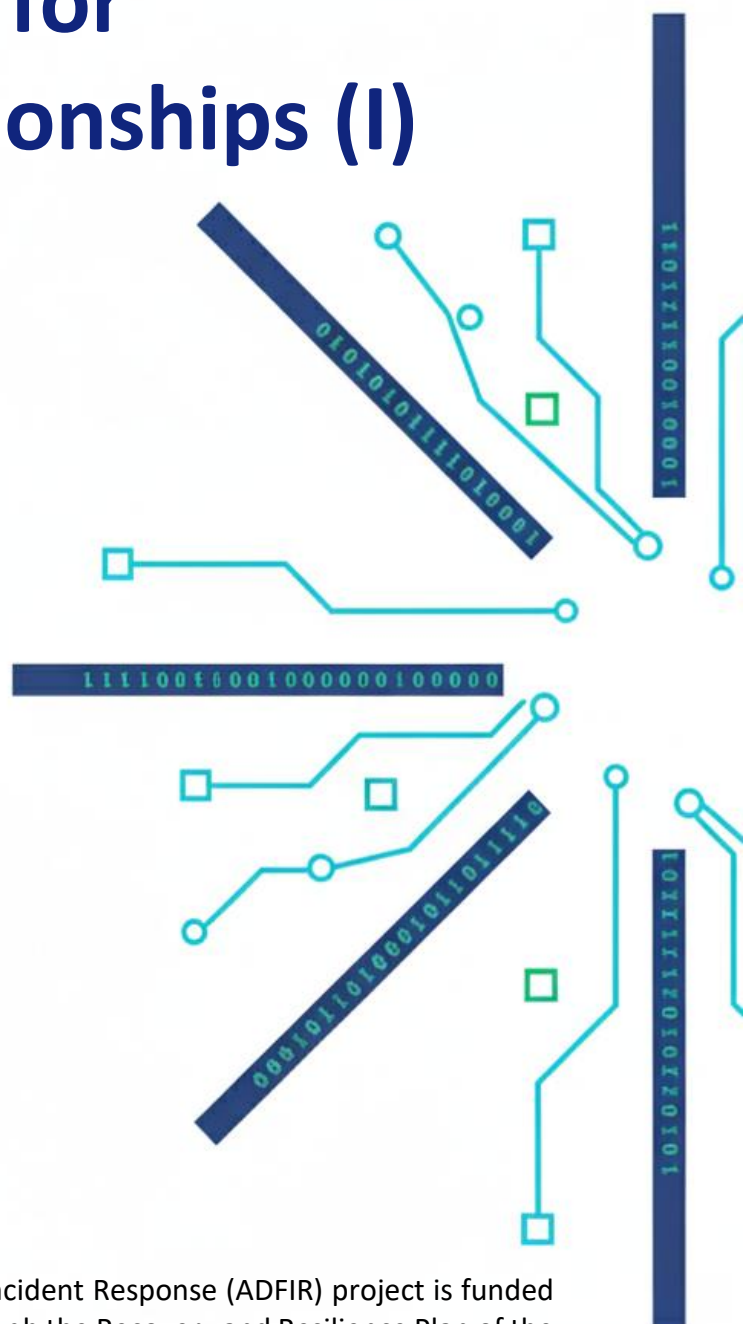


D20 – A method for identifying relationships (I)



The Automation of Digital Forensic Analysis and Incident Response (ADFIR) project is funded by the European Union – Next Generation EU through the Recovery and Resilience Plan of the Slovak Republic under project number 09I05-03-V02-00079.

Contents

Project description	2
Introduction.....	3
1 Formal Concept Analysis	4
2 Decomposition of the formal context	6
2.1 <i>Factorisation of binary matrices</i>	6
2.2 <i>Formal concepts as factors</i>	7
2.3 <i>The Set Cover Problem</i>	8
3 Factor-finding algorithm	10
4 Application of the algorithm to factor detection	12
4.1 <i>General and specific factors</i>	12
4.2 <i>Factors related to attacks</i>	15
Conclusion	17
Bibliography	18
Appendices	19

Project description

The project **Automation of Digital Forensic Analysis and Incident Response** (hereinafter referred to as 'ADFIR') is funded by **the European Union – Next Generation EU through the Recovery and Resilience Plan of the Slovak Republic** under project number 09I05-03-V02-00079. This project addresses one of the key challenges in the field of cyber security and information security – how to process the vast amount of digital evidence generated during cyber security incidents or forensic investigations. At present, this process is very demanding in terms of human resources and time. Automation using machine learning methods can therefore significantly **improve the quality of digital forensic analysis** and reduce the time required to carry it out. Overall, this enables security teams to respond more effectively to cyber threats. The main benefits of this project are:

