collusion among them to reveal the real identities of the users. In this way, CoRPPS ensures a level of anonymity for users served by a particular service provider.

We have done analytical and simulation based performance evaluation mostly to analyze the security and anonymity that CoRPPS provides. Our analytical and simulation results show that CoRPPS can be applied for different applications with different number of users. By a careful selection of CoRPPS's parameters, it is possible to gain high performance results.

The rest of the paper is organized as follows. In section 2, we provide a brief survey of the related work. In section 3, the design details of our system, CoRPPS, are discussed. Resistance of CoRPPS against some attacks is also discussed in this section. In section 4, we give the performance evaluation of our system. Finally section 5 concludes the paper.

2. Related Work

There are two approaches in the literature that address anonymous service access. The first approach is called anonymous blacklisting (a.k.a anonymous revocation). This approach allows revocation of misbehaved users without revealing their real identities [9]. It also maintains previous anonymity for even abusive users. The second approach is called revocable anonymity [9]. In this approach, abusive users are revoked and their real identities are revealed as well.

In anonymous blacklisting, various Trusted Third Party (TTP) schemes have been proposed. These schemes assume a level of trust between parties. The first anonymous TTP blacklisting scheme to appear in the literature was proposed by Johnson *et al.* [19] and called Nymble [9]. Nymble constructs unlinkable authentication token sequences using hash chains. A pair of TTPs, the Nymble manager and the pseudonym manager, help service providers to link future tokens from abusive users so their access can be blocked. Unfortunately, these TTPs can easily collude to de-anonymize any user.

Nymbler [17, 19], BNymble [23], and Jack [22] are similar schemes that have been proposed with some performance enhancements on the base scheme Nymble. With an aim to force an agreement between users and service providers, Schwartz *et al*. [26] have proposed a contractual anonymity system. In this system, a user is de-anonymized if she breaches the contract with the service provider. This system still depends on a TTP.

BLacklistable Anonymous Credential (BLAC) [27], Enhanced Privacy ID (EPID) [5], Privacy-Enhanced Revocation with Efficient Authentication (PEREA) [28], and the second generation onion router TOR [11], are anonymous service access systems in which abusive users are revoked without contacting a TTP. In these schemes, service providers simply add authentication tokens associated with misuse to a blacklist [24].

Revocable anonymity systems (the second approach) generally depend on cryptography to generate and verify anonymous identities that are sometimes called pseudonyms [21].

The use of TTP to sign credentials and reveal real identities of pseudonyms was employed by many service providing systems such as Vehicular Ad hoc Networks (VANETs) described in [6, 7, 8, 13, 16]. In these systems, the authors propose the use of pseudonyms to access the service anonymously while maintaining the ability to revoke abusive pseudonyms by revealing their real identities.

Group signature schemes, such as [2, 3, 4, 20] have been widely used for both anonymous blacklisting and revocable anonymity systems. Based on group signature features, an open authority can revoke abusive users and may reveal their real identities [12], the author proposed a light weighted protocol for anonymous communication over Internet, where the cryptography overhead is distributed over sources. A similar work was proposed in [29].

All previous systems are either punitive in a way that they allow TTPs to reveal past and future anonymity of a particular user, or they are restrictive in a way that they allow revealing only current pseudonym. Each previously described pseudonym system fits to a particular service providing systems and may not fit to others. Hence, there is a necessity for a flexible system that maintains anonymity, distributes trust, and enables fair identity de-anonymization. In this paper, we propose a collusion resistant pseudonym providing system that addresses these issues, this work is an based on the model proposed by [1].

3. CORPPS Design

Figure 1 shows the latest version of CORPPS on which our system is built on, the system is fully described in [28]. Initially,

- *Step 1*: CoRPPS's initial setup is carried out. The aim of this stage is to prepare CoRPPS units for registering users and providing services to them.
- *Step 2*: After that, users register to the registration authority using their identification information (2).
- *Step 3*: Users apply to a predetermined number of authentication servers, *ASs*, to get tickets. These *tickets* are used by Pseudonym Signer, *PS*.
- *Step 4*: To generate signed pseudonyms.
- *Step 5*: Users use the service by authenticating themselves using their signed pseudonyms.

3.1. Assumptions and Threat Model

In CoRPPS design, we assume that all communications among CoRPPS entities are secured using Secure Socket Layer (SSL) or another transport layer security protocol.

