VERSITA

Annales UMCS Informatica AI XI, 3 (2011)
101–115; DOI: 10.2478/v10065-011-0020-9

Annales UMCS
Informatica
Lublin-Polonia
Sectio AI
http://www.annales.umcs.lublin.pl/

Group signature revocable anonymity scheme for network monitoring

Krystian Baniak*

Institute of Telecommunications, Warsaw University of Technology, Poland

Abstract

Subscriber's Privacy is in a constant conflict with security and accountability providing controls employed for network monitoring activities of service providers and enterprises. This paper presents the results of the author's research in the field of distributed network security monitoring architectures and the proposal of such a system that incorporates cryptographic protocols and a group signature scheme to deliver privacy protecting, network surveillance system architecture that provides subscriber's accountability and controlled, revocable anonymity.

1. Introduction

The internet has grown to become the major means of communication for economy, industry education, politics as well as for people. It is very important for the contemporary world but it also brings threats and risks that are exploited successfully by a new type of cyber criminals. The security monitoring is one of the essential means of control that allows security individuals to know its enemy and to counter security threats. Network security monitoring is one of vital elements that provides visibility and accountability for network owners or network providers.

*E-mail address: K.Baniak@elka.pw.edu.pl

In a nutshell a typical network monitoring system shall satisfy the following functional requirements:

- acquiring necessary information for operation and maintenance processes [**1**],
- measurement of traffic parameters for service level agreements or quality of service validation,
- controlling of communication services,
- providing security for network subscribers and network resources,
- providing input for security incident and event management systems [**2**].

Network traffic monitoring has, however, some serious implications on the subscriber's privacy and thus privacy-aware property is very important especially in the case of the Internet service providers and mobile incumbents. The authors of `PRIsm` framework [**1, 3**] were among the first to address this problem in the professional literature and they have also put forward appropriate standardization proposals.

This paper presents a proposal for the privacy mechanism based on the group signature scheme that drives a network security monitoring system called the MANSF[†]. The MANSF provides conditional anonymity for the monitored subjects and provides subject's accountability in the case of security incident.

2. Privacy Aware Network Monitoring Architecture

The MANSF (Multi-Agent Network Surveillance Framework) is designed for packet networks running Internet protocol suite and performs distributed passive network traffic analysis. The passive interception ensures that no alteration is imposed on the inspected network flows. Packet interception is performed in key network locations for maximal visibility and accountability (see Fig. 1). The targeted audience for this platform are Internet service providers, mobile operators, enterprise security and management teams or the security incident management organizations. The proposed platform is designed to satisfy security monitoring and network measurement goals at the same time with ensuring that no network subscriber experiences a privacy or anonymity degradation. Functional requirements with respect to the subscriber's privacy, have been satisfied by incorporating the following controls into the design:

- Layered communication architecture that limits interfaces available for a potential attacker.
- Pseudonyms representing subscribers in the central repository of network events.

[†]Multi–Agent Network Surveillance Framework

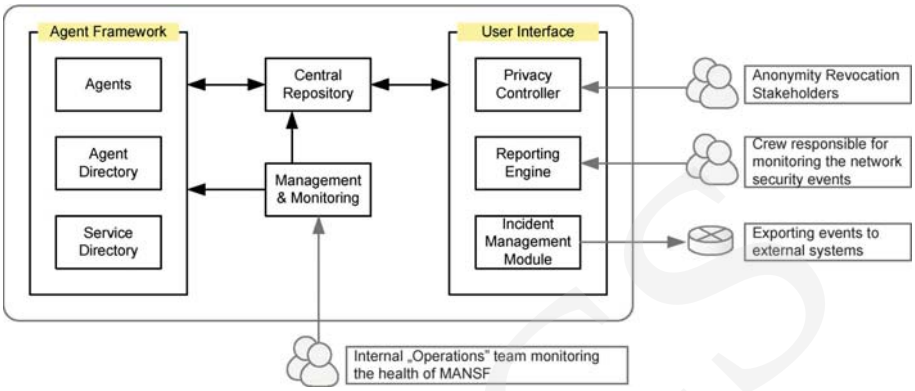

Fig. 1. Distributed network surveillance architecture.

