

Requirements of Information Reductions for Cooperating Intrusion Detection Agents^{*}

Ulrich Flegel and Joachim Biskup

University of Dortmund, D-44221 Dortmund, Germany
{flegel, biskup}@ls6.cs.uni-dortmund.de

Abstract. We consider cooperating intrusion detection agents that limit the cooperation information flow with a focus on privacy and confidentiality. Generalizing our previous work on privacy respecting intrusion detection for centralized systems we propose an extended functional model for information reductions that is used for cooperation between intrusion detection agents. The reductions have the following goals: detective effectiveness of cooperation alliances, privacy of honest individuals, further organizational confidentiality requirements, and efficiency. For the reductions we outline the basic requirements, and derive the specific requirements imposed by the cooperation methods used for intrusion detection. It is shown, how our existing solutions could be adapted and what restrictions apply.

1 Introduction

When designing IT systems we not only need to take the security requirements of the providers into perspective, but also the security requirements of the users. Both, users and IT system providers are interested in the dependability, in particular the integrity and availability of the IT system. In the recent years it has been recognized that preventive safeguards need to be complemented by reactive aspects of security.

A security incident comprises the violation of the given security policy. Reacting to security incidents requires detecting them in the first place. To be able to detect violations of the security policy, one must be able to observe all activity that could potentially be part of such violations. Modern services and operating systems either already supply mechanisms for observation or can be instrumented appropriately. The observed information is denoted as *audit data* in the following. The audit data can be analyzed by an *intrusion detection system* (IDS) in order to detect security incidents. If an IDS detects a security incident, appropriate reaction should be initiated. A reaction may require to hold a user accountable for the damage caused. Therefore, audit data usually provides information to account activity to persons.

Since most audit data can be used without much effort to identify individual users, recording and sharing such data may conflict with the users' expectancy for privacy and with pertinent legislation concerning the personal data of users. In Sect. 2 we summarize existing solutions to solve the conflict between the need for audit data for intrusion

^{*} This work has been partially funded by the German Research Council (DFG) under grant number Bi 311/10-3.

detection and the privacy requirements. The core idea is to replace personal data in audit data with carefully chosen pseudonyms. Current technology for pseudonymizing audit data is applicable to centralized intrusion detection systems, only. In Sect. 3 we argue, that centralized intrusion detection will not be sufficient in the future. Rather, intrusion detection agents need to cooperate to sustain adequate detection facilities. Recent work on cooperating intrusion detection agents is summarized in Sect. 4.

In Sect. 5 we generalize prior concepts of pseudonymization and propose an extended functional model for information reductions. Information reductions are a prerequisite for cooperating with partially trusted agents that should not learn certain information. The basic requirements of such information reductions are outlined in Sect. 6, whereas requirements for the specific cooperation methods used by intrusion detection agents are derived in Sect. 7 and Sect. 8. The derived results are related to audit data pseudonymization and according adaptations of prior solutions are proposed. The paper discusses related work and concludes in Sect. 9. The main contribution is fivefold:

- proposing a novel *functional model for information reductions* that is used for cooperation between intrusion detection agents,
- identifying the *basic requirements* of such information reductions,
- deriving *specific requirements* based on existing work on cooperation of intrusion detection agents,
- proposing according *adaptations of existing solutions* for audit data pseudonymization, as well as identifying the *inherent limitations*, and
- identifying the major *challenges wrt. intrusion detection, inference control and cryptography* in order to achieve secure and useful information reductions for cooperating intrusion detection agents.

2 Pseudonymization for Centralized Intrusion Detection

We proposed concepts for the pseudonymization of audit data for intrusion detection while balancing the conflicting requirements for accountability and anonymity [3]. In our approach Unix *syslog audit data* is pseudonymized by a *source agent* immediately after it has been generated by an *information source*, such that users appear under pseudonyms in the audit data (see Fig. 1). The *audit data with pseudonyms* maintains the degree of linkability required for intrusion detection by the analyzing agent. The pseudonymization process also produces additional data that allows for the recovery of the original data, subject to specific conditions. The audit data with pseudonyms and the *pseudonym recovery data* are forwarded by the source agent to the analyzing agent.

The *analyzing agent* normally works in a *surveillance mode*, where merely the audit data with pseudonyms is analyzed with respect to misuse suspicions. Only if a (*threshold*) *alert* occurs, i.e., a misuse suspicion has been detected, the pseudonym recovery data can be used for *reidentification*, i.e., the original audit data can be reconstructed. In the *alert mode* the analyzing agent can employ the *reconstructed audit data* to establish accountability for legal purposes, such as damage prevention and litigation.

For the pseudonym recovery data, the approach leverages Shamir's threshold scheme for cryptographic secret sharing [4]: The misuse suspicions for intrusion detection are modeled as thresholds of secret sharing schemes. The pseudonym recovery data contains the encrypted identifying data that is replaced by the pseudonyms, and it contains

shares of the respective decryption keys. As a result, the disclosure of the encrypted identifying data is enforced cryptographically, such that decryption is possible if and only if the pseudonyms are involved in a sufficient suspicion of misuse (*technical purpose binding*), i.e., the number of shares associated with the pseudonyms exceeds the threshold in the model of the misuse suspicion. Note that it may be necessary to provide the ability to recover the decryption keys independently of a priori defined models of misuse suspicion in order to investigate misuse that has not (yet) been modeled. In that case, the grounds for decryption must be scrutinized by one or more trusted parties (*organizational purpose binding*). Involving these parties can be enforced cryptographically, e.g. using threshold cryptosystems [1].