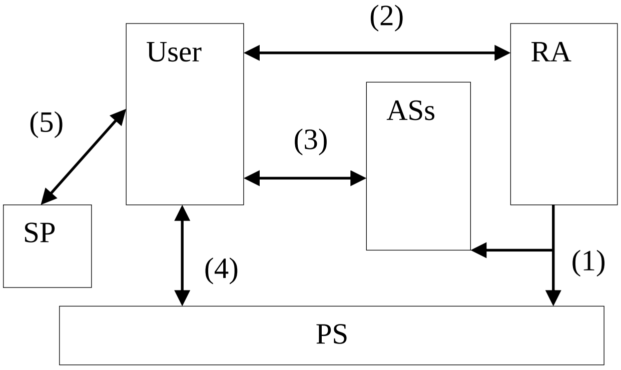

Figure 1. CoRPPS general design and flow.

Users are assumed to be semi-honest such that they follow the protocols properly and does not block the continuity of CoRPPS; however, they are curious and try to link pseudonyms to the real identities of particular users.

3.2. CoRPPS Basic Building Blocks

The basic building blocks of CoRPPS, namely tokens and token pool, counter, verification code, tickets, and pseudonyms. The following subsections summarises each of them.

3.2.1. Tokens and Token Pool

Tokens are temporary anonymous identifiers which help *PS* to verify that a user is a genuine user. Tokens are generated by the registration authority, *RA*, to be used by the authentication servers, *ASs*, to generate users' *tickets*.

3.2.2. Counter

In CoRPPS, a particular user is assigned a group of Authentications Servers, *ASs*, during registration and she always talk to this group of *ASs* to obtain *tickets*. A particular user *U* and her corresponding group of *ASs*, maintain a synchronized counter, Ctr_U. Ctr_U holds the number of times the user *U* has applied to *ASs* for *tickets* (i.e., it is something like a session counter).

3.2.3. Verification Code

Verification code, v, is a value calculated at each authentication server whenever a user applies for *tickets*. This code is unique for a particular user, *ID_U*, a particular group, *GID_U*, and a particular *Ctr_U* Each time user *U* applies to an *AS* for a *ticket*, *AS* calculates v as

$$v = \mathrm{hash}(ID_U||Ctr_U||GID_U\ \ )\, mod V_{max} \tag{1}$$

Where ID_U is the identity of the user, Ctr_U is the current counter value of user *U*, GID_U is the group identity of the *ASs* corresponding to that user, and V_{max} is the upper bound of v values.

3.2.4. Tickets

Tickets are pieces of encrypted information generated by authentication servers, *AS*s, and sent to the Pseudonyms Signer, *SP,* through the user.

3.2.5. Pseudonyms

Pseudonyms are temporary identities used by users to apply to the service provider for a service. Users generate *pseudonyms* as random values and send them to the Pseudonym Signer, PS, at which they are signed.

3.3. CoRPPS's Features

In this subsection, we entitle some extra features supported by CoRPPS. These features stem from the required characteristics and functionality that CoRPPS should provide as a pseudonym providing system. These include flexibility, identity revealing for liability and pseudonym revocation.

3.3.1. Flexibility

By flexibility, we mean the possibility of using CoRPPS as a general anonymous access system for different services. Since service providing systems vary according to the nature of the service, the following CoRPPS parameters can be tuned to fit into wider range of service providers:

1. Maximum number of pseudonyms allowed to be signed in each session NP_{max}.
2. The time period that unused pseudonyms are valid through, $V_{T\,period}$.
3. Once used, the lifetime of a pseudonym is restricted to VF_{period}.

The choice of the above mentioned parameters depends on the privacy threats and the required privacy level of a particular service providing system.

3.3.2. Identity Revealing for Liability

One of the main design criteria of CoRPPS is to achieve unlinkability between a pseudonym and the identity of its owner. However, law enforcement units may require to learn the identity of a pseudonym holder in case of a service abuse; a practical system should also support such an identity revealing for liability reasons. The process of revealing a real identity is carried out by collaboration among all CoRPPS trusted parties, *RA*, all *ASs*, *PS* and *SP*; the user entities do not take part in this process.