- **Faster resolution of cyber security incidents.** The ADFIR project introduces automated approaches to the collection, processing and analysis of digital traces. This enables security teams to identify the causes of incidents more quickly and take effective measures to resolve them.
- **Reduced workload for forensic analysts.** Routine and time-consuming tasks associated with processing digital traces will be replaced by automated methods. This will allow analysts to focus on more complex cases and strategic decision-making.
- **Higher quality and consistency of outputs.** The use of standardised methodologies and tools ensures that processed digital evidence will be more accurate, consistent and easier to verify. This significantly reduces the risk of errors caused by human factors.
- **Potential use in criminal proceedings.** The project's outputs will be developed in accordance with legal requirements and standards, enabling digital traces to be accepted as relevant evidence for investigations and court proceedings.

Introduction

Digital forensic analysis deals with large volumes of data from various sources, such as file systems, operating system logs and web artefacts. Individual events and their characteristics often only become meaningful when viewed in context.

The aim is to design and apply a method for identifying relationships in data from digital forensic analysis. One approach suitable for identifying relationships is formal concept analysis. It enables the search within data for formal concepts consisting of a set of objects and a set of attributes shared by these objects.

However, the number of formal concepts can grow exponentially depending on the structure of the formal context. With large datasets, generating a concept bundle is time-consuming, and the subsequent manual analysis and interpretation of a large number of concepts is practically impossible. The proposed method therefore does not require the generation of a complete conceptual lattice, but selects a subset of formal concepts that covers all instances in the input dataset. The method utilises matrix factorisation based on formal concepts. For factor selection, it uses the Rice-Siff Factorisation algorithm, which combines a greedy approach to the set-covering problem with the iterative generation of candidate formal concepts.

1 Formal Concept Analysis

Formal Concept Analysis (FCA)[1] is a mathematical framework based on set theory, which is used to discover concepts in binary data and organise them hierarchically. The input data can be represented as a binary table, in which rows correspond to objects, columns to attributes, and a value of 1 indicates that the object in question possesses the relevant attribute.

Formally, let B and A be non-empty sets of objects and attributes, and let $R \subseteq B \times A$ be a binary relation. Then the triple (B, A, R) is called a **formal context** – an example given at **Figure 1**.

	i	ii	iii
a	×		
b	×	×	
c		×	

Figure 1 – Formal context

To find the common attributes of a subset of objects and the common objects of a subset of attributes, the operators $\uparrow: \mathcal{P}(B) \rightarrow \mathcal{P}(A)$ and $\downarrow: \mathcal{P}(A) \rightarrow \mathcal{P}(B)$, are used, where $\mathcal{P}(B)$ and $\mathcal{P}(A)$ denote the power sets of the sets B and A respectively.

For the subset of objects $X \subseteq B$, we define

$$X^\uparrow = \{a \in A \mid \forall b \in X: (b, a) \in R\}.$$

Similarly, for the subset of attributes $Y \subseteq A$, we define

$$Y^\downarrow = \{b \in B \mid \forall a \in Y: (b, a) \in R\}.$$

The operators $\uparrow$ and $\downarrow$ form a **Galois connection**. By composing them, $X^{\uparrow\downarrow}$ and $Y^{\downarrow\uparrow}$, we obtain closure operators.

A pair (X, Y) , for which $X^\uparrow = Y$ holds and, at the same time, $Y^\downarrow = X$ holds, is called a **formal concept**. The object part of the concept X is called the extent, and the attribute part of the concept Y is called the intent.

A formal concept therefore expresses a complete pair – a closed subset of objects together with all the attributes that this subset has in common. In a tabular representation, we can think of a formal concept as the largest rectangle filled with units. This means that we cannot extend it by adding another row or column without it containing a zero.

We denote the set of all formal concepts of the formal context (B, A, R) as $\mathcal{B}(B, A, R)$.

Formal concepts can be ordered using the relation $(X_1, Y_1) \leq (X_2, Y_2)$ precisely when $X_1 \subseteq X_2$, or $Y_2 \subseteq Y_1$.

We call a partially ordered set of formal concepts $(\mathcal{B}(B, A, R), \leq)$ a **concept bundle**. A concept bundle can be visualised using a line diagram (**Figure 2**).