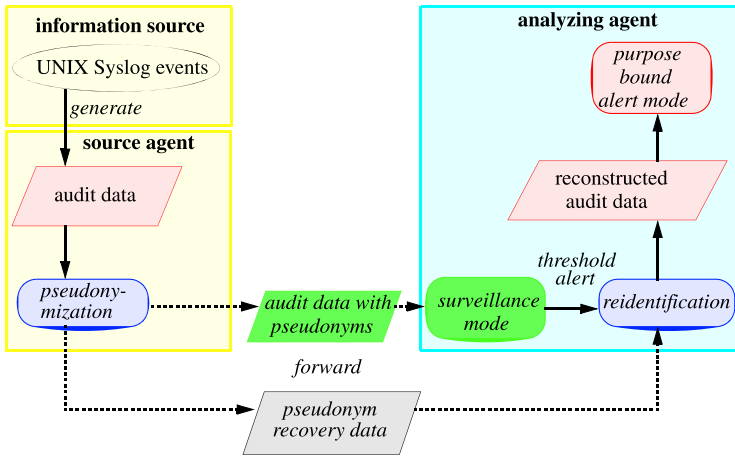


Fig. 1. Functional model for pseudonymization

For the pseudonymization approach we have implemented a suitable system design and described practical aspects of its use [2]. The evaluation has shown, that the concepts are viable in practice to handle real-world audit data volumes [2].

3 The Need for Cooperation

The approach described in Sect. 2 is suitable for centralized intrusion detection, where information sources and source agents are controlled by a single authority, such that pseudonyms and recovery data are generated consistently with respect to a central analyzing agent. We expect to see the following main problem in the future. Complex and orchestrated attacks increasingly span larger and more heterogeneous networks, and they cross organizational boundaries. From the attacker’s point of view this is a logical step to conceal his attacks by breaking it down into smaller parts, which might seem innocuous, if observed isolatedly. Also by using several distributed machines an attacker can gain firepower for denial of service attacks. Complex attacks of this kind cannot be monitored by local source agents of one intrusion detection system alone. Conventional

centralized approaches for intrusion detection will neither be able to collect sufficient information about complex attacks nor to support an appropriate response. Therefore, cooperation between distributed IDS agents is required.

The main problem comes along with further weaknesses of centralized systems. (1) The increasing capacity of networks and computers results in an ever increasing audit-data volume that needs to be analyzed. In the face of this development, centralized intrusion detection approaches do not scale adequately any more. (2) A single centralized analyzing agent is a single point of failure and cannot constitute a robust solution. (3) Complex attacks implement strategies that comprise several elementary sub-attacks. Conventional intrusion detection systems merely focus on the detection of the elementary attacks. As a consequence of a complex attack, security administrators are confronted with a multitude of alarms that only together describe the complex attack on a low level of abstraction. The recognition of attack strategies and their overall goals is left to the security administrators.

4 On Cooperating Intrusion Detection Agents

Cooperative and distributed intrusion detection agents are proposed as the solution for the sketched problems. There has been some development primarily focusing on the technical and practical issues that enable distributed intrusion detection to face problems (1) and (2). In order to move from centralized solutions to distributed IDS, algorithms for distributed audit data analysis have been proposed [5, 6, 7]. Recently, research also extends on improving the results of distributed analysis, in order to solve problem (3).

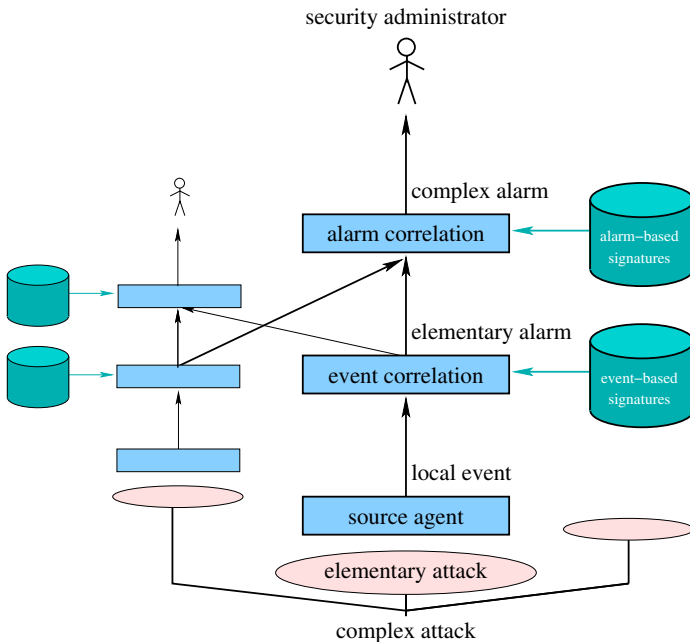


Fig. 2. Example with alarm correlation

Seminal proposals of Huang and Bass [8, 9] deal with the analysis of strategies and goals of attackers, utilizing results from the field of multisensor data fusion. Fusion or aggregation of alarms of elementary attacks that have been generated by different analyzing agents result in the reduction of the alarm volume, thereby relieving the security administrators. To do this, knowledge-based rule systems and probabilistic methods have been proposed [10, 11, 12, 13, 14, 7]. These approaches use simple heuristics based on domain knowledge about invariants of certain classes of attacks.