- Dynamic group signature scheme that allows sensor agents to send trusted messages anonymously.
- Multilayer distributed data aggregation and normalization are used to enhance privacy.
- Revocation of anonymity is controlled by the cryptographic mechanism secured by a secret sharing scheme.

The key elements of architecture are the multi–agent framework, centralized data repository, management and monitoring and the user interface. The multi–agent framework consists of autonomous computer agents and supporting nodes used by agents for registration (Agent Directory) and as the repository of the topology (Service Directory). Agents are divided into two classes that process network traffic at different levels of abstraction:

- Agent Collector: network probe and network topology discovery function.
- Agent Processor: data aggregation and normalization agent that associates a group of agent collectors.

The Central Repository is a special node that collects network security events and evidence from collector agents. It also stores the knowledge in a form of frame system that contains network baseline profile and calculated subscriber profile classes. This knowledge may be further used for detecting anomalies or characterizing the observed traffic. Evidence data is anonymized and privacy protected. The Management and Monitoring element is used for system's operations management. The User Interface is hosting applications for system's end–users. The important example of such applications is the Privacy Controller that is the interface for subscribers' anonymity control and revocation.

The combined privacy protection controls provide the following properties for the proposed network surveillance system:

Anonymity: – subscribers retain their privacy as the monitoring system uses pseudonyms and normalized aggregated data.

Full Traceability: – it is possible to trace back an action to the unique subscriber.

Unlinkability: – no pseudonym can be linked to the real identity without the proper revocation procedure.

Exculpability: – it is not possible to attribute a given action to a false source, the revocation of an anonymity is exact and unique.

Unforgeability: – it is not possible to forge a notification attributing a false action to any source (admissible evidence property).

Revocability: – invertible pseudonymity used for the network monitoring purposes.

Most of those properties is achieved by using an efficient and dynamic group signature scheme that allows agents to revocably identify a source of network incident. The group signature scheme used in the MANSF is based on the robust and rigorously defined BSZ05 [**4**] scheme that has been extended by the author with a group member revocation procedure and by the group opening manager secret key control.

3. Revocable Anonymity Scheme

The key cryptographic primitive delivering anonymity properties which is the MASF architecture is a dynamic group signature scheme secure with the assumption of existence of trapdoor permutations based on the formal description and the rigorous security model BSZ05 first proposed in paper [**4**].

The set of procedures for this scheme consists of the following items: `Issue, Join, Judge, Open, Remove, Sign, Verify`, where `Remove` is an extension of BSZ05. The key requirement assumed for the group signature construct is the distribution of group manager roles into the separate modules realized by different physical servers within the MANSF. Additionally the set of group signature scheme procedures is partitioned into the public and protected classes thus limiting the number of the oracles available for a potential attacker in the internal or external perspective.

It is assumed that the Agent Collector is provisioned in the secure environment where it is not possible to retrieve the group signature private part of an agent. It is assumed that the `Join` and `Issue` procedures are executed in a trusted environment. The secrecy of the agent internal structures, which are retaining sensitive subscriber information, should be also protected by the