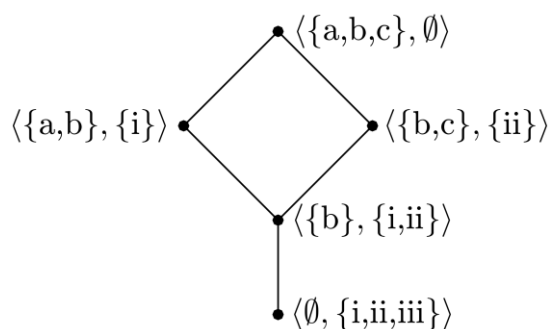


Figure 2 – Concept bundle

According to the fundamental theorem on conceptual bundles [1], this bundle is complete; that is, for every subset of formal concepts within it, there exists both an infimum and a supremum.

2 Decomposition of the formal context

Let us represent the formal context (B, A, R) by a binary incidence matrix. We can then understand the decomposition of the formal context as a decomposition of the incidence matrix.

The aim of the decomposition is to identify a smaller set of latent structures, referred to as **factors**, which capture the relationships between subsets of objects and subsets of attributes. We seek two binary matrices

$$U \in \{0,1\}^{|B| \times k}$$

and

$$V \in \{0,1\}^{k \times |A|},$$

where k represents the number of factors. The matrix U describes the relationship between objects and factors, whilst the matrix V describes the relationship between factors and attributes. The original incidence matrix is subsequently reconstructed using the Boolean product of these matrices.

2.1 Factorisation of binary matrices

Let $R \in \{0,1\}^{|B| \times |A|}$ be the incidence matrix of a formal context. **The factorisation of a binary matrix** is a decomposition

$$R = U \circ V,$$

where

$$U \in \{0,1\}^{|B| \times k}, V \in \{0,1\}^{k \times |A|}$$

and the symbol $\circ$ denotes the Boolean product of matrices. For every element of the resulting matrix, the following holds

$$(U \circ V)_{ij} = \bigvee_{\ell=1}^k (U_{i\ell} \wedge V_{\ell j}),$$

where $\wedge$ denotes logical conjunction and $\vee$ denotes logical disjunction.

An example of a matrix decomposition is shown in **Figure 3**.

$$\begin{array}{ccc}
R & U & V \\
\begin{pmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 0 & 1 & 0 \end{pmatrix} & = & \begin{pmatrix} 1 & 0 \\ 1 & 1 \\ 0 & 1 \end{pmatrix} \circ \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}
\end{array}$$

Figure 3 – Matrix decomposition

The value $(U \circ V)_{ij} = 1$ therefore occurs precisely when there is at least one factor ℓ for which the following holds simultaneously

$$U_{i\ell} = 1$$

and

$$V_{\ell j} = 1.$$

Depending on the required quality of the reconstruction, we distinguish between exact and approximate factorisation. In **exact factorisation**, the following holds

$$R = U \circ V,$$

the reconstructed matrix is identical to the original incidence matrix. In **approximate factorisation**, it is permitted that

$$R \approx U \circ V,$$

which may result in two types of errors:

- undercover – occurs when a unit element from the original matrix is not reconstructed during factorisation,
- overcover – occurs when the reconstructed matrix contains a unit at a position where the original matrix had a value of zero.

2.2 Formal concepts as factors

When decomposing a binary matrix, we can represent the individual factors using formal concepts [2]. Let

$$\mathcal{F} = \{\langle X_1, Y_1 \rangle, \dots, \langle X_k, Y_k \rangle\} \subseteq \mathfrak{B}(B, A, R)$$

A set of formal concepts is selected. This set subsequently induces a matrix of objects and factors $U_{\mathcal{F}}$ and a matrix of factors and attributes $V_{\mathcal{F}}$.

A fundamental property of formal concepts as factors is their **universality**. For every binary matrix R , there exists a set of formal concepts $\mathcal{F} \subseteq \mathfrak{B}(B, A, R)$ such that

$$R = U_{\mathcal{F}} \circ V_{\mathcal{F}}.$$

Formal concepts are therefore always sufficient for the exact reconstruction of the original incidence matrix.

Formal concepts are also **optimal factors**. If the matrix R has an arbitrary decomposition $R = U \circ V$ with k factors, then there exists a set of formal concepts $\mathcal{F}$ such that

$$|\mathcal{F}| \leq k$$

and

$$R = U_{\mathcal{F}} \circ V_{\mathcal{F}}.$$

A decomposition with the minimum possible number of factors can therefore always be found amongst the formal concepts. The set of formal concepts thus constitutes a sufficient and non-redundant space of candidate factors for finding the optimal decomposition.

We can also justify the optimality of formal concepts geometrically. Every rectangle induced by any factor is contained within some maximal rectangle corresponding to a formal concept. Replacing the original factors with formal concepts preserves the exact reconstruction of the matrix, and the number of factors does not increase.