Knowledge-based correlation of alarms of elementary attacks can be used to recognize complex attacks in detail (see Fig. 2). The correlation can be conducted using explicit models such as signatures interpreting alarms as events, where the models can be specified by experts [15, 10, 16, 17] or learned from labeled training data based on similar heuristics as used for alarm aggregation [18]. Alarm correlation based on explicit models fails to recognize not explicitly specified variants of complex attacks. Specifying instead the pre and post conditions of alarms, all (variants of) complex attacks can be recognized wrt. these conditions [19, 20, 21, 14].

Correctly abstracting many elementary attacks into a few complex attacks has several advantages. Firstly, the alarm volume is reduced. Secondly, strategies and possible intentions of attackers can be recognized [8, 22] or extrapolated, such that target systems can possibly be protected before the complex attack is completed. This extends the use of intrusion detection to the practically more important domain of intrusion prevention. Thirdly, uncorrelated alarms can be assumed to be false alarms [21].

5 Reduction of Cooperation Data

While distributed cooperating intrusion detection is strongly needed, it also has the potential to be abused as a surveillance technology on a large scale. The arising conflict is an open problem not sufficiently investigated by the literature, as surveyed in Sect. 9. We propose to extend our approach for pseudonymization, as introduced in Sect. 2, as a promising solution.

The distributed character of future intrusion detection not only brings out the importance of privacy protection, but also the significance of *efficient* intrusion detection. We can compare analyzing an ever increasing amount of audit data with the attempt to drink from a fire hose. To put future intrusion detection systems in the position to cope with the audit data volume in practice, the results of local audit data pre-processing must be shared with other analyzing agents, thereby reducing the message volume and the workload of the analyzing agents. Successful cooperation will only be possible, if the data is not filtered too restrictively. Additionally, the data shared by cooperating agents may cross organizational boundaries. Both sharing and crossing boundaries stress the importance of *confidentiality* issues in this context. Naturally, companies would like to keep their local secrets confidential, also towards the agents of their remote business partners. The crossing of organizational boundaries also amplifies *privacy* requirements of individuals as compared to the situation within closed organizations.

Summarizing, we identify the following four potentially conflicting goals:

1. *detective effectiveness* of cooperation alliances,
2. *privacy* of honest individuals,

3. further organizational *confidentiality* requirements of local agents, and
4. *efficiency*.

Our previous solution exploits domain knowledge (about the given models of attack scenarios), as well as organization-specific knowledge (about requirements for privacy, analyzability and accountability). Accordingly, the solution for the more general problem is expected to use domain knowledge about the cooperation method to account for efficiency and cooperation effectiveness. And for considering privacy and confidentiality, again organization-specific knowledge is required. Particularly, in practice we need solutions with realistic and implementable trust requirements, because cooperating intrusion detection agents can be operated by different organizations. Our extended solution will be based on a generalization of the functional model from Sect. 2. An *information source* generates *events* or just *information*, being consumed by a source agent. The *source agent* represents the events or information using *structured data objects*. Appropriate information reductions process the structured data objects to satisfy detective effectiveness, privacy, confidentiality and efficiency. We distinguish lossless reductions and lossy reductions, depending on whether the original information from the structured data objects can be reconstructed or not, respectively:

- *Lossy reductions* remove information from structured data objects before forwarding it as *open data* to remote agents. The removed information must not be needed for further remote processing, and it should be definitely kept secret from remote agents, even under *inferences*. When information is coarsened, the detective effectiveness of the surveillance mode should not be affected unreasonably.
- *Lossless reductions* work by *splitting the information* contained in structured data objects into *open data* and *covered (masked, blinded) data* before forwarding it to remote agents (see Fig. 3).

The open data of a lossless reduction is sufficient for the normal surveillance mode of analyzing agents, possibly in conjunction with *exploiting* certain properties of the covered data, or in conjunction with some *supporting data* that must be additionally generated depending on the specific application. As an example, the covered data may be the protected input to a surveillance mode that is implemented using secure multi-party computation. If a specific detective purpose is met in the surveillance mode, a *purpose alert* is triggered. The respective open data together with the covered data allows for the reconstruction of the original information, subject to the detective purpose, e.g. sufficient suspicion. The *data with the reconstructed information* can be used in the *alert mode*, e.g. to hold perpetrators accountable. Note that lossless reductions are a generalization of the pseudonymization approach described in Sect. 2 (compare Fig. 1 and Fig. 3). The audit data with pseudonyms is open data being used in the normal surveillance mode, and the pseudonym recovery data is covered data, that can be used to reconstruct the identifying information in the original audit data, if and only if the detective purpose is met (purpose alert), i.e., a sufficient suspicion of misuse has occurred (threshold alert).

Both kinds of open data, as well as the supporting data should keep some information secret, even under inferences. In general, inference control is a highly challenging task that can only partially be solved in a purely algorithmic way, see e.g. [23, 24, 25]. Accordingly, in the context of intrusion detection, specific considerations are due.