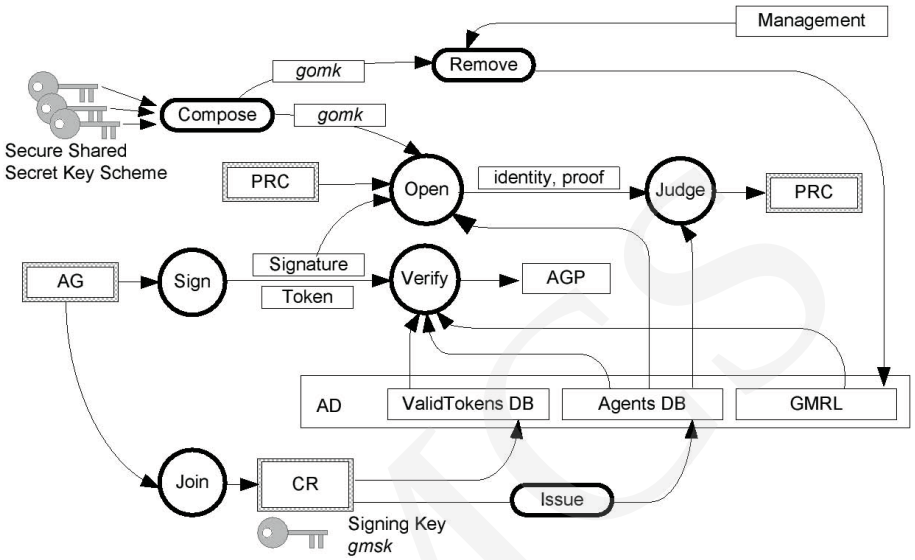

Fig. 2. Revocable Anonymity System's Architecture.

hardware means of protection. From the operations security perspective, the separation of duties and restricted console functionality are used to ensure that agent collector nodes are managed securely.

The key elements of the MANSF group signature framework are the following:

Group Manager: (GM) – that is responsible for the provisioning of group members and maintenance of the secret database of member certificates. The Group Manager has a *gmsk* key used for provisioning new members and implements group signature scheme procedures like `Join`, `Judge`, `Revoke` and `Verify`. It is located on the Central Repository and provides the following public services for the multi–agent framework:

- `Verify`, used by Agent Processors and the Central Repository to verify the authenticity of agent collector's messages,
- `Remove`, used to disable compromised or decommissioned agent collector,
- `Judge`, used by an user–plane application like the Privacy Controller, performing a revocation of subscriber's anonymity.

Group Opening Manager: (GOM) – implemented on the Agent Directory. GOM has a key *gomk* that is used to open a signature and reveal the identity of the signer. Provides `Open` procedure and hosts

the Group Member Revocation List (abbr. `GMRL`) that is used to verify whether the signature is issued by an authorized group member without disclosing the identity of the signer.

Group Controller: (GC) – implemented on the monitoring system of the MANSF. It is used to decomission an agent collector in the case of compromise. GC uses the Agent Directory for a reference to the list of agent collectors and uses the `Remove` procedure hosted on he Central Repository.

Group Member: – any agent collector agent within the MANSF multi-agent platform, implements the `Sign` procedure.

The Privacy Controller is not a part of the group signature scheme, but it plays an important role within the MANSF framework. It is an application layer module responsible for the evidence inspection and subscriber incident reporting. It may revoke the identity of a subscriber based on the decision of an operator and the authority responsible for privacy protection.

Agent collectors, the members of the group signature scheme, use the `Sign` procedure to authenticate messages broadcast toward their associated agent processor. In general, the signature is constructed over the digest of the exchanged message and the time stamp to record the time of sending a message.

Following the group signature construct proposed in [**4**], the result of the `Open` procedure may be verified with the second procedure `Judge` that is hosted on a separate system: the Central Repository. This solution is necessary to eliminate the scenario when one of the key group signature scheme members, like GM or GOM, is compromised. The Privacy Controller, aiming at revoking anonymity of a given subscriber, first follows the `Open` procedure and then checks the validity of results with the `Judge` procedure.

3.1. Scheme Details

Using the BSZ05 [**4**] proposal as the basis, let ρ_1 and ρ_2 are the NP relations over the domain D and $(P_1, V_1), (P_2, V_2)$ are the NIZK proofs for those relations and $k \in \mathbb{N}$ be the security parameter. Let $DS = (Sign, Ver)$ be a digital signature scheme, $ES = (Encrypt, Decrypt)$ be the public key encryption scheme and $H_k()$ is the k–bit message digest, as defined in the BSZ05. In addition to the base scheme cryptographic primitives this proposal extends it with the following items:

- public key infrastructure certificate authority on the Agent Directory with the public key K_{AD} and the private key K_{AD}^{-1} that issues certificates $cert_{name}$,

- secure the secret sharing scheme $SSEC = (Compose, n, k)$, where $Compose$ is the procedure that takes k key parts out of total n in order to derive a secret key value that is protected,
- GMRL list of revoked group members that contains the following records: (TK_i, T_r), where the token TK_i is the revocation token of an agent ag_i of index i, created during the `Join` procedure and T_r is the time when revocation happened.
- a message digest function $Hash$ used for revocation token integrity protection and satisfying the requirements of the PKI scheme.