Another important property of formal concepts as factors is their **direct interpretability**. In factor analysis, factors are usually interpreted as latent dimensions that manifest themselves through the original observed attributes. When formal concepts are used, the meaning of a factor is explicitly determined by its extent and intent. The extent X_{ℓ} identifies the set of objects to which the factor relates, whilst the intent Y_{ℓ} determines the set of attributes through which the factor manifests itself.

In digital forensic analysis, a single factor may represent a recurring pattern of events, such as a combination of file artefacts, operating system registries and network communications that occurs across multiple objects. Factors thus make it possible to replace a large number of individual occurrences with a smaller number of interpretable structures. A security analyst can then examine these factors rather than analysing each object individually, which facilitates the identification of recurring patterns of behaviour and potential security incidents.

2.3 The Set Cover Problem

The search for a matrix decomposition with a minimum number of factors is closely related to the set cover problem.

Since each factor is represented by a formal concept (X, Y) , which covers the incidence rectangle $X \times Y \subseteq R$, the aim is to find the smallest possible set of formal concepts $F \subseteq \mathcal{B}(B, A, R)$ such that their corresponding rectangles cover all unit elements of the incidence matrix, i.e.

$$R = \bigcup_{(X,Y) \in F} (X \times Y).$$

The set-coverage optimisation problem is NP-hard. For this reason, approximation and heuristic algorithms are used in practice. The greedy approach selects, at each step, a formal concept that covers the largest number of as yet uncovered incidences, and repeats this process until the entire matrix is covered.

Furthermore, the computational complexity is increased by the fact that the number of formal concepts can generally be exponential. For a formal context with n objects and m attributes, the number of formal concepts can reach as high as $2^{\min(n,m)}$. For large contexts, it is challenging to generate the entire concept bundle and subsequently search through all candidate factors. Therefore, algorithms are used which, instead of searching through all formal concepts, progressively generate and evaluate only suitable candidates.

3 Factor-finding algorithm

The **Rice-Siff Factorisation (RSF)** algorithm [3] is designed for the exact factorisation of an incidence matrix using formal concepts. It is based on a greedy approach to the set-covering problem, but instead of searching the entire concept bundle, it generates a set of candidate factors by progressively merging formal concepts.

During execution, the algorithm maintains the set $R_{\text{un}} \subseteq R$, which contains incidences not yet covered by any selected factor. Initially, $R_{\text{un}} = R$ holds. After each factor is selected, the incidences covered by that factor are removed from the set R_{un} . The process continues until the set R_{un} is empty ($R_{\text{un}} = \emptyset$).

In each iteration of the main loop, two sets of concepts are created: $\mathcal{C}_1$ and $\mathcal{C}_2$. Both sets are initialised with the set of object concepts

$$\mathcal{C}_1 = \mathcal{C}_2 = \{(\{b\}^\downarrow, \{b\}^\uparrow) \mid b \in B\}.$$

The set $\mathcal{C}_1$ represents the current working set of concepts that are being progressively merged. The set $\mathcal{C}_2$ stores all candidates created during agglomeration, i.e. both the original object concepts and their subsequent supremums.

A state-dependent semi-metric ρ_{un} is used to select pairs of concepts for merging. For the formal concepts (X_1, Y_1) and (X_2, Y_2) , it is defined by the relation

$$\rho_{\text{un}}((X_1, Y_1), (X_2, Y_2)) = 1 - \frac{|(X_1 \times Y_1) \cap (X_2 \times Y_2) \cap R_{\text{un}}|}{|(X_1 \times Y_1) \cup (X_2 \times Y_2)|}.$$

Its value depends on the current set of uncovered incidences. Pairs of concepts with a lower value of ρ_{un} have greater common coverage in the residual relation and therefore represent more suitable candidates for merging.

The residual coverage weight is used to select the factor

$$w(X, Y) = |(X \times Y) \cap R_{\text{un}}|.$$

The value $w(X, Y)$ indicates the number of as yet uncovered incidences that can be covered by the concept (X, Y) . Upon completion of the agglomeration phase, the concept with the maximum residual weight is selected from the set $\mathcal{C}_2$.

Concept merging takes place in an inner loop, in which a set of suitable pairs is first constructed

$$\mathcal{P} = \{(O_i, O_j) \in \mathcal{C}_1 \times \mathcal{C}_1 \mid Y_i \cap Y_j \neq \emptyset\}.$$

The condition that the intersection of intents must be non-empty is important. For if $Y_i \cap Y_j = \emptyset$, then the supremum of these concepts has an empty intent and does not cover any incidence. Such a merger is irrelevant to solving the coverage problem.

