FACULTY OF
LAW AND CRIMINOLOGY

Harmonising the Fight Against Ransomware:

Technological and National Implementation Challenges in Cybercrime Frameworks

Word count: 2500

Ainhoa Zamora Martínez

Student number: 02415847

B001717A - Selected Issues: Cybercrime, surveillance & technology

International Master in Advanced Research in Criminology (IMARC)

Academic year: 2024 – 2025

GHENT
UNIVERSITY

LIST OF CONTENTS

1. Problem statement **3**

2. Research Questions **3**

3. Methodology **4**

3.1. Sources and Search Strategies 4

3.2. Considerations 5

4. Content Analysis **5**

4.1. Technological Adaptations: dominant categories and new terminologies 7

4.2. National Implementation: harmonisation and oversight 9

5. Discussion and Conclusion **10**

6. Bibliography **13**

6.1. Primary Sources 13

6.2. Secondary Sources 13

7. Appendix **16**

LIST OF FIGURES

Image 1. (Some) Subcategories for the Technological adaptation dimension 4

Image 2. (Some) Subcategories for the National implementation dimension 5

Table 1. Evolution of Ransomware Framing (2001–2022) 6

Graph 1. References to Ransomware Technological Evolution 8

1. Problem statement

Over the past two decades, ransomware has evolved from an opportunistic malware distributed via spam emails to a sophisticated global threat. Modern cybercriminals leverage advanced encryption techniques, anonymous payment like cryptocurrencies, and worm-like propagation to maximise disruption and profit.[1] High-profile attacks on critical infrastructure, healthcare systems, and governments have exposed vulnerabilities in digital ecosystems, costing billions annually and undermining public trust. [2]

The propagation methods of these attacks have taken place via “remote desktop protocols that do not rely on any form of user interaction”,[3] demanding payments from users in cryptocurrencies, which are less regulated and harder to control. But the problem comes from the broad target affecting and invading every industry and private individuals.[4] To combat this threat, countries have been implementing cybercrime legislation to make front to the various forms of online criminal activities. According to a United Nations Conference on Trade and Development report (UNCTAD), 156 countries, representing 80% of the global community, have implemented cybercrime legislation.[5]

2. Research Questions

Despite this escalating crime, international and EU cybercrime frameworks—the Budapest Convention on Cybercrime, EU Directive 2013/40/EU, NIS 2 Directive (EU) 2022/2555, and the T-CY Ransomware Guidance Note—struggle to keep pace with ransomware’s technological evolution and varying national implementations, creating gaps in prevention, criminalisation and response. This paper aims to examine how these international and EU cybercrime frameworks frame ransomware as a criminological challenge amidst its technological evolution from 2001 until today? Further, it will assess to what extent do these frameworks adapt to ransomware’s advancements (e.g., encryption, AI), and how does national implementation influence their ability to counter it?

Analysing these frameworks is vital for understanding whether legal responses align with ransomware’s dynamic threat landscape and for informing policies to address gaps in prevention, criminalisation, and response. This paper argues that while these frameworks increasingly recognise ransomware complexity, inconsistencies in addressing technological advancement and variations in national implementation limit their criminological impact.

[1] Mohiuddin Ahmed, *Ransomware Evolution* (CRC Press 2024). p.3.

[2] Aaron Zimba and Mumbi Chishimba, ‘Understanding the Evolution of Ransomware: Paradigm Shifts in Attack Structures’ (2019) 11 International Journal of Computer Network and Information Security p.26.

[3] Sharon Shea and Alissa Irei, ‘What Is Ransomware? How It Works and How to Remove It’ (*Search Security*) <https://www.techtarget.com/searchsecurity/definition/ransomware> accessed 15 April 2025.

[4] Asaf Lubin, ‘The Law and Politics of Ransomware’ (2022) 55 Vanderbilt Journal of Transnational Law 1177 <https://heinonline.org/HOL/P?h=hein.journals/vantl55&i=1223> accessed 15 April 2025. p.1185.

[5] Lika Chimchiuri, ‘the Evolution Of Cybercrime Legislation’ (2024) 2 Scientific works of National Aviation University. Series: Law Journal ‘Air and Space Law’ <https://jrnl.nau.edu.ua/index.php/UV/article/view/18813> accessed 15 April 2025. p.221