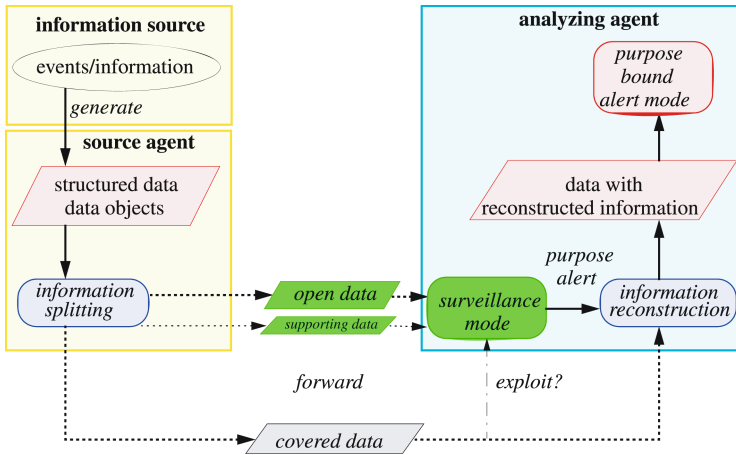


Fig. 3. Functional model for lossless reductions

6 Basic Requirements for Lossless Reductions

We summarize the required properties of the output of lossless reductions as follows:

- *Open data* in conjunction with *supporting data* must be sufficient for the normal *surveillance mode* of given kinds of analyzing agents.
- *Covered data* may possibly be exploited for the normal *surveillance mode* of given kinds of analyzing agents. The main use of the covered data is in the *alert mode* of given kinds of analyzing agents. For this purpose, the covered data must exhibit the following properties:
 - *correctness*: If the given purpose of the alert mode is met, the original information can be reconstructed by the analyzing agent from the open data together with the covered data. Note that this definition comprises technical and organizational purpose binding (cf. Sect. 2). However, in the following we focus on aspects of technical purpose binding.
 - *secrecy*: If the given purpose of the alert mode is not (yet) met, the original information cannot be determined by the analyzing agent from the open data, the supporting data and the covered data.
 - *verifiability*: If the analyzing agent does not trust the source agent to correctly perform the lossless reduction, the analyzing agent can recognize the covered data as useful for the surveillance mode, and for the information reconstruction for the alert mode.

7 Specific Requirements for Open Data

As required in Sect. 6 the open output data of lossless reductions must be sufficient for the normal surveillance mode of given kinds of analyzing agents. This also applies to lossy reductions. As a consequence and as mentioned in Sect. 5, accounting for the goal

detective effectiveness requires domain knowledge about the cooperation method(s) used by the analyzing agent(s). Krgel et al. survey existing work on such cooperation methods and propose a straightforward model incorporating useful heuristics for reducing the number of alarms, while retaining the relevant information [7].

Identifying Requirements for the Effectiveness of Cooperation Methods

In the following, we briefly motivate the rationale of these heuristics and focus on those of the operations, which use the open output data as operands. The ability to obtain useful results via these operations must be sustained in open data. Thereby we derive concrete requirements that must be met when generating open data using lossless reductions for cooperating intrusion detection agents.

Unlike local events, which are considered atomic events with only one *timestamp*, alarms have a *start time* and an *end time*, because they may describe a set of events that occurred at possibly distinct points in time. As can be seen below, start and end times play a crucial role in alarm correlation.

Most proposed approaches use exact matches of certain common features to identify alerts to be fused. However, two approaches employ probabilistic similarity measures for IP addresses [11, 18], and one approach reasons about implications of IP network addresses and directory paths [14]. Such constraints determine exact matches in the first p bits of the features, where p is the length of the common prefix of the features.

When aggregating/fusing a set of alarms into a complex alarm, the values of a given feature for all fused alarms are copied to the corresponding feature in the complex alarm, and the start and end time are determined from the set of alarms according to the semantics of the respective heuristic.

Firstly, alarms are mapped to a common format with common semantics and mandatory features (start time, end time, source, target), which are complemented in a best-effort manner. Then, various heuristics are applied to reduce the number of alarms by prioritizing alarms, or by fusing alarms into more complex alarms. For the names of the heuristics we follow the terminology of Krgel et. al. [7].

Alarm Fusion aims at discarding obvious duplicate alarms generated by different sensors when observing a given activity. Two given alarms are fused, if their start times fall in a configurable time interval, if they are generated by different sensors, and if for a given feature in both alarms the values are equal, if available. For alarm fusion, the distance of timestamps needs to be computed and compared to a constant value, and features are tested for equal content.

Alarm Verification aims at identifying irrelevant alarms and false alarms. As a prerequisite, a *verification database* is required, which maps each provided service to a verification method. The effect of a reported attack can be verified by determining the affected system service via the target port feature or the service identifier of the alarm and executing the associated verification method. For alarm verification, features are compared to entries in the verification database.

Attack Thread Reconstruction aims at representing subsequent alarms describing attacks from a given attacker on a given target. Two given alarms are fused, if the end time

of one alarm and the start time of the other alarm fall in a configurable time interval, and they are reported for the same source and the same target. For attack thread reconstruction the distance of timestamps needs to be computed and compared to a constant value, and features are tested for equal content.