The setup phase of the group signature scheme is identical with the base scheme proposal except for the initialization of the additional elements that extend the original scheme.

Procedure 3.1 Group Signature Scheme Setup

$R_1 \leftarrow \{0,1\}^{\rho_1}$; $R_2 \leftarrow \{0,1\}^{\rho_2}$; $R_3 \leftarrow \{0,1\}^k$;
$(pk_e, sk_e) \leftarrow K_e(1^k, r_e)$, r_e is a random value
$(pk_s, sk_s) \leftarrow K_s(1^k)$;
$gpk \leftarrow (1^k, R_1, R_2, R_3, pk_e, pk_s)$, group public key
$gmsk \leftarrow (sk_s)$, group manager key
$gomk \leftarrow SSEC(sk_e, r_e)$, GOM private key protected with the secret sharing scheme
$cert_{GPK} \leftarrow (gpk, T_c, E(K_{AD}^{-1} : Hash(gpk, T_c)))$, group manager certificate
$cert_{GMRL} \leftarrow (R_3, T_c, E(K_{AD}^{-1} : Hash(R_3, T_c)))$, GMRL certificate
$GMRL \leftarrow \emptyset$

The Group Manager and Group Member Revocation List certificates are issued and signed by the certificate authority that is implemented in the agent directory of the MANSF multi-agent framework. The public key of the Agent Directory PKI service does not belong to the group signature scheme. In detail, the agent verifying the group signature uses the `Verify` oracle located on the central repository and it has to validate the integrity of the *gpk* or the response from the GMRL service by inspecting an appropriate certificate. The mere certificate's validity is checked using the self-signed certificate of the Certification Authority of the MANSF platform. This certificate is also provided by the agent directory node.

Table 1 outlines the cryptographic attributes used by different types of group signature scheme members. The square bracket denotes a variable of type array and $table[i]$ denotes the i-th element of that array.

Table 1. Data structures used by members of the group signature scheme.

Role	**Data Structures**
Agent Directory	$Agents[i] = (pk_i, sig_i, cert_i, agc_{id}, T_c, rsig_i)$ $GMRL[i] = (i, T_c, T_r, TK_i)$ $ValidTokens[i] = Hash(TK_i)$ $gomk$
Central Repository	$(gmsk, pk_s)$
Agent Collector	$(pk_i, sk_i, cert_i, TK_i)$

The notation used in the cryptographic attributes descriptions is as follows:

(pk_i, sk_i) : public and private key of a given agent collector instance
$cert_i$: certificate of agent i generated during the provisioning
agc_{id} : agent collector's id, reference to the Agent Collector table
T_c : table entry creation time stamp
T_r : revocation entry creation time stamp $entry.time\_revoked$
RT_i : revocation token generated for a group member in `Join`
TK_i : revocation token generated for a group member in `Sign`
$Hash(TK_i)$ digest of a revocation token
$rsig_i$: revocation token signed by the agent i
pk_e : public key of the group opening manager
(sk_e, r_e) private key of the group opening manager

The exchange of messages in the join procedure is done only between applications during the provisioning procedure and it is not traversing the multi-agent communication network. The secret key generation procedure is according to the BSZ05 model assumed as trusted. The adversary cannot see the result of generation procedure. The appropriate security controls have to be deployed to satisfy this functional requirement.

The signature generation, the `Sign` procedure, is used by an agent collector instance to prove authenticity of the anonymous message it is distributing. An agent collector sends also its revocation token R_{GMRL} to claim its revocation status.