From the set $\mathcal{P}$, the pair with the smallest value ρ_{un} is selected. These two concepts are removed from the working set $\mathcal{C}_1$ and their supremum is formed

$$O_{new} = O_i \vee O_j.$$

For $O_i = (X_i, Y_i)$, $O_j = (X_j, Y_j)$, it is defined by the relation

$$O_i \vee O_j = ((X_i \cup X_j)^{\uparrow\downarrow}, Y_i \cap Y_j).$$

A new concept is inserted into both sets $\mathcal{C}_1$ and $\mathcal{C}_2$. Agglomeration continues until the working set contains at most one concept or until there is no pair with a non-empty intersection of intents.

Once the internal agglomeration cycle has ended, the concept (X, Y) with the maximum weight $w(X, Y)$ is selected from the set $\mathcal{C}_2$. The selected concept is added to the set of factors F and the incidences covered by it are removed.

The candidate sets are then reinitialised with object concepts, and the entire procedure is carried out with respect to the updated residual relation.

The output of the algorithm is the set of factors F . The pseudocode for the algorithm is shown in **Figure4**.

Algorithm 1: Rice-Siff Factorization (RSF)

Input: A formal context (B, A, R) with $R \subseteq B \times A$
Output: Factor set F

```

1  $F \leftarrow \emptyset$ 
2  $R_{un} \leftarrow R$ 
3 while  $R_{un} \neq \emptyset$  do
4    $C_1 \leftarrow \{(\{x\}^{\uparrow\downarrow}, \{x\}^{\uparrow}) \mid x \in B\}$ 
5    $C_2 \leftarrow C_1$ 
6   while  $|C_1| > 1$  do
7      $P \leftarrow \{(O_i, O_j) \in C_1 \times C_1 \mid Y_i \cap Y_j \neq \emptyset\}$ 
8     if  $P = \emptyset$  then break
9     Find  $(O_i, O_j) \in P$  minimizing  $\rho_{un}(O_i, O_j)$ 
10    Remove  $O_i, O_j$  from  $C_1$ 
11     $O_{new} \leftarrow O_i \vee O_j$ 
12    Insert  $O_{new}$  into  $C_1$  and  $C_2$ 
13  Select  $(X, Y) \in C_2$  maximizing  $w(X, Y)$ 
14   $F \leftarrow F \cup \{(X, Y)\}$ 
15   $R_{un} \leftarrow R_{un} \setminus (X \times Y)$ 
16 return  $F$ 

```

Figure4 – Pseudocode of the RSF algorithm [3]

One iteration of the main loop has a time complexity of $O(|B|^3|A|)$. Since each iteration covers at least one previously uncovered incidence, this loop is executed at most $|R|$ times.

4 Application of the algorithm to factor detection

To apply the RSF algorithm, we used a dataset from the 'NIST Data Leakage Case' dataset, which was aggregated using a time window of 1,800 seconds. Detailed pre-processing of the dataset is described in output KPB2-02. Individual rows of the dataset represent time clusters of events generated within this interval.

Most attributes are already binary in nature, with a value of 1 indicating the presence of the relevant attribute within a given time cluster and a value of 0 indicating its absence. Prior to applying the algorithm, five non-binary attributes containing temporal, identification or auxiliary information were removed. The result of the pre-processing is a binary matrix $R \in \{0,1\}^{276 \times 218}$, which is subsequently used as input to the RSF algorithm.

The algorithm's output consists of **86 factors**. Each factor corresponds to a single formal concept and represents a specific combination of attributes common to a group of objects. The complete output is provided in the appendix NIST_factors_1800s.xlsx, where the individual factors, the attributes characterising each factor, the number of attributes and the number of objects in the factor are listed in separate rows.

Individual factors differ in both the number of attributes in the intent and the number of objects in the extent. In the resulting decomposition, the factors contain between 1 and 107 attributes and between 1 and 220 objects.

4.1 General and specific factors

Factors with a small number of attributes often capture more general patterns that occur across a larger number of objects. Conversely, factors with a larger number of attributes generally represent more specific combinations of properties and contain a smaller number of objects. This relationship stems from the properties of formal concepts – the more attributes an intent contains, the fewer objects possess all the selected attributes. For example, factor 83 contains 1 attribute and 220 objects, whilst factor 41 contains 107 attributes and only 1 object. However, neither the number of attributes nor the number of objects in itself determines the significance of a factor. When interpreting the results, it is particularly important to take into account the significance of the attributes and the factor's relevance to the security scenario under analysis.