3. Methodology

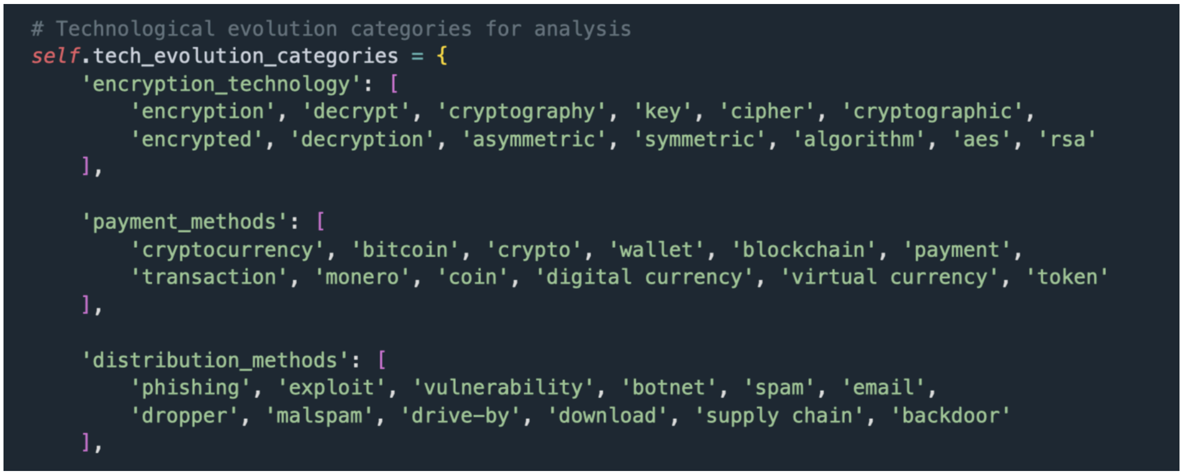
This paper employs a mixed-methods approach that integrates computational content analysis, combining natural language processing techniques with comparative legal analysis to extract insights from the selected legal instruments. The analysis focuses on two primary dimensions: *technological adaptation* (how frameworks address ransomware's evolving technical characteristics) and *national implementation* (how jurisdictions operationalise these frameworks)*.* The methodology leverages Python within a Google Colab environment, allowing for reproducible research, a custom framework uses libraries including Pandas for data manipulation, NLTK for natural language processing, and Matplotlib/Seaborn for visualisation. Additionally, Claude (GenAI) model Sonnet 3.7 was used to assist in creating and refining the Python scripts, enhancing the accuracy of the coding process.[6]

3.1. Sources and Search Strategies

The corpus consists of the full texts of the 4 instruments, sourced from the Council of Europe (Budapest Convention and T-CY) and EUR-lex (EU Directive 2013/40/EU, NIS2 Directive). Later, a comprehensive keyword list was developed through a three-step process: (1) manual reading of documents to identify and select recurring terms, (2) term frequency analysis using NLTK prioritising high-relevance terms, and (3) refinement with synonyms and context-specific variants.

The two dimensions are operationalised through detailed subcategories. For each category, a comprehensive set of terms will capture relevant concepts, synonyms, and technical/legal terminology. Using pattern matching and contextual analysis, this paper will be able to identify, quantify and, analyse systematically how these legal instruments adapt to the evolving ransomware threat landscape. Below are some examples of both dimensions: [7]

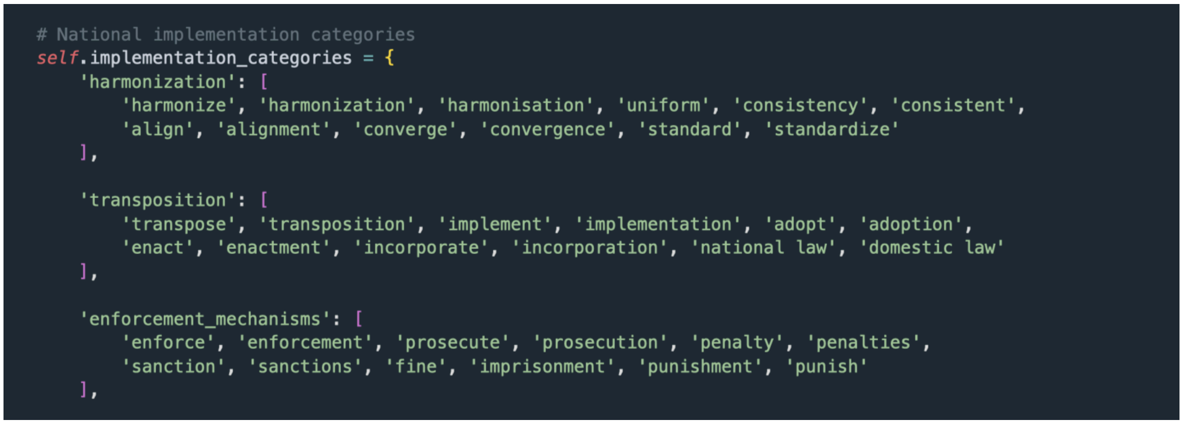
Image 1. (Some) Subcategories for the Technological adaptation dimension



```
# Technological evolution categories for analysis
self.tech_evolution_categories = {
    'encryption_technology': [
        'encryption', 'decrypt', 'cryptography', 'key', 'cipher', 'cryptographic',
        'encrypted', 'decryption', 'asymmetric', 'symmetric', 'algorithm', 'aes', 'rsa'
    ],

    'payment_methods': [
        'cryptocurrency', 'bitcoin', 'crypto', 'wallet', 'blockchain', 'payment',
        'transaction', 'monero', 'coin', 'digital currency', 'virtual currency', 'token'
    ],

    'distribution_methods': [
        'phishing', 'exploit', 'vulnerability', 'botnet', 'spam', 'email',
        'dropper', 'malspam', 'drive-by', 'download', 'supply chain', 'backdoor'
    ],
```

[6] Please refer to Appendix 1 to see the full Python script.
[7] Full categorisation of each dimension can be found in Appendix 1.