Verification of the revocation status for the group signature's owner is implemented by the `GMRL` procedure. The Group Member Revocation List (GMRL) is the database of all revoked agents which is used to check whether the signer has a right to sign in a given point of time. The time stamp is used to record

Procedure 3.2 Join & Issue Group Procedure

The agent collector generates a key pair $(pk_i, sk_i) \leftarrow K_s(1^k)$ and signs it to produce $sig_i \leftarrow E(sk_i : pk_i)$. It also creates a revocation token $RT_i \leftarrow (R_3, T_a)$ and creates its signature $rsig_i \leftarrow E(sk_i : R_3, T_a)$. Both items are sent to the Central Repository that is the Group Manager:

$$AG_i \rightarrow CR: \quad E(K_{CR} : (pk_i, sig_i, RT_i, rsig_i), T_a, N_a, K_r)$$

The Agent Directory verifies the signatures sig_i and sig_R before continuing the procedure. When signatures match it generates the certificate $cert_i \leftarrow Sign(sk_s : (i, pk_i))$ and formulates the response [**4**]. The response from the central repository agent is encrypted with the challenge K_r proposed by the agent collector:

$$CR \rightarrow AG_i: \quad E(K_r : (i, pk_i, sig_i, cert_i), N_a, T_c, E(K_{CR}^{-1} : Hash((i, pk_i, sig_i, cert_i), N_a, T_c)))$$
$$CR \rightarrow AD: \quad E(K_{CD} : agent = (i, pk_i, sig_i, cert_i, agc_{id}, T_c, TK_i), E(K_{CR}^{-1} : H(agent)))$$

The Agent Directory receives the new agent collector registration information and populates $Agents[\ldots]$ the table with the new entry. Additionally, the list of valid tokens $ValidTokens[\ldots]$ is appended with the digest of the revocation token. The variable TK_i is discarded after making the digest out of it. The revocation token is protected by the key *gomk* and secure secret sharing scheme in order to limit the possibility in using the token for traffic analysis by the corrupted Agent Directory:

$(R_3, T_a) \leftarrow RT_i$
$TK_i \leftarrow Encrypt(pk_e : R_3, T_a)$
$Agents[i] \leftarrow (i, pk_i, sig_i, cert_i, agc_{id}, T_c, rsig_i)$
$ValidTokens[\ldots] \leftarrow Hash(TK_i)$

Procedure 3.3 Sign procedure $GSig(gpk, m)$

$H_k(m)$ is a k-bit message digest function run on the original message m being signed.

$TK_i \leftarrow Encrypt(pk_e : R_3, T_a)$
$R_{GMRL} \leftarrow E(K_{AD} : TK_i, T_s, N_a)$
$h_m \leftarrow H_k(m, R_{GMRL}, T_s)$
$sgn \leftarrow Sign(sk_i : h_m); \quad r \leftarrow \{0,1\}^k$
$c \leftarrow Encrypt(pk_e : (i, pk_i, cert_i, sgn), r)$
$\pi_1 \leftarrow P_1(R_1, (pk_e, pk_s, h_m, c), (i, pk_i, cert_i, s, r))$
return $(\pi_1, c, T_s, R_{GMRL})$

a time since when the signatures issued by a given member are treated as non trusted. In general, the GMRL procedure is a service implemented on the agent directory node. The agent directory is also the owner of the Open procedure. The revocation check procedure takes a revocation token R_{GMRL} as the parameter and the time T_s to verify that it is consistent with the time stamp embedded in the encrypted token. The token is always sent in an encrypted form protected with random nonces and a time stamp to ensure that given transmitted token is always fresh for a particular agent. This technique is used to protect anonymity of the group signature member and security of the whole scheme. This procedure also verifies that the revocation token value belongs to the valid group member by consulting the $ValidTokens[\ldots]$ table.

Procedure 3.4 Revocation status check procedure $GMRL(gpk, T_s, R_{GMRL}))$

This procedure uses the object notation for the entries of the GMRL CRL table.

$(TK_i, T_a, N_a) \leftarrow E(K_{AD}^{-1} : R_{GMRL})$
If $T_a \neq T_s$ return false
Unless $ValidTokens.has(Hash(TK_i))$ then return false
Foreach *entry* in $GMRL[]$ do
  If $entry.token = TK_i$ then
    If $T_a >= entry.time\_revoked$ then
      return false
    End
  End
End
return true

The `Open` procedure is hosted on the Agent Directory of the MANSF platform. This procedure checks whether the signature is correct and then returns the set of identifiers pointing to the Agent table along with the NIZK proof verifiable by the Judge procedure. It is necessary to retrieve the secret key sk_e that is protected by the secure secret sharing scheme $SSEC(N, K)$, where K out of N part–key holders are required to commit the procedure.