- *Step 1*: The service, for which the corresponding *pseudonym* is to be revealed, is sent to service provider, *SP*. *SP*, then, queries its database and returns the target *pseudonym*, P_U^i.
- *Step 2*: P_U^i is sent to the pseudonym signer, *PS*, which returns the corresponding combination of tokens and the verification code, t_{comb}, by searching its database.
- *Step 3*: t_{comb} is sent to all authentication servers, *AS*s, in the system. Each *AS* queries its database for the set of all user identities, *ID_Us*, to whom any combination of tokens and verification code, $(\mathbb{T}, v)$, was given. The result, S_j, of each AS_j's query is sent back to the identity revealing process. Actually, results from the group of *AS*s that took part in generation of P_U^i would suffice, but the pseudonyms, tokens and verification codes do not carry this information; thus, all *AS*s are needed to be queried. To finish
- *Step 4*: The identity revealing process takes the intersection S_js for each group of *AS*s (remember that *AS*s are grouped in the setup phase; each group has gASs and there are $\binom{N_{AS}}{g}$ groups). The intersection set for each group, denoted as $\mathbb{CU}_t$, is an empty set if the corresponding *AS* group has not been employed in the generation process of P_U^i. Finally, the union of all $\mathbb{CU}_t$ sets are calculated to find out candidate set of *ID_U*, denoted as $\mathbb{CU}$. The set $\mathbb{CU}$ is, actually, the set of the user IDs for which real identities are to be revealed by *RA*.
- *Step 4*: $\mathbb{CU}$ is sent to *RA*, which returns the real identity/identities of the user(s) in $\mathbb{CU}$, since *RA* keeps the ID_U–real identity mappings in its database.

3.3.3. Revocation

Revocation is the process of stopping to provide service to a user. There could be several reasons to revoke a user, which are out of scope of this paper. In this section, we describe how a user is revoked in CoRPPS. To revoke a user, all his signed pseudonyms should be blocked from accessing a service.

- *Step 1*: *RA* sends the identity of the user, ID_U, to be revoked to the group of authentication servers she is assigned to. These *AS*s, respond by sending back a set of all tickets issued for *ID*$_U$. These tickets are listed according to the order of their issuance.

- *Step 2*: *RA* groups each g tickets of the same order of issuance together, decrypts them, and generates a t_{comb} from each decrypted group of the g tickets. All t_{comb}s are then grouped in a set called $\mathbb{TC}_U$. $\mathbb{TC}_U$contains all t_{comb}s used by user U for applying to sign pseudonyms. *RA* then sends $\mathbb{TC}_U$to *SP* and asks her to find out all *pseudonyms* signed for all $t_{comb}s$ in $\mathbb{TC}_U$. *SP* lists these pseudonyms in $\mathbb{RP}_U$, which is the set of revoked pseudonyms signed for a particular user U.
- *Step 3*: *PS* sends$\mathbb{RP}_U$to theservice provider, *SP*. *SP* updates her revocation list accordingly. Each time a user applies to *SP* for a service, *SP* checks the user's pseudonym against the revocation list and then proceeds with the service if the provided pseudonym is not there.

3.4. Resistance Against Attacks

In this section, we describe the level of CoRPPS's resistance against collusion and data disclosure attacks mentioned in section 3.1.

3.4.1. Resistance Against Disclosure of Data

The basic idea of our design in CoRPPS is to prevent the ability of linking a pseudonym to a particular user and hence to a particular service. This means that a particular party must not be able to combine both service and identity information.

To summarize, in order to find out who has used a particular service, the chain of *pseudonyms→tickets→User identity→Real identity* must be followed and this is not possible without collusion of all trusted entities of CoRPPS. Partial collusions only cause partial problems but do not effectively reveal the real identity of a user who used particular service.

3.4.2. Resistance Against Collusions Among CoRPPS Entities

Collusion is defined as "a secret agreement between two or more parties for a fraudulent, illegal, or deceitful purpose". In our case, the attack of collusion among CoRPPS entities, *RA*, *AS*s, *PS* and *SP*, aims at revealing the real identity of a user who used a particular service. As described section 3.4.1, all of these entities must collude together (including all *AS*s) in order to reveal the real identity. On the other hand, it is also possible to have partial collusions, in which some - but not all-of the entities collude. Here, we examine different scenarios of partial collusions between system parties and explain the resistance level of CoRPPS against them.

The collusions between *RA-AS*s, *RA-PS*, *RA-SP* and *SP-AS*s do not cause any problems since these entity pairs do not have a common information-base to yield the real identity of a user who used a service.