Image 2. (Some) Subcategories for the National implementation dimension



```
# National implementation categories
self.implementation_categories = {
    'harmonization': [
        'harmonize', 'harmonization', 'harmonisation', 'uniform', 'consistency', 'consistent',
        'align', 'alignment', 'converge', 'convergence', 'standard', 'standardize'
    ],

    'transposition': [
        'transpose', 'transposition', 'implement', 'implementation', 'adopt', 'adoption',
        'enact', 'enactment', 'incorporate', 'incorporation', 'national law', 'domestic law'
    ],

    'enforcement_mechanisms': [
        'enforce', 'enforcement', 'prosecute', 'prosecution', 'penalty', 'penalties',
        'sanction', 'sanctions', 'fine', 'imprisonment', 'punishment', 'punish'
    ],
```

3.2. Considerations

While this methodological approach offers several advantages, including achieving quantitative precision and qualitative depth, while maintaining reproducibility, it is important to consider potential limitations. Computerised procedures used in automated coding, including those by Claude Sonnet 3.7, might introduce systemic bias in the coding of texts. According to Matthes and Kohring (2008), all computer-assisted methods for coding assume that words have the same meaning in different contexts, while human coders are able to make distinctions.[8] To mitigate this, the keyword refinement process incorporated contextual analysis to enhance its robustness.[9] This included checking specific sections or paragraphs where the keywords are likely to be relevant to the selected dimensions. For instance, when looking at national implementation measures, the focus was on sections discussing enforcements or penalties.

4. Content Analysis

The analysis reveals a clear evolution in how ransomware is framed across the documents from 2001 to 2022. Quantitative analysis shows that technical framing terms decreased from 67% of total framing references in early documents to 38% in more recent frameworks, while economic, societal, and governance framings collectively increased from 15% to 48%. These changes reflect a broader understanding that ransomware involves more than just malicious software, as well as indicating a growing awareness of ransomware's non-technical dimensions. With financial motives and economic impacts, driven by sophisticated ransomware models and

[8] Rutger de Graaf and Robert van der Vossen, 'Bits versus Brains in Content Analysis. Comparing the Advantages and Disadvantages of Manual and Automated Methods for Content Analysis' (2013) 38 p.439. <https://doi.org/10.1515/commun-2013-0025> accessed 15 April 2025.

[9] Dan Jurafsky and James H Martin, *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition with Language Models* (3 ed, Stanford 2025) <https://web.stanford.edu/~jurafsky/slp3/> accessed 16 April 2025. p.231.

cryptocurrencies, or e.g. societies' essential services in healthcare [10] and governance framings, in coordinated legal and policy responses.

Explicit mentions of ransomware increased dramatically over time, with the T-CY Guidance Note (2022) containing 126 direct mentions of 'ransomware' compared to 0 mentions in the Budapest Convention (2001). There is also a shift from generic to specific framings; while Budapest primarily uses terms like 'illegal access', 'data interference', and 'system interference'[11], these are without specific mention to encryption-based extortion. On another side, the 2013 Directive begins to acknowledge 'botnets'[12]. And the T-CY marked a change, explicitly defining ransomware as a malware that encrypts data or systems to demand ransoms, introducing terms such as 'decryption', 'cryptocurrency' and 'Raas'. This specificity mirrors ransomware's evolution into a professionalised crime, with organised groups operating, including 24/7 victim support and dark-web payment systems. The NIS 2 Directive, also in 2022, networked systems and risk management, addressing ransomware's threat to interconnected infrastructures.

To illustrate this evolving framing across key frameworks, the following table summarises the shift in terminology and criminological perspectives. It shows how international and EU cybercrime framework have reframed ransomware, capturing its shift from a vague cyberthreat to a complex criminological issue.

Table 1: Evolution of Ransomware Framing (2001–2022)

Framework	**Year**	**Framing of Ransomware**	**Key Terms**	**Criminological Framing**	**Technological Context**
Budapest Convention	2001	Not explicitly defined, but applicable through provisions on data and system interference	Illegal access, data interference	Generic cybercrime; criminal justice focus	Basic ransomware (e.g., floppy-disk extortion)
EU Directive	2013	Not explicitly defined, but addresses ransom-ware related activities.	Botnets, large-scale attacks	Acknowledges sophisticated attacks; still broad	Rise of crypto-ransomware (e.g., CryptoLocker)

[10] G Kirubavathi, W Regis Anne and UK Sridevi, 'A Recent Review of Ransomware Attacks on Healthcare Industries' (2024) 15 International Journal of System Assurance Engineering and Management 5078 <https://doi.org/10.1007/s13198-024-02496-4>.
[11] Budapest Convention (Art. 2-5)
[12] Directive 2013/40/EU (Art. 5)

T-CY Guidance Note	2022	Malware designed to deny user access to data or systems by encryption.	Ransomware, encryption, cryptocurrency, RaaS	Specific; technical, economic, societal; lifecycle focus	Advanced RaaS, cloud attacks, critical infrastructure
NIS 2 (EU) Directive	2022	Ransomware as a significant cybersecurity threat requiring risk management and incident reporting.	Network, risk management	Proactive; societal, cross-border cooperation	Networked system, interconnected threats

Source: Own elaboration

With this overview, the following sections explore the specific framing categories in more detail.

4.1. Technological adaptations: dominant categories and new terminologies

Criminologically, frameworks show distinct shifts. Technological framings are more dominant across all frameworks, but it has evolved from static 'computer' and 'data' (300 mentions in 2001) to networked systems and encryption (127 mentions in 2022 T-CY). Allowing a visible shift from standalone PC attacks to cloud targeting infrastructure and critical systems. Its criminal framing decreases relatively (169 mentions in 2001 to 110 in 2022) as focus diversifies. Newer terms such as 'extortion' and 'blackmail' better capture ransomware's coercive nature, moving beyond generic offences. Furthermore, early frameworks ignored societal impacts, but by 2022, the T-CY Guidance Note started to link ransomware to critical infrastructure attacks (e.g. hospitals, Costa Rica's emergency) and societal harms (healthcare disruptions, national security), reflecting real-world consequences as attacks occur in interconnected systems.