We consider factors with the largest number of objects in the extent to be the most general. Such factors capture attributes or combinations of attributes that occur in a large proportion of the analysed time clusters. They represent frequently recurring patterns and therefore are unlikely to indicate non-standard or suspicious behaviour. However, they provide a basic

picture of the dominant attributes in the analysed dataset. In *Chyba! Nenašiel sa žiaden zdroj odkazov.* we list the three most general factors.

Factor No.	Number of attributes	Attributes	Number of objects
83	1	mask_MSIE_WebCache_container_record	220
2	2	is_url_visit, mask_MSIE_WebCache_container_record	213
5	1	mask_MSIE_Cache_File_URL_record	103

Table 1– The most general factors in the analysed dataset

We consider factors with a small number of objects in the extent to be specific. Such factors capture combinations of attributes that occur only rarely in the dataset. Their significance from the perspective of digital forensic analysis depends on the specific attributes within the factor's extent, for example, the presence of attributes related to script execution, registry changes, access to sensitive files or other suspicious activities. In the obtained decomposition, 15 factors contain only 1 object. In **1** we present 3 of them.

Factor No.	Number of attributes	Attributes	Number of objects
41	107	access_count_gt_1, contains_cab, contains_inf, contains_update_task, contains_vbs, ContainsCmdCommand, ContainsScriptExtension, event_registry, has_exe_target, has_macb_accessed, has_macb_modified, has_modified_flag, has_modified_time, has_sha256, has_transition_type, HasAandBNoMC, is_accessed, is_admin_activity, is_advertising_cookie, is_analytics_cookie, is_apis_google, is_application_experience, is_bing_domain, is_error_event, is_executable, is_failed_login, is_file, is_file_entry, is_filename_attr, is_fileshare_access, is_from_driverstore, is_from_ie, is_from_pinned_location, is_from_system32, is_from_users_dir, is_google_domain, is_gstatic_domain, is_high_access_count, is_history_container, is_https, is_image_file, is_in_appdata, is_in_system32, is_in_temp_or_spool, is_in_winsxs, is_link_file, is_login_event, is_modified_after_access, is_network_related, is_ntfs, is_policy_change, is_privilege_change, is_repeated_record, is_root_entry, is_script_file, is_service_event, is_standard_info, is_suspicious_domain, is_suspicious_task_name, is_system_path, is_terminal_services, is_third_party, is_tracking_cookie, is_unallocated, is_unpinned, is_unusual_location, is_url_visit, is_user_profile_shortcut, is_xml_file, is_youtube_domain, IsCSS, IsDynamicLibrary, IsInAppData, IsSystemHive, IsUserHive, key_autorun_related, key_contains_winlogon, key_new_registry_branch, key_suspicious_value, macb_anomaly, macb_is_only_access_birth, MissingMInMACB, ModifiesStartType, task_starts_at_logon, visit_to_search_engine, visited_microsoft, visited_msn, mask_WinEVTX, mask_File_entry_shell_item, mask_NTFS_file_stat, mask_File_stat, mask_Windows_Shortcut, mask_AppCompatCache_Registry_Key, mask_BagMRU_Registry_Key, mask_Chrome_Cache, mask_Chrome_Cookies, mask_Chrome_History,	1