Collusion between *SP* and *PS* yields the tickets used to obtain a pseudonym, which was used to access a service. Normally a particular *AS* does not know the other *AS*s in its groups. However, collusion among a subset of all *AS*s may cause to identify the groups of *gAS*s that issued the tickets of this pseudonym. This, in turn, causes to identify the *ID*$_U$ and then real identity with the help of *RA*. As the number of colluding *AS*s increase, the probability of the attack of linking pseudonyms to real identity increase. A detailed analysis of this collusion attack is given in section 4.2.

3.4.3. RA-AS-PS Trio Collusion

A corrupt *RA* may cooperate with a single *AS* and the *PS* to identify all pseudonyms by assigning the corrupt AS to each group during the setup phase. The corrupted *AS* then replaces $t||v$ in $E_K(t||v)$ with $HMAC(ID_U||Ctr_U)$ truncated or extended to the appropriate length. The *PS* can then recognize these identifiers and associate a pseudonym with a particular user. Since it is assumed that the encryption scheme is secure, no user will detect this attack.

Fortunately, this attack can be understood by legitimate *AS*s. The total number of *AS*s (*N*$_{AS}$) and the number of *AS*s per group (g) are publicly known. Then it is easy to infer the expected number of groups assigned to each *AS*. Therefore, *AS*s other than corrupted *AS* in the mentioned attack can easily discover this attack by the significance decrease in the number of groups they are assigned to.

3.4.4. Collusions Among CoRPPS Users

Another attack is the collusion among two or more users in order to escape from liability. Remember that one of the features of CoRPPS is that real identities can be revealed by law enforcement units for a liability issue. In order to smoothly run this process, a particular user should obtain her tickets from her designated group of *AS*s. In this attack, the cheater user exchanges some tickets with some other users and submits a mixed set of tickets to the *PS* to obtain signed pseudonyms. In this way, the cheating user seems to obtain tickets from some ASs other than her group of designated *AS*s. This situation causes the identity revealing process to fail and, therefore, the cheater cannot be tracked down by law enforcement. However, in order to succeed, the cheater should submit tickets of other users that can be verified by PS; this is not so possible, as discussed below.

For two colluding users, with known *ID*$_U$ and *Ctr*$_U$, it is not possible to calculate verification code precisely. This is because they do not know the group ID, *GID*$_U$, which is incorporated in the verification code calculation shown in Equation 1. Moreover, they cannot obtain v out of the tickets since the tickets are encrypted and the users do now know the encryption key *K*. However, it is still possible to exchange tickets and to have the same verification code with a probability of $1/V_{max}$, where $V_{\max}$ is the upper bound of the verification code values. As mentioned in Section 4, the typical value of V_{max} is 100; this means that the probability of successfully choosing a ticket of the same verification code is only 1%. Moreover, submitting another user's ticket to *PS* is a blind trial for the cheating user. The reason is that users exchanging the tickets cannot precisely determine that the exchanged tickets have the same verification code, because they do not know GID_U.

It is easy to discover such an attack by comparing the verification code of each ticket. Fortunately, it is also possible to identify the cheating users by careful selection of CoRPPS parameters. In a CoRPPS system of gASs in each group, the best chance for a successful attack is to use g-1 tickets having the same verification code (i.e., generated for the same ID_U, Ctr_U, and GID_U) and then try one ticket from another user. The g-1 tickets alone may then help in revealing the identity of abusive user if we design CoRPPS to have a very low collision probability for g-1*AS*s. The process of identifying abusive users is summarized below:

1. *PS* detects this attack by testing verification codes and storing ticket combinations involved in each trial.
2. If the numbers of trials exceed a threshold, PS reports *RA* with trials and ticket combinations.
3. *RA* then runs identity revealing process described in section 3.4.1 for all combinations of g-1 tickets in each trial.

4. Performance Evaluation

We provide detailed performance evaluation of CoRPPS in this section. It includes anonymity analysis, collusion analysis, and collision analysis. Both analytical and simulation results are given.

4.1. Anonymity Analysis

k-anonymity metric is widely used to describe the anonymity level, it refers to the state of being anonymous among another *k*-1 objects [15, 25]. A particular ID_U is k-anonymous at a particular *AS* if there exist other $k-1ID_U$s in *AS*'s records with the same ticket value.