Attack Session Reconstruction aims at correlating alarms that describe events on the network and in a host. As a prerequisite, a *port-process database* is required, which (1) maps network transport protocol port numbers to the identifiers of the processes on the host, who listen on the respective ports, and which (2) describes the parent-child relationship of the processes on the host. Two given alarms are fused, if the end time of the network-based alarm falls in a configurable time interval with the start time of the host-based alarm, and the target port feature of the network-based alarm identifies a port that is listened on by the process that is identified by the process id feature of the host-based alarm. For attack session reconstruction, the distance of timestamps needs to be computed and compared to a constant value, and features are compared to entries of the port-process database.

Attack Focus Recognition aims at fusing alarms describing attacks where a given attacker attacks many targets, e.g. reconnaissance scans, or where a given target is attacked by many sources, e.g. distributed denial of service attacks. Two given alarms are fused, if their start times fall in a configurable sliding time window, and they are reported for the same source or the same target. For attack focus recognition, the distance of timestamps needs to be computed and compared to a constant value, and features are tested for equal content.

Multi Step Correlation aims at representing alarm patterns constituting complex attacks as complex alarms. This is achieved by alarm correlation. Different approaches to centralized [15, 19, 10, 20, 16, 18, 17, 21, 14] and distributed [5, 6, 7] alarm correlation have been proposed (cf. Sect. 4). The operations relevant wrt. lossless reductions are independent from the possibly distributed nature of these approaches. We have already analyzed the problem of lossless reduction by pseudonymization in depth for the centralized case [1], which also applies to the distributed case. For multi step correlation, the timestamps need to be compared wrt. to their order, and features are tested for equal content, or their features are compared to constant values.

Impact Analysis and Alarm Prioritization aim at determining the effect of an attack in order to prioritize the respective alarm accordingly. As a prerequisite, an *asset database* is required, which (1) maps provided services to the importance of the services, and which (2) models the dependencies of the services. The impact of an attack described by a given alarm can be determined by identifying the target port or service identifier feature of the alarm in the asset database and determining all services depending on that service. The identified services can then be correlated with service failures detected by service monitors.

Moreover, the asset database is used to determine the importance of the affected services and to prioritize the involved alarms accordingly. Alarms that have already been fused into more complex alarms, as well as alarms found to be irrelevant are set to a low priority, such that they are still available for review. For impact analysis and for alarm prioritization, features are compared to entries of the asset database.

Resulting Requirements for Open Data

It can be seen, that the following requirements are crucial for sustaining the functionality of the relevant operations of the above heuristics:

- R1:** certain alarm features (except for timestamps) need to be compared to certain alarm features for equal content, or equal prefix content
- R2:** certain alarm features (except for timestamps) need to be compared to values outside of the open data, e.g. constant values, entries of a database (see above)
- R3:** distances of alarm timestamps need to be computed and compared to values outside of the open data, i.e. a constant value
- R4:** the order of alarm timestamps needs to be determined

As a result, in order to sustain the effectiveness of the surveillance mode of the analyzing agent(s), lossy reductions must be designed, such that they do not remove timestamps and features that are used in the aforementioned operations (cf. Sect. 9).

Lossless reductions must be designed, such that the above operations can still be computed on the described alarm timestamps and alarm features, and such that the results of the operations are still meaningful, i.e., for an operation $\circ \in \{=_p, <, >\}$, where $=_p$ compares the features up to a suitably determined prefix p , including $=_\infty$ for the full length, and two operands op_1 and op_2 in the open data and for a lossless reduction $r()$ holds $op_1 \circ op_2 = r(op_1) \circ r(op_2)$.

Note that sustaining the ability to compare alarm features or operation results on alarm features to values outside of the open data may require to provide *supporting data*, i.e., the reduction $r_s()$ uses some parameter s , such that $op_1 \circ op_2 = r_s(op_1) \circ r_s(op_2)$ holds, where $r_s(op_1)$ is computed by the source agent and $r_s(op_2)$ is computed by the analyzing agent.

Specific Requirements for Pseudonyms in Open Data

In the following, we consider pseudonymization as a special case of lossless reductions, and we focus on the specific requirements for pseudonyms in the open data, such that the surveillance mode of the analyzing agent(s) sustains its detective effectiveness. Hence, the lossless reduction $r_s()$ replaces some feature f with an appropriate pseudonym $r_s(f)$, where s is a parameter that can be used to generate distinct pseudonyms for f . Note that $r_s(f)$ needs to preserve the comparability of feature prefixes, if required by R1, e.g. [26]. Also note that pseudonyms traditionally are used to hide personal data, such as identifiers of users, but within our generalized scope we consider pseudonyms as place-holders for arbitrary features. A pseudonym $r_s(f)$ is appropriate, if

- $r_s(f)$ respects the syntax constraints of the surveillance mode wrt. f
- $f =_p f' \Rightarrow r_s(f) =_p r_s(f')$ holds if R1 requires that f must be testable for equal content or prefix to an alarm feature f' ; note that both $r_s(f)$ and $r_s(f')$ are computed by the source agent
- $f \neq_p f', s \neq s' \Rightarrow r_s(f) \neq_p r_s(f'), r_s(f) \neq_p r_{s'}(f')$ holds generally, i.e., $r_s()$ is collision-resistant, such that no unrelated alarms are correlated by accident
- $f = c \Rightarrow r_s(f) = r_s(c)$ holds if R2 requires $r_s(f)$ to be testable for equal content of a clear-text value c ; note that $r_s(f)$ is computed by the source agent, who also provides s in the supporting data, such that the analyzing agent can compute $r_s(c)$

- $f \neq c \Rightarrow r_s(f) \neq r_s(c)$ holds generally, i.e. $r_s()$ is collision-resistant (see above)

Note that R2 can be required independently from R1 wrt. to f , such that R2 may be required in addition to R1. The source agent only needs to provide s in the supporting data, if R2 holds. Also note that a database lookup for a given $r_s(f)$ requires the analyzing agent to compute $r_s(c)$ for all c visited in the database, until a match is found.