Further, Budapest Convention lacks ransomware-specific terminology, using generic terms like 'data' (92 mentions), and 'access' (14 mentions). While other documents such as the T-CY Guidance Note introduces highly specific ransomware terms: 'Ransomware' (126 mentions), 'Ransom' (12 mentions), 'Encrypted/encryption', 'Cryptocurrency' and 'Bitcoin' (mentioned as payment method), and 'Malware' (specific to ransomware function).

In addition, other technological adaptation patterns demonstrate a clear progression in how legal frameworks address ransomware's components. With the earliest instruments, Budapest showing minimal engagement with encryption technologies (score: 1)[13] and payment methods (1), reflects the relatively nascent state of ransomware at that time. It also explains how digital

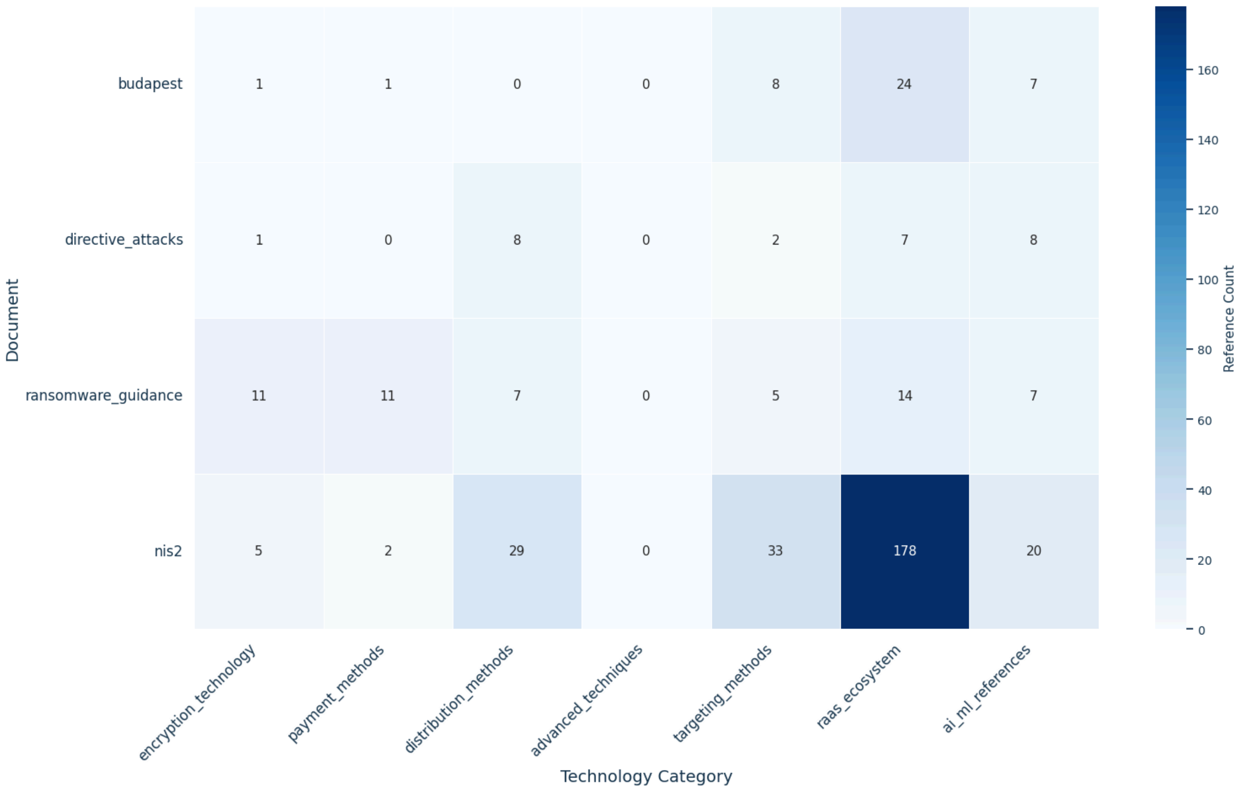
[13] The *'score'* quantifies the emphasis a legal framework places on ransomware-related concepts, based on the frequency and specificity of references. For example, a score of 1 indicates minimal mention, a score of 11 reflects significant focus. Throughout the paper, scores are shortened to the number in brackets, e.g., (11).

extortion was conceptualised within broader cyber crime categories rather than as a unique criminological challenge. In the same way, the 2013 Directive displays limited reference to encryption (1) and no significant addressing of payment methods done (0).

Further, an increase in references to encryption and payment methods in the 2022 T-CY, both scoring 11, represented a fundamental shift in the criminological framing of ransomware. Acknowledging that the technical mechanisms of ransomware—particularly encryption—are not merely incidental to the crime but constitute its defining characteristics. Also, the increased focus on payment methods further indicates an evolution in how legal frameworks understand the economic dimension of ransomware crime. Reflecting that tracking and disrupting payment flows represents a critical point in combating ransomware operations—an important shift from earlier frameworks that largely ignored this aspect of the criminal enterprise.

However, most revealing is the NIS2 extensive coverage of the Ransomware-as-a-Service ecosystem (178), which represents a fundamental reconceptualisation of ransomware from an individualised crime to an organised criminal industry. It helps to frame that modern ransomware operations work as sophisticated criminal businesses. This document also had high scores for distribution (29) and targeting methods (33), indicating a more nuanced understanding of how ransomware actors select and compromise victims.