`Judge` procedure is used for validation of opening manager output. Opening manager uses NIZK proof over ρ_2 to prove the knowledge of his signing key material. This is mandatory procedure in case there is a possibility of disgruntled opening manager. This concept has been introduced in [**5, 4**].

`Remove` procedure is, in fact, applicable for disabling an agent collector instance and populating the GMRL database with the reference to the revoked agent. The agent entry from the Agents[] table is never erased. This is the requirement that allows identifying the source of evidence event even after the

Procedure 3.5 Open procedure $Open(gpk, gomk, m, \pi_1, c, T_s, R_{GMRL})$

$h_m \leftarrow H_k(m, T_s, R_{GMRL})$
$sk_e \leftarrow Compose(gomk)$
$(i, pk, cert, s) \leftarrow Decrypt(sk_e : c)$
If $Agents[i] \neq NULL$ or $pk_i = pk$ then
  $(pk_i, sig_i) \leftarrow Agents[i]$
Else return false
If $V_1(R_1, (pk_e, pk_s, h_m, c), \pi_1) = 0$ then return false
$\pi_2 \leftarrow P_2(R_2, (pk_e, c, i, pk, cert, s), (sk_e, r_e))$
return $(i, \pi_2, pk_i, sig_i, cert, c, s)$

Procedure 3.6 Judge procedure $Judge(gpk, c, i, pk, cert, s)$

$(pk_i, sig_i) \leftarrow Agents[i]$
If $pk_i \neq pk$ then return false
If $V_2(R_2, (pk_e, c, i, pk, cert, s), \pi_2) = 0$ then return false
If $Ver(pk : sig) \neq pk_i$ then return false
return true

agent collector is disabled. The entry in the GMRL table consists of two time stamps, where T_c denotes the entry creation time and the T_r is the time since when the agent has been regarded as revoked. The T_r time stamp allows an operator to decide whether formerly issued signatures are also regarded as non trusted. This kind of functionality allows for more granularity in handling compromised agent collectors.

Procedure 3.7 Remove procedure $Remove(gomk, i, T_r)$

CR receives a command to disable agent i from the management station. CR fetches the revocation token from the agent directory agents database $Agents[i]$ and new entry is inserted into GMRL table. The token is protected with the *gomk* so it has to be decrypted first. It is necessary to retrieve the secret key sk_e that is protected by the secure secret sharing scheme $SSEC(N, K)$, where K out of N part–key holders are required to commit the procedure.:

$T_c \leftarrow Time.now()$
$sk_e \leftarrow Compose(gomk)$
$TK_i \leftarrow Decrypt(sk_e : Agents[i].TK_i)$
$entry \leftarrow (TK_i, Tc, T_r)$
$GMRL[\ldots] \leftarrow entry$

One comment is necessary for the Remove procedure. It becomes evident that the successful revocation of subscribers identity in MANSF is related to

the availability of the historic or archive data from agent collectors. In order to provide the accountability of subscribers an appropriate data retention policy has to be in place. This entails the retention of regular backups of given agent collector database of subscribers and pseudonym maps. Those backups have to be stored outside the multi-agent platform and have to be encrypted with the secret key also protected with additional means.

The `Verify` a procedure is used in order to validate the authenticity and correctness of a group signature. This procedure is implemented on the central repository and on agent processor agents. The verification is implemented as the zero knowledge proof check issued by a group member signing the message m. If the result of the NIZK proof is positive than it is confirmed that the message, the signing time and the revocation token are issued by a valid group member. The final check consists of verification of the revocation status of the signer using the revocation token R_{GMRL}.

Procedure 3.8 Verify procedure $Verify(gpk, m, \pi_1, c, T_s, R_{GMRL})$

$H_k(m)$ is a k-bit message digest function run on the original message m that has been signed by a group member.