In order to reduce the inferences an attacker can make on the transitive closure of a given pseudonym, it is desirable to use $r_s()$ in way, such that additionally

- $f =_p f' \Rightarrow r_s(f) \neq_p r_{s'}(f'), s \neq s'$ holds if R1 does not require that f must be testable for equal content or prefix to an alarm feature f' ; note that the analyzing agent then does not need to and therefore is incapacitated to decide whether $f =_p f'$ or $f \neq_p f'$

We assume that all (source/analyzing) agents a priori know the heuristics, such that all agents know, when R1 and/or R2 are required. This assumption can be met by proper coordination of the configuration of all agents.

All source agents that pseudonymize a given f must choose s in a coordinated way, if R1 is required. Then, the analyzing agents can correlate alarms originating from distinct source agents by means of $r_s(f)$. If there is no coordination wrt. s , the following error can occur: $r_s(f) \neq_p r_{s'}(f'), s \neq s'$, despite $f =_p f'$, resulting in failure to correlate related alarms, i.e., the number of alarms is not reduced.

Depending on the given communication infrastructure it may be viable to choose s in a coordinated way. However, this seems to be impossible to achieve for simultaneously generated alarms, if there are real-time constraints to be respected, i.e., alarms cannot be buffered until s has been agreed upon by all agents. Note that a single centralized source agent can choose s randomly [1] and thus can significantly reduce the transitive closure of a given pseudonym, i.e. reduce the working surface of an attacker who reasons about pseudonyms. Assuming that a timely coordination of s with all source agents is impractical, s should be a static parameter that is known to all source agents cooperating with a given set of analyzing agents. If a source agent wants to prevent the analyzing agent from dictionary attacks yielding information about the source agent's private network, it could use a secret value s to pseudonymize all highly critical information. The down-side of this approach is, that the pseudonyms for the highly critical information cannot be correlated with alarms from other source agents. Our solutions for the centralized case need to be carefully adapted to the requirements wrt. s [1].

Regarding R3 and R4, currently no useful $r_s()$ is known, which meets these requirements in the distributed case. Thus, so far timestamps must not be pseudonymized in alarms. Note that time-shifting and enumeration can be used in the centralized case [27], but require timely coordination of all source agents in the distributed case, which we assumed to be impractical. However, timestamps could be coarsened by lossy reductions removing the more fine-grained time units (cf. [27, 28, 29] in Sect. 9), or the point in time could be hidden by removing the more coarse-grained time units [27].

8 Specific Requirements for Covered Data

In contrast to the open data the covered data is basically independent from the cooperation methods used by the analyzing agent(s). This results from the fact that the covered data is not used in the surveillance mode. In the following, we ignore the extension that the surveillance mode may exploit certain properties of the covered data. Rather, the covered data is used for information reconstruction when a purpose alert is triggered in the surveillance mode. This obviously results in fewer specific constraints for the design of lossless reductions.

The basic requirements for lossless reductions must still be met in a distributed setup. Regarding correctness, the lossless reduction must generate covered data that allows for information reconstruction, even if the covered data used for information reconstruction is generated by distinct source agents accounting for simultaneous events at different locations of the system. As an example, a lossless reduction may provide the information, which is hidden in the open data, in several parts in masked form in the covered data, where each *insufficient set of parts* satisfies the secrecy requirement. The information reconstruction requires a *sufficient set of parts* to enable reconstructing the hidden information. Then, if several distinct source agents generate such parts accounting for events at different locations of the system, the parts in the sufficient set must still “fit together”, such that the original information can be reconstructed.

Clearly, this requires the source agents to generate covered data in a coordinated way. Note the analogy to the situation wrt. to the parameter s in Sect. 7. The same rationale applies here, and we assume that a timely coordination of all source agents is impractical. As a result, covered data must be generated using only the original information to be hidden and information that is known a priori to all source agents.

Extending Existing Solutions for Pseudonym Recovery Data

Considering our previous work using Shamir’s threshold scheme for cryptographic secret sharing [4], we find that it is viable in the centralized case to choose keys for encryption and decryption randomly, and that shares can be generated by using linear sample points in an increasing order [3]. Both of these methods do not extend to the distributed case. We envision the following two solutions for the problem of distinct share generation. First, each source agent is assigned his own interval for sample points, such that each source agent generates distinct sample points. The downside of this approach is, that the number of possible sample points per source agent is artificially limited depending on the number of source agents. Second, source agents can choose sample points pseudo-randomly, where each source agent initializes his pseudo-random number generator with a distinct secret. While this solution does not reduce the number of possible sample points per source agent, there is a chance for collisions.

The problem of synchronized key generation in the absence of timely coordination could be solved by using cryptographic one-way functions to compute a key, where some common secret and the feature to be encrypted are used as parameters. Such a solution is less secure than using random keys, because the search-space is reduced to all syntactically possible features, opening an avenue for dictionary attacks.