Graph 1: References to Ransomware Technological Evolution



Source: Own design, generated using Python on Google Colab

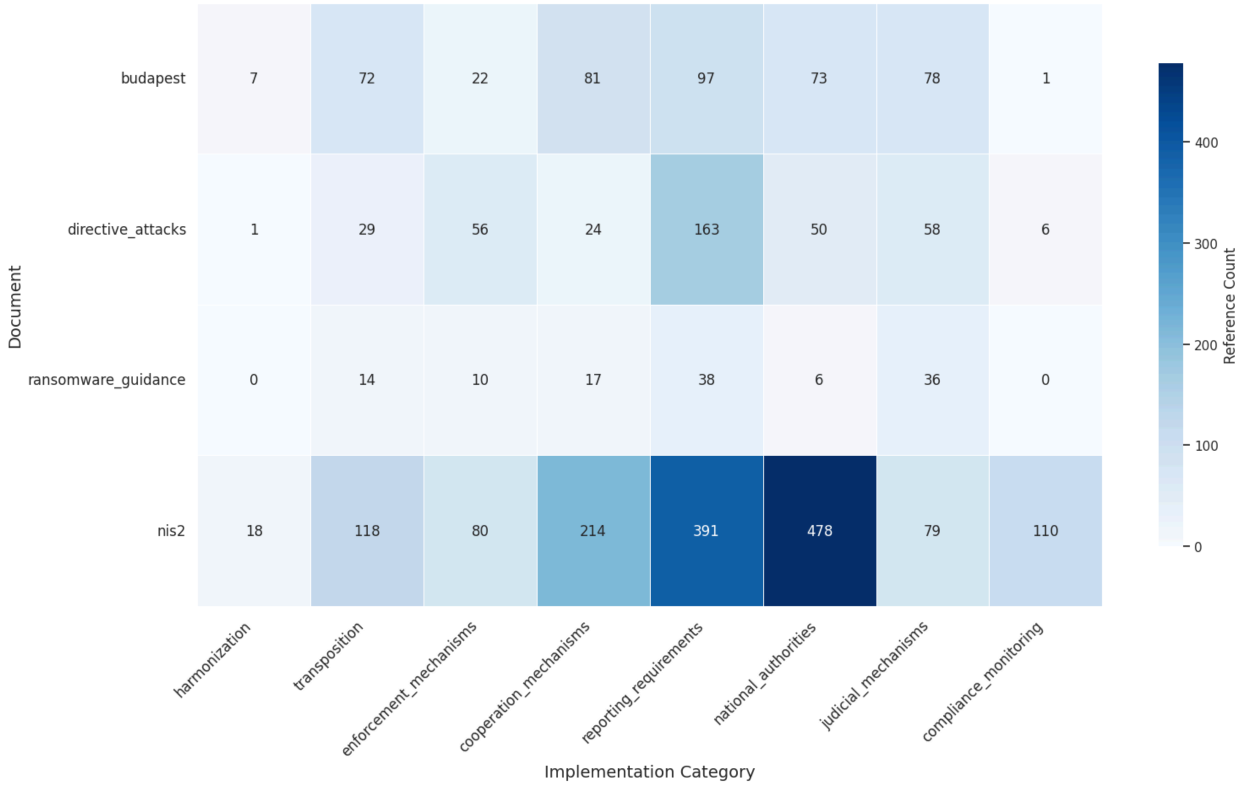
4.2. National Implementation: harmonisation and oversight

The implementation matrix from my python script reveals significant variations in how these frameworks approach the practical aspects of countering ransomware. However, the data points to a clear progression towards more comprehensive implementation mechanisms in newer frameworks.

The harmonisation efforts show a steady increase across the chronological development of these instruments. Here, the Convention contains modest harmonisation provisions (score 7.0), while the NIS 2 demonstrates a substantially stronger emphasis on harmonising approaches across jurisdictions (18). Interestingly, the Ransomware Guidance shows no significant harmonisation provisions (0), suggesting it functions more as a technical guidance rather than an instrument for legal harmonisation. Perhaps, most striking is the dramatic increase in provisions relating to national authorities in the NIS 2 (478) compared to earlier frameworks like the Budapest Convention (73) and Directive on Attacks (50). This substantial increase indicates a recognition that effective ransomware countermeasures require robust national institutional frameworks with clear mandates and capabilities.

Furthermore, compliance monitoring mechanisms show a similar pattern of evolution. The Budapest convention contains minimal compliance monitoring provisions (1), while the NIS 2 includes over one hundred times more references to compliance monitoring (110). This suggests a shift from voluntary implementation toward more structured oversight mechanisms to ensure consistent application of countermeasures of ransomware attacks. The analysis also revealed a change in the priorities in implementation approaches over the time. The Budapest Convention emphasises judicial mechanisms (78), reflecting its focus on establishing foundational legal frameworks for addressing cybercrime. The 2013 Directive, on the other side, places greater emphasis on reporting requirements (163), indicating a shift toward information gathering as a key aspect of effective response. Lastly, the NIS document maintains this focus on reporting (391), while significantly expanding provisions for national authorities and cooperation mechanisms (214).