The ticket reuse is the source of anonymity in each *AS*. If we consider a ticket reuse as a success, we can model ticket generation process as a binomial experiment. The total number of trials

$$n = N_U * Ctr_{max}, \tag{2}$$

and the probability of success

$$p = 1/(V_{max} * N_t). \tag{3}$$

The resulting binomial distribution is *B*(*n*,*p*).

We simulated the ticket generation process at a particular *AS*. Figure 2 shows the results of both the simulation experiment and the fitted binomial distribution.

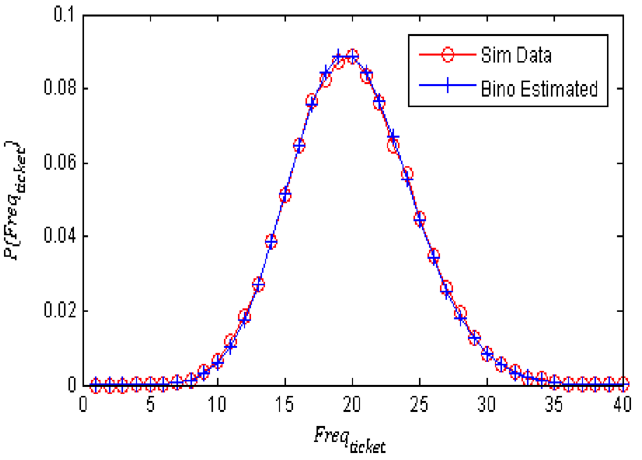

Figure 2. Analytical and simulation results of ticket generation process N_U=1000,Ctr_{max}=1000,N_t =1000, and $V_{\max}$=100).

Table 1 shows *k*-anonymity levels with *c*=0.999.

Table 1. Anonymity level with **c**=0.999,Ctr_{max}=1000,*Vmax* =100.

N_U	N_t	
	1000	**10000**
10000	71	2
50000	432	30
100000	904	71
250000	2347	203
500000	4783	432
1000000	9692	904

4.2. Analysis of Collusion Among *AS*s

*AS*s are grouped in groups of g members of total number of groups

$$g\_No = \binom{N_{AS}}{g} \tag{4}$$

Where N_{AS} is the number of *AS*s in the system, and g is the number of *AS*s in each group. Each user is assigned a particular group and should apply only to *AS*s of that group. As a result, if the data of *AS*s of a particular group is disclosed, then all tickets provided by those *AS*s to users of the same group are disclosed as well.

4.2.1. Arbitrary Group Disclosure with Respect to Disclosed ASs

The number of arbitrarily disclosed groups by collusion among *xAS*s is

$$N_{AGD}(x) = \binom{x}{g} \mid x \geq g \tag{5}$$

Where x is the number of colluding *AS*s, and g is the number of *AS*s in each group. Table 3 shows $N_{AGD}(x)$ for a system of N_{AS}=10 *AS*s and $g=$ 4*AS*s in each group. As the number of colluding *AS*s, *x*, increases, the number of disclosed groups, $N_{AGD}(x)$**,** also increases as shown in Table 2.

Table 2. Collusio5n among *AS*s.

x	$N_{AGD}(x)$
4	1
5	5
6	15
7	35
8	70
9	126
10	210

4.2.2. Probability of Particular Group Disclosure

Assuming that *x* represents the number of colluding *AS*s of a particular user's groups. The probability of finding the other *g-xAS*s of that group is shown in Table 3 and is calculated by:

$$P_{PGD}(g-x) = 1/\prod_{i=x}^{g-x}(N_{AS}-x) \tag{6}$$

Where *x* is the number of colluding *AS*s of a particular user's group, and *g* is the number of *ASs* in each group. Table 4 shows $P_{PGD}(g\text{-}x)$ for a system of N_{AS}=10*AS*s and *g* =4*AS*s in each group.

Table 3. Probability of revealing a particular group.

$g-x$	$P_{PGD}(g-x)$
4	0.00012
3	0.00198
2	0.01786
1	0.14286

4.2.3. Collision Analysis

The collision probability is calculated as

$$P_{collision} = \frac{EC}{N_{t_{comb}}} \tag{7}$$

Where *EC* is the expected number of previously used t_{comb}s in signing pseudonyms, and $N_{t_{comb}}$ is the total number of different t_{comb}s in the system.