$(R_1, pk_e, pk_s) \leftarrow gpk$
$h_m \leftarrow H_k(m, R_{GMRL}, T_s)$
If $V_1(R_1, (pk_e, pk_s, h_m, c), \pi_1)$ then
  return $GMRL(gpk, T_s, R_{GMRL})$
Else return false

4. Proposal's Security

The group signature scheme relies on the BSZ05 [**4**] model that, under the assumption of trapdoor permutation's existence, provides correctness, anonymity, non–frameability and traceability and delivers a dynamic signature scheme. The cryptographic primitives implied by this scheme are very complicated and inefficient and thus the mere scheme is not practical. However, the scheme offers a rigorous and sound formal structure that is a good basis as the reference model.

The BSZ05 scheme also introduces separate roles for the group manager and opening manager which enhances security by reducing frameability threats. From the anonymity perspective the scheme ensures that the group signature cannot be forged which implies the accountability for a group member issuing a signature.

Group Member Cadence Security Impact. The dynamics of the BSZ05 scheme is delivered in a sense of flexible expanding of the number of group

members with the use of the Join procedure. The group public key is not dependent on the group size as the signature is based on the non-interactive zero knowledge proofs. The BSZ05 scheme is extended with the revocation mechanism based on the GMRL list (Group Member Revocation List) that is available via the GMRL oracle to the public requesters without a threat for the group signature scheme members' anonymity. The GMRL oracle returns *true* or *false* and the only advantage of the party that verifies the group signature is the knowledge of the fact that it is issued by a group member during his cadence. If we assume that GMRL Oracle is only available for the Verify Oracle that we further limit the knowledge acquired by a potential adversary. In the worst case the GMRL oracle will not help in reducing the anonymity of the group member unless the attacker knows the list of revoked agents and thus the anonymity set can be significantly reduced. In the case if there is only one revoked member, the identity of the agent may be broken. The corruption of the Agent Directory allows the adversary to obtain access to the $Agents[]$ table and to the $GMRL[]$ table and is able to retrieve the group manager opening key. Taking into account the fact that this key is protected with the secure secret sharing scheme this type of event requires stakeholders to collude. As an additional security control, the public key infrastructure is used to protect integrity and deliver non–repudiation for the group signature message exchange, like in the case of the GRML procedure.

The signer cadence check, has to have a minimal impact on the information that may be leaked during the verification process. Therefore the GRML list contains only the date and time when an agent has been `Removed`. The lower bound check is realized as implementation security control in a way where every message is checked for the time stamp window. When arrival time and creation time stamps are too distant from the received time stamp that the message has to be discarded as invalid. Of course, this area may easily become a weakness if the protocol is not maintained properly.

Architectural Strengths.From the architectural point of view, the security level is further enhanced with the use of the following concepts related to the specifics of the MANSF platform:

- **Communication platform layering** – different procedures are invoked over the separate network planes like for instance the `Remove` procedure can only be done by the management platform whereas the `Verify` procedure may be invoked by any agent processor agent or the central repository agent.
- **Protected group signature procedures** – `Open` and `Remove` oracles are not available for the attacker in the adversary model of the

MANSF. Also the GMRL contents are private from the agent directory perspective.

The adversary model assumes that the `Remove` oracle is not available for the multi-agent members. Also the GMRL oracle is only available for the Agent Directory agent. In the case the Agent Directory is compromised, in normal conditions when the access to the group opening secret key is not protected, the anonymity would be broken. In our case the secure secret sharing scheme removes this weakness.

Weaknesses. Potential weaknesses are concentrated around the corruption of individual group signature elements. First, the agent collector which is the group signature scheme member, is a weak point in the case the adversary hijacks the secret keys used to generate signatures. In such a case a group member may forge messages until the fact of corruption is detected. The second weak point is the agent directory that is the group opening manager which hosts GMRL. The potential adversary may try to get the number of revoked members, however, it is not possible to deanonymize them without colluding with the secure secret sharing scheme's stakeholders.