		mask_MRUListEx_Registry_Key, mask_MSIE_WebCache_container_record, mask_MSIE_WebCache_containers_record, mask_MSIE_Cache_File_URL_record, mask_OLECF_Dest_list_entry, mask_OLECF_Item, mask_Registry_Key, mask_System, mask_Setup_API_Log, mask_Task_Cache_Registry_Key	
34	90	access_count_gt_1, contains_inf, contains_path_recent, ContainsPowershellCommand, event_registry, has_hostname_reference, has_macb_accessed, has_macb_modified, has_modified_flag, has_modified_time, has_sha256, has_transition_type, is_accessed, is_admin_activity, is_analytics_cookie, is_apis_google, is_application_experience, is_error_event, is_executable, is_failed_login, is_file, is_file_entry, is_filename_attr, is_from_system32, is_from_users_dir, is_google_domain, is_gstatic_domain, is_high_access_count, is_https, is_image_file, is_in_appdata, is_in_system32, is_in_temp_or_spool, is_link_file, is_login_event, is_network_share, is_ntfs, is_policy_change, is_privilege_change, is_root_entry, is_runonce_key, is_script_file, is_service_event, is_standard_info, is_suspicious_domain, is_system_location, is_system_path, is_terminal_services, is_tracking_cookie, is_unallocated, is_unpinned, is_unusual_location, is_url_visit, is_user_account_event, is_user_profile_shortcut, is_xml_file, is_youtube_domain, IsCSS, IsInAppData, IsSuspiciousPath, IsSystemHive, IsUserHive, key_autorun_related, key_contains_winlogon, key_new_registry_branch, key_suspicious_value, macb_is_only_access_birth, MissingMinMACB, ModifiesImagePath, ModifiesStartType, opened_control_panel, unknown_shell_item_type, mask_WinEVTX, mask_File_entry_shell_item, mask_NTFS_file_stat, mask_File_stat, mask_Windows_Shortcut, mask_BagMRU_Registry_Key, mask_Chrome_Cache, mask_Chrome_Cookies, mask_Chrome_History, mask_MRUList_Registry_Key, mask_MRUListEx_Registry_Key, mask_MSIE_WebCache_container_record, mask_OLECF_Dest_list_entry, mask_OLECF_Item, mask_Registry_Key, mask_Run_Run_Once_Registry_Key, mask_System, mask_User_Account_Information_Registry_Key	1
43	84	access_count_gt_1, accessed_local_file, contains_cab, contains_inf, has_exe_target, has_macb_accessed, has_macb_modified, has_modified_flag, has_sha256, is_accessed, is_application_experience, is_content_container, is_executable, is_file, is_file_entry, is_filename_attr, is_fileshare_access, is_frequent_execution, is_from_system32, is_from_users_dir, is_high_access_count, is_history_container, is_in_appdata, is_in_system32, is_in_temp_or_spool, is_in_winsxs, is_link_file, is_login_event, is_modified_after_access, is_multiple_volumes, is_network_path, is_ntfs, is_obfuscated_path, is_privilege_change, is_rare_app, is_repeated_record, is_root_entry, is_run_often, is_service_event, is_specific_container, is_standard_info, is_suspicious_extension, is_system_hive, is_system_location, is_system_path, is_system_process, is_temp_launch, is_unallocated, is_url_visit, is_user_profile_shortcut, is_xml_file, IsDynamicLibrary, IsInAppData, IsSystemHive,	1

		IsUserHive, key_autorun_related, key_contains_winlogon, key_new_registry_branch, key_suspicious_value, macb_is_only_access_birth, MissingMInMACB, ModifiesImagePath, ModifiesStartType, opened_control_panel, opened_recycle_bin, unknown_shell_item_type, UsesQuotesInPath, mask_WinEVTX, mask_File_entry_shell_item, mask_NTFS_file_stat, mask_File_stat, mask_Windows_Shortcut, mask_AppCompatCache_Registry_Key, mask_BagMRU_Registry_Key, mask_MRUListEx_Registry_Key, mask_MSIE_WebCache_container_record, mask_MSIE_WebCache_containers_record, mask_MSIE_Cache_File_URL_record, mask_OLECF_Item, mask_Registry_Key, mask_Run_Run_Once_Registry_Key, mask_Setup_API_Log, mask_User_Account_Information_Registry_Key, mask_WinPrefetch	
--	--	--	--

1 table – Factors with a small number of objects in an extent

4.2 Factors related to attacks

The significant attribute *'is_attack_related'* is found in 4 of the extracted factors.

The most general of these, factor 24, contains 2 attributes and 21 objects. This factor does not provide a detailed description of the attack, but captures a common characteristic of a larger group of labelled objects.

The extents of the other 3 factors each contain 1 object, whilst their intents consist of between 50 and 60 attributes. These intents include combinations of attributes derived from file and web artefacts, Windows Registry artefacts and system events. Factor 30 primarily contains attributes relating to access to sensitive files and network sharing, whilst factor 32 includes attributes associated with the execution of commands and scripts and with changes to autostart mechanisms. These factors thus capture specific combinations of characteristics present in time clusters flagged as attack-related. In **2** we list the attack-related factors described above.

Factor No.	Number of attributes	Attributes	Number of objects
24	2	mask_MSIE_WebCache_container_record, is_attack_related	21
27	50	access_count_gt_1, accessed_local_file, has_macb_accessed, has_macb_modified, has_modified_flag, has_sha256, has_transition_type, is_accessed, is_apis_google, is_application_experience, is_file, is_file_entry, is_filename_attr, is_from_system32, is_from_users_dir, is_google_domain, is_gstatic_domain, is_https, is_image_file, is_in_appdata, is_in_system32, is_in_temp_or_spool, is_login_event, is_modified_after_access, is_ntfs, is_policy_change, is_privilege_change, is_runonce_key, is_script_file, is_service_event, is_standard_info, is_suspicious_domain, is_system_hive, is_unallocated, is_url_visit, is_xml_file, IsCSS, IsSystemHive, IsUserHive, ModifiesStartType, mask_WinEVTX, mask_NTFS_file_stat, mask_File_stat, mask_Chrome_Cache, mask_Chrome_History,	1