9 Related Work and Conclusion

Slagell and Yurcik summarize related work in the field of audit data anonymization, analyze the problem based on a collection of attacks against anonymization schemes and propose a straightforward architecture for audit data anonymization considering merely lossy reductions [30]. However, their analysis focuses on the identifying information found within the audit data and does not sufficiently consider the linkability of features, as required by the applications using the audit data. The CANINE tool implemented by Slagell et al. is heavily geared to pseudonymization of network flows [27], merely provides for lossy reductions, and does not meet requirement R2. Similarly, Pang and Paxson turn their IDS Bro into a tool for network data anonymization [31], merely providing lossy reductions and no support for R2.

Lincoln et al. propose a solution for alarm repositories, which collect alarms from source agents and publish them for further analysis [28]. They analyze, which kinds of surveillance modes are supported by their solution with the result that they only support limited analysis capabilities. We take the other road, firstly analyzing the requirements for surveillance modes proposed by the community and then deriving requirements for designing appropriate solutions. The solution proposed by Lincoln et al. does not meet R1 wrt. prefixes and R2, and supports merely lossy reductions, it however introduces a notion of uncertainty by coarsening timestamp features. Xu and Ning propose the first solution directly aimed at supporting cooperation of intrusion detection agents [29]. They merely consider lossy information reduction by coarsening features using concept hierarchies. Obviously, coarsening features introduces uncertainty, leading to worse correlation results, which may or may not be acceptable.

Summarizing, there are currently no solutions supporting lossless reductions for cooperating distributed intrusion detection agents. Lossless reductions provide a chance for privacy protection without sacrificing the precision of surveillance mode results.

Our previous work and the proposed general model for information reduction for cooperating intrusion detection agents are aimed towards technically balancing the conflicting interests in intrusion detection. The general model brings up challenges with respect to intrusion detection, inference control and cryptography. For both, lossy and lossless information reductions, we identified the information requirements of the normal surveillance mode of the given analyzing agents. For lossless reductions we still need to identify and formalize the given purposes for the alert mode. According to these requirements we need to define appropriate data transformations and prove their effectiveness as well as their efficiency for both modes (intrusion detection). We provided first ideas, how the transformations we proposed for the centralized case [1] could be adapted to the distributed case.

For lossy reductions we need to prove that (a sufficient degree of) confidentiality is achieved with respect to remote agents (inference control) and in the case of coarsening we need to show that the detective effectiveness of the surveillance mode is only mildly affected. Finally, for lossless reductions we need to prove that they exhibit the properties correctness, secrecy preservation, verifiability and possibly further properties that are important in the context of the given analyzing agents (cryptography).

References

- [1] Ulrich Flegel. *Pseudonymizing Audit Data for Privacy Respecting Misuse Detection*. PhD thesis, University of Dortmund, Dept. of Computer Science, 2005.
- [2] Ulrich Flegel. Pseudonymizing Unix log files. In George Davida, Yair Frankel, and Owen Rees, editors, *Proceedings of the Infrastructure Security Conference (InfraSec2002)*, number 2437 in Lecture Notes in Computer Science, pages 162–179, Bristol, United Kingdom, October 2002. Springer.
- [3] Joachim Biskup and Ulrich Flegel. Threshold-based identity recovery for privacy enhanced applications. In Sushil Jajodia and Pierangela Samarati, editors, *Proceedings of the 7th ACM Conference on Computer and Communications Security*, pages 71–79, Athens, Greece, November 2000. ACM SIGSAC, ACM Press.
- [4] Adi Shamir. How to share a secret. *Communications of the ACM*, 22:612–613, 1979.
- [5] Giovanni Vigna, Richard A. Kemmerer, and Per Blix. Designing a web of highly-configurable intrusion detection sensors. In Lee et al. [32], pages 69–84.
- [6] Peng Ning, Sushil Jajodia, and X. Sean Wang. *Intrusion Detection in Distributed Systems*. Number 9 in Advances in Information Security. Springer, 2004.
- [7] Christopher Krgel, Fredrik Valeur, and Giovanni Vigna. *Intrusion Detection and Correlation*. Number 14 in Advances in Information Security. Springer, 2005.
- [8] Ming-Yuh Huang, Robert J. Jasper, and Thomas M. Wicks. A large scale distributed intrusion detection framework based on attack strategy analysis. *Computer Networks*, 31(23–24):2465–2475, 1999.
- [9] Tim Bass. Intrusion detection systems and multisensor data fusion. *Communications of the ACM*, 43(4):99–105, 2000.
- [10] Hervé Debar and Andreas Wespi. Aggregation and correlation of intrusion-detection alerts. In Lee et al. [32], pages 85–103.
- [11] Alfonso Valdes and Keith Skinner. Probabilistic alert correlation. In Lee et al. [32], pages 54–68.
- [12] Frédéric Cuppens. Managing alerts in a multi-intrusion detection environment. In *Proceedings of the 17th Annual Computer Security Applications Conference (ACSAC 2001)*, pages 22–31, New Orleans, Louisiana, USA, December 2001. IEEE Computer Society Press.
- [13] Philip A. Porras, Martin W. Fong, and Alfonso Valdes. A mission-impact-based approach to INFOSEC alarm correlation. In Andreas Wespi, Giovanni Vigna, and Luca Deri, editors, *Proceedings of the Fifth International Symposium on Recent Advances in Intrusion Detection (RAID 2002)*, number 2516 in Lecture Notes in Computer Science, pages 95–114, Zurich, Switzerland, October 2002. Springer.
- [14] Dingbang Xu and Peng Ning. Alert correlation through triggering events and common resources. In *Proceedings of the 20th Annual Computer Security Applications Conference (ACSAC 2004)*, pages 360–369, Tucson, Arizona, USA, December 2004. IEEE Computer Society Press.
- [15] Louis Perrochon, Eunhei Jang, and David C. Luckham. Enlisting event patterns for cyber battlefield awareness. In *Proceedings of the DARPA Information Survivability Conference and Exposition (DISCEX 2000)*, pages 1411–1422, Hilton Head, South Carolina, January 2000. DARPA and the IEEE Computer Society, IEEE Press.
- [16] Nathan Carey, Andrew Clark, and George Mohay. IDS interoperability and correlation using IDMEF and commodity systems. In *Proceedings of the Fourth International Conference on Information and Communications Security (ICICS 2002)*, number 2513 in Lecture Notes in Computer Science, pages 252–264, Singapore, December 2002.