Efficiency. The original formal BSZ05 model relies on the very complicated and CPU–intensive cryptographic primitives. For instance the GMR digital signature scheme that is claimed to be secure under chosen ciphertext attack (CCA–secure) [**4**], has in its enhanced form [**6**] the computation cost comparable to the RSA ($O(\log(N))$) scheme. This, however, may not be efficient for an agent analyzing the intensive traffic and which has to produce signatures for thousands of generated messages. Fortunately, the latest advancements in the field of pairing based cryptography allow to compose schemes that have constant size group public keys and short group signatures. The most efficient schemes that use pairing based cryptography like [**7, 8, 9**], offer signatures as short as 6–8 group elements of a 520–bit prime order group constructed using an elliptic curve over a finite field.

Implementation. In the research work, the selection of the group signature scheme was dictated by the practical application requirements such as computation cost and the size of the signature domain. The choice was the scheme using a bilinear map over the prime number elliptic curve finite field. The basis for the implementation is a version of the GRO07 group signature scheme [**8**], which is anonymous under the chosen plain text attack (CPA–anonymous). The implementation is reinforced with the "PBC library" and the "PBC signature" libraries that implement the platform for bilinear maps generation and are easily extensible. The core cryptographic primitive used by [**8**] and in the implementation of MANSF group signature is the BB04 [**10**] short signature.

5. Conclusions

In conclusion, many existing dynamic group signature schemes may be extended with a fully dynamic option by adoption of architecture related or external techniques like public key cryptography and revocational lists maintained by one of the group managers. The scheme presented in the MANSF frameworks also implements this principle with success. The monitoring system is mostly concerned with maximal subscriber privacy and a verifiable evidence source. Signature cadence check of the MANSF, realized with the revocation list based on revocation tokens and protected with secure shared secret scheme introduces minimal impact on the privacy and anonymity of event source. Further research is, however, needed for ensuring resistance to GMRL Oracle corruption and ability for an adversary to infer on agent's identity knowing the date and time of given agent revocation.

References

[1] Bianchi G., Boschi E., Kaklamani D.I., Koutsoloukas E.A., Lioudakis G.V., Oppedisano F., Petraschek M., Ricciato F., and Schmoll C., Towards privacy-preserving network monitoring: Issues and challenges, Personal, Indoor and Mobile Radio Communications - PIMRC'07 (2007): 1.

[2] Xu K., Zhang Z. L., and Bhattacharyya S., Internet traffic behavior profiling for network security monitoring, IEEE/ACM 16(6) (2008): 1241.

[3] Gogoulos F., Antonakopoulou A., Mousas A. S., Lioudakis G. V., Kaklamani D. I., and Venieris I. S., Privacy-aware passive network monitoring, Panhellenic Conference on Informatics 0 (2009): 171.

[4] Bellare M., Shi H., and Zhang C., Foundations of group signatures: The case of dynamic groups, Topics in Cryptology - CT-RSA '05, Lecture Notes in Computer Science 3376 (2005): 136.

[5] Bellare M., Micciancio D., and Warinschi B., Foundations of group signatures: Formal definitions, simplifieed requirements, and a construction based on general assumptions, Advances in Cryptology - EUROCRYPT '03, Lecture Notes in Computer Science 2656 (2003): 644.

[6] Goldreich O., Two remarks concerning the goldwasser-micali-rivest signature scheme, Proceedings of CRYPTO '86, London, UK, Springer–Verlag (1987): 104.

[7] Boyen X. and Waters B., Full-domain subgroup hiding and constant-size group signatures,Public Key Cryptography – PKC '07, Lecture Notes in Computer Science 4450 (2007): 1; Available at http://www.cs.stanford.edu/∼xb/pkc07/.

[8] Groth J., Fully anonymous group signatures without random oracles, Proceedings of ASIACRYPT '07, Berlin, Heidelberg, Springer–Verlag (2007): 164.

[9] Liang X., Cao Z., Shaoand J., and Lin H., Short group signature without random oracles, 9th International Conference on Information and Communications Security – ICICS '07, Berlin, Heidelberg, Springer–Verlag (2007): 69.

[10] Boneh D. and Boyen X., Short signatures without random oracles, Proceedings of EUROCRYPT '04, Lecture Notes in Computer Sciences 3027 (2004): 56.