		mask_MSIE_WebCache_container_record, mask_Registry_Key, mask_Run_Run_Once_Registry_Key, mask_Setup_API_Log, is_attack_related	
30	60	access_count_gt_1, accessed_local_file, accessed_sensitive_file, contains_secret_path, has_macb_accessed, has_macb_modified, has_modified_flag, has_sha256, has_transition_type, is_accessed, is_application_experience, is_file, is_file_entry, is_filename_attr, is_from_users_dir, is_google_domain, is_gstatic_domain, is_high_access_count, is_https, is_image_file, is_in_appdata, is_in_temp_or_spool, is_modified_after_access, is_network_share, is_ntfs, is_obfuscated_path, is_root_entry, is_run_often, is_script_file, is_sensitive_file, is_service_event, is_specific_container, is_standard_info, is_suspicious_domain, is_suspicious_keyword, is_system_location, is_unpinned, is_url_visit, is_xml_file, IsInAppData, IsSystemHive, IsUserHive, key_contains_run, key_new_registry_branch, opened_control_panel, unknown_shell_item_type, mask_WinEVTX, mask_File_entry_shell_item, mask_NTFS_file_stat, mask_File_stat, mask_BagMRU_Registry_Key, mask_Chrome_Cache, mask_Chrome_History, mask_MSIE_WebCache_container_record, mask_MSIE_Cache_File_URL_record, mask_OLECF_Dest_list_entry, mask_OLECF_Item, mask_Registry_Key, mask_WinPrefetch, is_attack_related	1
32	51	access_count_gt_1, accessed_local_file, contains_cab, contains_cur, contains_inf, contains_ppd, contains_vbs, ContainsCmdCommand, ContainsPowershellCommand, has_macb_accessed, has_macb_modified, has_modified_flag, has_sha256, is_accessed, is_executable, is_file, is_file_entry, is_filename_attr, is_from_driverstore, is_from_system32, is_from_users_dir, is_in_appdata, is_in_system32, is_in_temp_or_spool, is_in_winsxs, is_modified_after_access, is_network_path, is_ntfs, is_runonce_key, is_script_file, is_standard_info, is_suspicious_extension, is_unallocated, is_url_visit, is_xml_file, IsDynamicLibrary, IsSystemHive, IsUserHive, key_autorun_related, key_contains_winlogon, key_new_registry_branch, key_suspicious_value, ModifiesImagePath, ModifiesStartType, mask_NTFS_file_stat, mask_File_stat, mask_MSIE_WebCache_container_record, mask_Registry_Key, mask_Run_Run_Once_Registry_Key, mask_System, is_attack_related	1

2 Table – Attack-related factors

Conclusion

The method for identifying relationships in digital forensic analysis data utilises matrix factorisation based on formal concepts and the RSF algorithm, which combines greedy selection with the iterative generation of candidate formal concepts. The result is an accurate decomposition of the input incidence matrix, which enables the identification of groups of objects characterised by common attributes.

The method identifies relationships between temporal clusters of events and their properties, and captures both general and specific combinations of attributes present in the analysed data. Factors with fewer attributes generally describe more general patterns present across a larger number of objects. Conversely, factors with a greater number of attributes capture more specific combinations of properties, and their extents are usually smaller. When analysing the factors obtained, we can focus on a selected significant attribute and observe which other attributes it occurs with in individual factors. Another important benefit of the proposed method is the interpretability of the factors identified.

The proposed method provides a systematic and interpretable way of investigating the relationships between forensic events and their attributes.

Bibliography

- [1] Ganter, B., Wille, R. (2024). *Formal Concept Analysis: Mathematical Foundations*. Springer.
<https://doi.org/10.1007/978-3-031-63422-2>
- [2] Belohlavek, R., Vychodil, V. (2010). *Discovery of optimal factors in binary data via a novel method of matrix decomposition*. Journal of Computer and System Sciences.
<https://doi.org/10.1016/j.jcss.2009.05.002>
- [3] Antoni, Ľ., Kotlárová, D., Krídlo, O., López-Rodríguez, D., Ojeda-Aciego, M. (2026). *Effective greedy Boolean matrix factorisation via the Rice-Siff algorithm*. International Journal of Approximate Reasoning.
<https://doi.org/10.1016/j.ijar.2026.109747>

Appendices

ADFIR-D20-KPB4-01-NIST_factors_1800s.xlsx