Graph 2: References to national implementation approaches



Source: Own design, generated using Python on Google Colab

5. Discussion and Conclusion

The shift in ransomware framing—from generic cybercrime in the Budapest Convention to explicit, multidimensional definitions in the T-CY and 2022 Directive—reflects growing awareness of its criminological scope. Evolving from a narrow, technical focus to a broader approach, incorporating economic, societal, and governance dimensions. For instance, while NIS 2 addresses societal harms, earlier frameworks prioritise technical violations, ignoring victim's psychological and economic trauma.[14] Furthermore, the increase in references to national authorities in the documents show how cybersecurity evolved from primarily a law enforcement issue to an important governance challenge requiring specialised authorities.[15] Not only this, also reporting systems and compliance regimes portray the criminological challenge needing judicial mechanisms, shifting from purely criminal justice approaches to regulatory frameworks.

[14] Jamie MacColl and others, 'Ransomware: Victim Insights on Harms to Individuals, Organisations and Society' <https://www.rusi.orghttps://www.rusi.org> accessed 16 April 2025.

[15] FATF (Financial Action Task Force), 'Countering Ransomware Financing' <http://www.fatf-gafi.org/publications/Methodsandtrends/countering-ransomware-financing.html> accessed 16 April 2025.

The extent to which these frameworks adapt to ransomware's advancements is further illustrated by the increasing focus on encryption and payment methods. The analysis has revealed a significant increase in references to both in the T-CY compared to the 2001 convention and the 2013 Directive. This acknowledges the technical mechanisms of ransomware—particularly encryption, cryptocurrency, AI and automation. Here, the convention's data interference provisions[16] and Directive 2013/40/EU's system interference rules[17] remain applicable to encryption-based ransomware, as they focus on outcomes rather than specific technologies, and neither explicitly addresses specific decryption challenges or complexities from cryptocurrencies. These earlier frameworks rely more on investigative tools such as data preservation.[18] For AI-driven ransomware, neither the 2001 nor the 2013 frameworks have the necessary provisions for AI-specific threats, though their broad language allows interpretation. In contrast, the NIS Directive adapts better to modern threats by mandating proactive risk assessments,[19] demonstrating a more forward-thinking approach to cybersecurity.

Additionally, the guiding EU core values and fundamental rights such as freedom of expression and the right to privacy and protection, and the promotion of the open, free and secure cyberspace,[20] guide its cybercrime frameworks. However, national implementation significantly shapes these framework's effectiveness. While the Convention on Cybercrime, as a treaty, depends on domestic legislation for success, countries with robust cybercrime laws (e.g. the U.S with CFAA) align well. But uneven adoption, especially in jurisdictions with limited resources creates enforcement gaps. For instance, this aligns with Nguyen and Golman's paper where it was found that Pacific Island Countries, like Vanuatu, lack human resources and technical ability in cybersecurity and ICT, impeding an effective law enforcement against cybercrime.[21]

Further, EU member states must transpose directives, but variations in implementation and enforcement persist across the Union. For instance, Germany's stringent cybersecurity laws enhance NIS 2 compliance[22], while smaller states may lag in technical capacity, leading to uneven application. Therefore, Directive 2013/40/EU's minimum penalties[23] aims to ensure a degree of consistency, considering that prosecution rates and judicial priorities still differ

[16] Convention on Cybercrime of the Council of Europe of 23 November 2021, European Treaty Series - No.185. (Budapest Convention) (Art. 49)

[17] European Parliament and Council Directive 2013/40/EU of 12 August 2013 on attacks against information systems and replacing Council Framework Decision 2005/222/JHA [2013] OJ L. 218
(Art. 6)

[18]Art. 16 Budapest Convention

[19] Cybercrime Convention Committee (2022) T-CY Guidance Note #12 Aspects of ransomware covered by the Budapest Convention (T-CY(2022)14). Council of Europe. https://rm.coe.int/t-cy-2022-14-guidancenote-ransomware-v4adopted/1680a9355e (Art. 17.)

[20] Commission, 'Join Communication to the European Parliament and the Council. Resilience, Deterrence and Defence: Building strong cybersecurity for the EU' JOIN(2017) 450 final.

[21] Dr Chat Le Nguyen and Dr Wilfred Golman, 'Diffusion of the Budapest Convention on Cybercrime and the Development of Cybercrime Legislation in Pacific Island Countries: "Law on the Books" vs "Law in Action"' (2021) 40 Computer Law & Security Review 105521 p.12. <https://www.sciencedirect.com/science/article/pii/S0267364920301266>.

[22] Béla Droppa, 'NIS2 in Germany: What You Need to Know and Where to Start' (*Black Cell*, 10 April 2025) <https://blackcell.io/nis2-in-germany-what-you-need-to-know-and-where-to-start/> accessed 15 April 2025.

[23] Art. 7 Directive 2013/40/EU

between member states. Finally, inconsistent forensic capabilities and training across jurisdictions hinder ransomware investigations, particularly when dealing with sophisticated AI-driven attacks. The NIS 2 reporting requirements[24] aims to standardise responses and improve cooperation, but national agencies like ENISA note ongoing challenges and delays in harmonisation.[25, 26]

Looking ahead, while progress has been made in framing and addressing ransomware, the path forward requires a stronger emphasis on achieving greater harmonisation in national implementation, as well as building capacity in countries with limited resources. The effectiveness of international and EU cybercrime frameworks will ultimately depend on consistent global application and the ability of all signing countries to prevent, criminalise and respond to ransomware attacks in an effective way. Also, adapting legal definitions is a must to anticipate the future technological advancements that will inevitably bring to the landscape of digital extortion.