- [17] Benjamin Morin and Hervé Debar. Correlation of intrusion symptoms: An application of chronicles. In Giovanni Vigna, Erland Jonsson, and Christopher Krgel, editors, *Proceedings of the Sixth International Symposium on Recent Advances in Intrusion Detection (RAID 2003)*, number 2820 in Lecture Notes in Computer Science, pages 94–112, Pittsburgh, Pennsylvania, USA, September 2003. Springer.
- [18] Oliver M. Dain and Robert K. Cunningham. *Applications of Data Mining in Computer Security*, chapter Fusing Heterogeneous Alert Streams into Scenarios. Kluwer, Boston, 2002.
- [19] Steven J. Templeton and Karl Levitt. A requires/provides model for computer attacks. In *Proceedings of the New Security Paradigms Workshop*, pages 31–38, Cork, Ireland, September 2000. ACM, ACM Press.
- [20] Frédéric Cuppens and Alexandre Miège. Alert correlation in a cooperative intrusion detection framework. In *Proceedings of the IEEE Symposium on Research in Security and Privacy*, pages 202–215, Berkeley, California, USA, May 2002. IEEE, IEEE Press.
- [21] Peng Ning, Yun Cui, Douglas S. Reeves, and Dingbang Xu. Techniques and tools for analyzing intrusion alerts. *ACM Transactions on Information and System Security*, 7(2):274–318, May 2004.
- [22] Peng Ning and Dingbang Xu. Learning attack strategies from intrusion alerts. In Sushil Jajodia, Vijay Atluri, and Trent Jaeger, editors, *Proceedings of the 10th ACM Conference on Computer and Communications Security*, pages 200–209, Washington, D.C., USA, October 2003. ACM SIGSAC, ACM Press.
- [23] Dorothy E. Denning. *Cryptography and Data Security*. Addison-Wesley, 1982.
- [24] Csilla Farkas and Sushil Jajodia. The inference problem: a survey. *ACM SIGKDD Explorations Newsletter*, 4(2):6–11, 2002.
- [25] Joachim Biskup and Piero A. Bonatti. Controlled query evaluation for enforcing confidentiality in complete information systems. *International Journal of Information Security*, 3(1):14–27, 2004.
- [26] Jun Xu, Jinliang Fan, Mostafa Ammar, and Sue B. Moon. Prefix-preserving IP address anonymization: Measurement-based security evaluation and a new cryptography-based scheme. In *Proceedings of the 10th IEEE International Conference on Network Protocols (ICNP)*, pages 280–289, 2002.
- [27] Yifan Li, Adam Slagell, Katherine Luo, and William Yurcik. CANINE: A combined converter and anonymizer tool for processing netflows for security. In *Proceedings of the international Conference on Telecommunication Systems - Modeling and Analysis (ICTSM 2005)*, Dallas, Texas, USA, November 2005.
- [28] Patrick Lincoln, Phillip Porras, and Vitaly Shmatikov. Privacy-preserving sharing and correlation of security alerts. In *Proceedings of the 13th USENIX Security Symposium*, pages 239–254, San Diego, California, USA, August 2004.
- [29] Dingbang Xu and Peng Ning. Privacy-preserving alert correlation: A concept hierarchy based approach. In *Proceedings of the 21st Annual Computer Security Applications Conference (ACSAC 2005)*, pages 537–546, Tucson, Arizona, USA, December 2005. IEEE Computer Society Press.
- [30] Adam Slagell and William Yurcik. Sharing computer network logs for security and privacy: A motivation for new methodologies of anonymization. In *Workshop on the Value of Security through Collaboration (SECOVAL)*, 2005.
- [31] Ruoming Pang and Vern Paxson. A high-level programming environment for packet trace anonymization and transformation. In *Proceedings of the 2003 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications (SIGCOMM 2003)*, pages 339–351, Karlsruhe, Germany, August 2003. ACM, ACM Press.
- [32] Wenke Lee, Ludovic Mé and Andreas Wespi, editors. *Proceedings of the Fourth International Symposium on Recent Advances in Intrusion Detection (RAID 2001)*, number 2212 in Lecture Notes in Computer Science, Davis, California, October 2001. Springer.