Each time a user applies to *PS*, *EC* is incremented by one in the absence of collision, or remains the same if collision occurs. On this basis, *EC* is defined recursively in terms of the number of times, *n*, different users apply to *PS* for signing pseudonyms as:

$$EC(n) = EC(n-1) + 1 * \left(P_{no_{collision}}\right) \tag{8}$$

$$P_{no\_collision} = 1 - P_{collision} \tag{9}$$

$$= 1 - \frac{EC(n)}{N_{t_{comb}}} \tag{10}$$

$$EC(n) = EC(n-1) + 1 * \left(1 - \frac{EC(n)}{N_{t_{comb}}}\right) \tag{11}$$

$$EC(n) = x * (EC(n-1) + 1) \tag{12}$$

Where $x = \left(\frac{N_{t_{comb}}}{N_{t_{comb}}+1}\right)$

The later is a recurrence Equation with basis *EC*(1)=0. By solving this recurrence using repeated substitutions we get

$$EC(n) = x^{n-1} + x^{n-2} + \cdots + x \tag{13}$$

$$= (1 - x^{n-1}) \tag{14}$$

We have simulated the pseudonyms signing process, and calculated the collision probability for different values of users' counters, Ctr_U. The simulation and analytical collision probability results are shown in Figure 3.

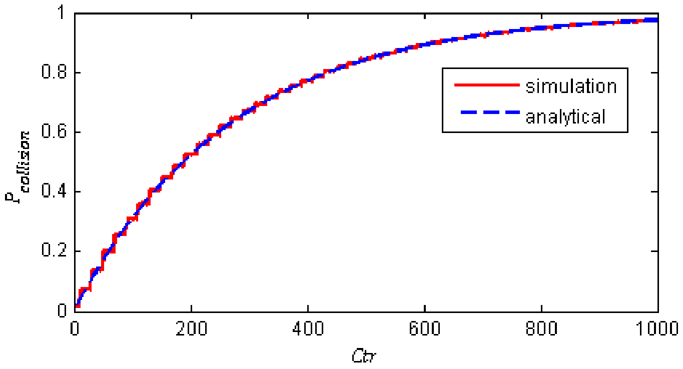

Figure 3. Simulation and analytical collision results N_t=200, V_{max}=20, N_U=100000, Ctr_{max}=1000, and *g*=3.

5. Conclusions

In this paper, we proposed a novel privacy-preserving pseudonym providing system, called Collusion Resistant Pseudonym Providing System (CoRPPS). In CoRPPS, several trusted entities are employed and the task of user authentication is split among several authentication servers. Other tasks and the corresponding user data are also split among trusted entities such that the collusion among them does not effectively link the real identity of a user to a pseudonym. This approach and the use of reusable tokens as anonymous identifiers in our design yielded high level of privacy for the users. The challenge of this design was that the link between the real user identities and pseudonyms should have been established by the request of law enforcement. In other words, there should have been a backdoor in the system, which contradicts the privacy requirements. We addressed this challenging issue in CoRPPS by enforcing all trusted parties to collaborate in the process of identity revealing. Analytical and simulation results showed that CoRPPS is applicable for different types of services; it can be tuned for different number of users according to anonymity level required, and the desired maximum collision probability. Our performance results also showed that CoRPPS is highly resistant against collusion attacks.

References

[1] Amro B., Levi A., and Syagin Y., "CoRPPS: Collusion Resistant Pseudonym Providing System," *in Proceedings of IEEE 3rd International Conference on Privacy, Security, Risk and Trust and IEEE 3rd International Conference on Social Computing*, Boston, pp. 1056-1063, 2011.

[2] Ateniese G., Camenisch J., Joye M., and Tsudik G., "A Practical and Provably Secure Coalition-Resistant Group Signature Scheme Advances in Cryptology-CRYPTO 2000," *in Proceedings of Annual International Cryptology Conference*, Santa Barbara, pp. 255-270, 2000.

[3] Ateniese G., Song D., and Tsudik G., "Quasi-Efficient Revocation of Group Signatures Financial Cryptography," *in Proceedings of International Conference on Financial Cryptography and Data Security*, Southampton, pp. 183-197, 2003.

[4] Boneh D. and Shacham H., "Group Signatures with Verifier-Local Revocation," *in Proceedings of the 11th ACM Conference on Computer and Communications Security*, Washington, pp. 168-177.