[24] Art. 23 NIS 2 Directive

[25] ENISA, 'ENISA NIS 360 2024 Report: A Comprehensive Look at Cybersecurity Maturity and Criticality of NIS2 Sectors | ENISA' (*ENISA*, 5 March 2025) <https://www.enisa.europa.eu/news/enisa-nis360-2024-report> accessed 15 April 2025.

[26] ECO (Association of the Internat Industry), 'Rat Running Instead of the Fast Lane? NIS2 Delay Slows Down EU Cybersecurity' (*ECO*, 2024) <https://international.eco.de/news/rat-running-instead-of-the-fast-lane-nis2-delay-slows-down-eu-cybersecurity/> accessed 15 April 2025.

6. Bibliography

6.1. Primary Sources

European Directives

European Parliament and Council Directive 2013/40/EU of 12 August 2013 on attacks against information systems and replacing Council Framework Decision 2005/222/JHA [2013] OJ L. 218

European Parliament and Council Directive 2022/2555 of 14 December 2022 on measures for a high common level of cybersecurity across the Union, amending Regulation (EU) No 910/2014 and Directive (EU) 2018/1972, and repealing Directive (EU) 2016/1148 (NIS 2 Directive) [2022] OJ L. 333/80

Commission Communications

Commission, 'Join Communication to the European Parliament and the Council. Resilience, Deterrence and Defence: Building strong cybersecurity for the EU' JOIN(2017) 450 final

International Treaties

Convention on Cybercrime of the Council of Europe of 23 November 2001, European Treaty Series - No.185. (Budapest Convention)

Cybercrime Convention Committee Documents

Cybercrime Convention Committee (2022) T-CY Guidance Note #12 Aspects of ransomware covered by the Budapest Convention (T-CY(2022)14). Council of Europe. https://rm.coe.int/t-cy-2022-14-guidancenote-ransomware-v4adopted/1680a9355e

6.2. Secondary Sources

Ahmed, M. (Ed.). (2024). Ransomware Evolution (1st ed.). CRC Press. https://doi.org/10.1201/9781003469506

Chimchiuri L, 'The Evolution Of Cybercrime Legislation' (2024) 2 Scientific works of National Aviation University. Series: Law Journal 'Air and Space Law' 221 https://jrnl.nau.edu.ua/index.php/UV/article/view/18813accessed 15 April 2025

de Graaf R and van der Vossen R, 'Bits versus Brains in Content Analysis. Comparing the Advantages and Disadvantages of Manual and Automated Methods for Content Analysis' (2013) 38 433 https://doi.org/10.1515/commun-2013-0025 accessed 15 April 2025

Droppa B, 'NIS2 in Germany: What You Need to Know and Where to Start' (Black Cell, 10 April 2025) https://blackcell.io/nis2-in-germany-what-you-need-to-know-and-where-to-start/ accessed 15 April 2025

ECO (Association of the Internet Industry), 'Rat Running Instead of the Fast Lane? NIS2 Delay Slows Down EU Cybersecurity' (ECO, 2024) https://international.eco.de/news/rat-running-instead-of-the-fast-lane-nis2-delay-slows-down-eu-cybersecurity/ accessed 15 April 2025

ENISA, 'ENISA NIS360 2024 Report: A Comprehensive Look at Cybersecurity Maturity and Criticality of NIS2 Sectors | ENISA' (ENISA, 5 March 2025) https://www.enisa.europa.eu/news/enisa-nis360-2024-report accessed 15 April 2025

FATF (Financial Action Task Force), 'Countering Ransomware Financing' http://www.fatf-gafi.org/publications/Methodsandtrends/countering-ransomware-financing.html accessed 16 April 2025

Joint Communication To The European Parliament And The Council. Resilience, Deterrence and Defence: Building Strong Cybersecurity for the EU - EU Monitor' https://www.eumonitor.eu/9353000/1/j9vvik7m1c3gyxp/vkht9q5wlmzvaccessed 15 April 2025

Jurafsky D and Martin JH, *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition with Language Models* (3 ed, Stanford 2025) https://web.stanford.edu/~jurafsky/slp3/ accessed 16 April 2025

Kirubavathi G, Regis Anne W and Sridevi UK, 'A Recent Review of Ransomware Attacks on Healthcare Industries' (2024) 15 International Journal of System Assurance Engineering and Management 5078 https://doi.org/10.1007/s13198-024-02496-4

Lubin A, 'The Law and Politics of Ransomware' (2022) 55 Vanderbilt Journal of Transnational Law 1177 https://heinonline.org/HOL/P?h=hein.journals/vantl55&i=1223 accessed 15 April 2025

MacColl J and others, 'Ransomware: Victim Insights on Harms to Individuals, Organisations and Society' https://www.rusi.orghttps://www.rusi.org accessed 16 April 2025

Nguyen DrCL and Golman DrW, 'Diffusion of the Budapest Convention on Cybercrime and the Development of Cybercrime Legislation in Pacific Island Countries: "Law on the Books" vs "Law in Action"' (2021) 40 Computer Law & Security Review 105521 https://www.sciencedirect.com/science/article/pii/S0267364920301266

PricewaterhouseCoopers, 'European NIS2 Directive' (PwC) https://www.pwc.de/en/cyber-security/european-nis2-directive-implications-for-businesses-and-institutions.html accessed 15 April 2025

Runte C and Biendl M, 'Data Protection and Cybersecurity Laws in Germany' (2021) https://cms.law/en/int/expert-guides/cms-expert-guide-to-data-protection-and-cyber-security-laws/germany accessed 15 April 2025