[5] Brickell E. and Li J., "Enhanced Privacy Id: A Direct Anonymous Attestation Scheme with Enhanced Revocation Capabilities," *in Proceedings of the ACM Workshop on Privacy in Electronic Society*, Alexandria, pp. 21-30, 2007.

[6] Buttyán L., Holczer T., and Vajda I., "On the Effectiveness of Changing Pseudonyms to Provide Location Privacy in VANETs Security and Privacy in Ad-Hoc and Sensor Networks," *in Proceedings of the 4th European Conference on Security and Privacy in Ad-Hoc and Sensor Networks*, Cambridge, pp. 129-141, 2007.

[7] Calandriello G., Papadimitratos P., Hubaux J., and Lioy A., "Efficient and Robust Pseudonymous Authentication in VANET," *in Proceedings of the 4th ACM International Workshop on Vehicular Ad Hoc Networks*, Montreal, pp. 19-27, 2007.

[8] Chaum D. and Evertse J., "A Secure and Privacy-Protecting Protocol for Transmitting Personal Information between Organizations," *in Proceedings of Conference on the Theory and Application of Cryptographic Techniques*, Santa Barbara, pp. 118-167, 1987.

[9] Chaum D., "Security without Identification: Transaction Systems to Make Big Brother Obsolete," *Communications of the ACM*, vol. 28, no. 10, pp. 1030-1044, 1985.

[10] Clauβ S. and Kohntopp M., "Identity Management and its Support of Multilateral Security," *Computer Networks*, vol. 37, no. 2, pp. 205-219, 2001.

[11] Dingledine R., Mathewson N., and Syverson P., "Tor: the Second-Generation Onion Router," *in Proceedings of the 13th Conference on USENIX Security Symposium*, San Diego, pp. 21-21, 2004.

[12] Eissa T. and Cho G., "Lightweight Anti-Censorship Online Network for Anonymity and Privacy in Middle Eastern Countries," *The International Arab Journal of Information Technology*, vol. 12, no. 6A, pp. 650-657, 2015.

[13] Fonseca E., Festag A., Baldessari R., and Aguiar R., "Support of Anonymity in VANETs-Putting Pseudonymity into Practice," *in Proceedings of IEEE Wireless Communications and Networking Conference*, Kowloon, pp. 3402-3407, 2007.

[14] Fox S., Rainie L., Horrigan J., Lenhart A., Spooner T., and Carter L., "Trust and Privacy Online: Why Americans Want to Rewrite the Rules," Pew Internet and American Life Project 2000.at http://www.pewinternet.org/~/media//Files/Reports/2000/PIP_Trust_Privacy_Report.pdf.pdf. Last Visited, 2011.

[15] Gedik B. and Liu L., "Protecting Location Privacy with Personalized K-Anonymity: Architecture and Algorithms," *IEEE Transactions on Mobile Computing*, vol. 7, no. 1, pp. 1-18, 2008.

[16] Gerlach M. and Guttler F., "Privacy in VANETs Using Changing Pseudonyms-Ideal and Real," *in Proceedings of IEEE 65th Vehicular Technology Conference-VTC2007-Spring*, Dublin, pp. 2521-2519, 2007.

[17] Henry R., Nymbler: Privacy-enhanced Protection from Abuses of Anonymity, MSc Thesis, University of Waterloo, 2010.

[18] IBM, "IBM Multinational Consumer Privacy Survey,"1999, at http://www.ibm.com/services/files/privacy_survey_oct991, Last Visited, 2003.

[19] Johnson P., Kapadia A., Tsang P., and Smith S., "Nymble: Anonymous IP-Address Blocking," *in Proceedings of the 7th International Conference on Privacy Enhancing Technologies*, Ottawa, pp. 113-133, 2007.

[20] Kiayias A., Tsiounis Y., and Yung M., "Traceable Signatures," *in Proceedings of International Conference on the Theory and Applications of Cryptographic Techniques*, Interlaken, pp. 571-589, 2004.

[21] Köpsell S., Wendolsky R., and Federrath H., "Revocable Anonymity Emerging Trends in Information and Communication Security," *International Conference on Emerging Trends in Information and Communication Security* Freiburg, pp. 206-220, 2006.

[22] Lin Z. and Hopper N., "Jack: Scalable Accumulator-Based Nymble System," *in Proceedings of the 9th Annual ACM Workshop on Privacy in the Electronic Society*, Chicago, pp. 53-62, 2010.

[23] Lofgren P. and Hopper N., "BNymble More Anonymous Blacklisting at Almost no Cost," *in Proceedings of the International Conference on Financial Cryptography and Data Security*, Gros Islet, pp. 268-275, 2011.

[24] Reiter M. and Rubin A., “Crowds: Anonymity for Web Transactions,” *ACM Transactions on Information and System Security*, vol. 1, no. 1, pp. 66-92, 1998.

[25] Samarati P., “Protecting Respondents' Identities in Microdata Release,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 13, no. 6, pp. 1010-1027, 2001.

[26] Schwartz E., Brumley D., and McCune J., “A Contractual Anonymity System,” *in Proceedings of the in Network and Distributed System Security Symposium*, San Diego, pp. 1-18, 2010.

[27] Tsang P., Au M., Kapadia A., and Smith S., “Blacklistable Anonymous Credentials: Blocking Misbehaving Users without TTPs,” *in Proceedings of the 14th ACM Conference on Computer and Communications Security*, Alexandria, pp. 72-81, 2007.

[28] Tsang P., Au M., Kapadia A., and Smith S., “PEREA: Towards Practical TTP-Free Revocation in Anonymous Authentication,” *in Proceedings of 15th ACM Conference on Computer and Communications Security*, Alexandria, pp. 333-343, 2005.

[29] Wu X., “Applying Pseudonimity for Anonymous Data Delivery in Location-Aware Mobile Ad Hoc Networks,” *IEEE Transactions on Vehicular Technology*, vol. 55, pp. 1062-1073, 2006.

Belal Amro is an assistant professor and the head of Computer Science Department at Hebron University - Palestine, where he has been working since 2003. Currently, he is conducting research in network security, wireless security, privacy preserving data mining techniques. From 2003 to 2004, he was a research assistant at Hebron University. From 2005 to 2007, he was an instructor in the Computer Science Department at Hebron University after having his MSc. degree in complexity and its interdisciplinary applications form Pavia- Italy. During 2008-2011 he received an ERASMUS PhD grant in Sabanci University-Turkey. From 2011-2012 he worked as research assistant at Sabanci University. In 2012, Belal received a PhD in Computer Science and Engineering from Sabanci University- Istanbul, turkey. He he has served as technical program committee member of different international conferences and journals, and reviewed more than 45 paper in the field of information technology including privacy and security.

Albert Levi received B.S., M.S. and Ph.D. degrees in Computer Engineering from Boğaziçi University, Istanbul, Turkey, in 1991, 1993 and 1999, respectively. He served as a visiting faculty member in the Department of Electrical and Computer Engineering, Oregon State University, OR, between 1999 and 2002. He was also a postdoctoral research associate in the Information Security Lab of the same department. Since 2002, Dr. Levi is a faculty member of Computer Science and Engineering in Sabanci University, Faculty of Engineering and Natural Sciences, Istanbul, Turkey. He has been promoted to associate professor in January 2008, and to full professor in May 2015. His research interests include computer and network security with emphasis on mobile and wireless system security, public key infrastructures (PKI), privacy, and application layer security protocols. Dr. Levi has served in the program committees of various international conferences. He also served as general and program co-chair of ISCIS 2006, general chair of SecureComm 2008, technical program co-chair of NTMS 2009, publicity chair of GameSec 2010, workshop chair of NTMS 2011 and general chair of NTMS 2012. He is editorial board member of The Computer Journal published by Oxford University Press, and of Computer Networks published by Elsevier.

Yucel Saygin is a Professor of Computer Science with the Faculty of Engineering and Natural Sciences at Sabanci University in Turkey. He received his B.S., M.S., and PhD. degrees from the Department of Computer Engineering at Bilkent University in 1994, 1996, and 2001, respectively. His main research interests include data mining, and privacy preserving data management. Yucel Saygn has published in international journals like ACM Transactions on Database Systems, VLDB Journal, IEEE Transactions on Knowledge and Data Engineering, IEEE Transactions on Engineering Management, and in proceedings of international conferences. He co-chaired various conferences and workshops in the area of data mining and privacy preserving data management. He was the coordinator of the MODAP (Mobility, Data Mining, and Privacy) project funded by EU FP7 under the Future and Emerging Technologies Program.