Shea S and Irei A, 'What Is Ransomware? How It Works and How to Remove It' (Search Security) https://www.techtarget.com/searchsecurity/definition/ransomware accessed 15 April 2025

Zimba A and Chishimba M, 'Understanding the Evolution of Ransomware: Paradigm Shifts in Attack Structures' (2019) 11 International Journal of Computer Network and Information Security 26

7. Appendix

The Python code of the paper is organised into three main parts:

- Part 1 defines the RansomwareTechEvolutionAnalyzer class, initialising the framework for analysing legal instruments and their technological and implementation categories;
- Part 2 contains the analysis methods, including temporal evolution and cross-document comparison, to quantify how frameworks adapt to ransomware;
- Part 3 focuses on report generation and visualisation, extracting text from PDFs and producing visual outputs like heatmaps.

The matrix chart (heatmap) was selected over other visualisation types, such as line graphs or bar charts, because it effectively displays the comparative intensity of references across multiple technological and implementation categories simultaneously, offering a clear overview in a single, visually intuitive graphic.

PART 1: Python Tech Evolution Analyser

```
class RansomwareTechEvolutionAnalyzer:
    """
    A framework for analyzing how international and EU cybercrime instruments
    adapt to the technological evolution of ransomware and the role of
    national implementation.
    """

    def __init__(self):
        """Initialize the analyzer with the legal instruments and their metadata."""
        self.documents = {
            'budapest': {
                'title': 'Convention on Cybercrime (Budapest Convention)',
                'year': 2001,
                'body': 'Council of Europe',
                'type': 'International Convention',
                'text': None
            },
            'directive_attacks': {
                'title': 'Directive on Attacks Against Information Systems (2013/40/EU)',
                'year': 2013,
                'body': 'European Union',
                'type': 'EU Directive',
                'text': None
            },
            'ransomware_guidance': {
                'title': 'T-CY Guidance Note #12: Aspects of Ransomware covered by the Budapest
Convention',
                'year': 2022,
                'body': 'Cybercrime Convention Committee (T-CY)',
                'type': 'Guidance Note',
                'text': None
            },
            'nis2': {
                'title': 'Directive on measures for a high common level of cybersecurity across the Union
(NIS 2 Directive)',
                'year': 2022,
                'body': 'European Union',
                'type': 'EU Directive',
                'text': None
            }
        }

        # Technological evolution categories for analysis
        self.tech_evolution_categories = {
            'encryption_technology': [
                'encryption', 'decrypt', 'cryptography', 'key', 'cipher', 'cryptographic',
                'encrypted', 'decryption', 'asymmetric', 'symmetric', 'algorithm', 'aes', 'rsa'
            ],

            'payment_methods': [
                'cryptocurrency', 'bitcoin', 'crypto', 'wallet', 'blockchain', 'payment',
                'transaction', 'monero', 'coin', 'digital currency', 'virtual currency', 'token'
            ],

            'distribution_methods': [
                'phishing', 'exploit', 'vulnerability', 'botnet', 'spam', 'email',
                'dropper', 'malspam', 'drive-by', 'download', 'supply chain', 'backdoor'
            ],

            'advanced_techniques': [
                'obfuscation', 'polymorphic', 'fileless', 'stealth', 'persistence',
                'lateral movement', 'privilege escalation', 'zero-day', 'evasion', 'rootkit'
            ],

            'targeting_methods': [
                'reconnaissance', 'targeting', 'scanning', 'fingerprinting', 'profiling',
                'selection', 'spear', 'tailored', 'specific', 'high-value'
            ],

            'raas_ecosystem': [
                'service', 'affiliate', 'model', 'business model', 'marketplace', 'platform',
                'profit sharing', 'underground', 'dark web', 'forum', 'rental', 'subscription'
            ],

            'ai_ml_references': [
                'artificial intelligence', 'ai', 'machine learning', 'ml', 'automated', 'automation',
                'algorithm', 'neural', 'deep learning', 'predictive', 'intelligent'
            ]
        }

        # National implementation categories
        self.implementation_categories = {
            'harmonization': [
                'harmonize', 'harmonization', 'harmonisation', 'uniform', 'consistency', 'consistent',
                'align', 'alignment', 'converge', 'convergence', 'standard', 'standardize'
            ],

            'transposition': [
                'transpose', 'transposition', 'implement', 'implementation', 'adopt', 'adoption',
                'enact', 'enactment', 'incorporate', 'incorporation', 'national law', 'domestic law'
            ],

            'enforcement_mechanisms': [
                'enforce', 'enforcement', 'prosecute', 'prosecution', 'penalty', 'penalties',
                'sanction', 'sanctions', 'fine', 'imprisonment', 'punishment', 'punish'
            ],

            'cooperation_mechanisms': [
                'cooperate', 'cooperation', 'collaborate', 'collaboration', 'assist', 'assistance',
                'coordinate', 'coordination', 'share', 'sharing', 'exchange', 'joint'
            ],

            'reporting_requirements': [
                'report', 'reporting', 'notify', 'notification', 'disclose', 'disclosure',
                'inform', 'information', 'alert', 'warn', 'warning', 'communicate'
            ],

            'national_authorities': [
                'authority', 'authorities', 'agency', 'agencies', 'regulator', 'regulatory',
                'competent', 'national', 'csirt', 'cert', 'point of contact', 'single point'
            ],

            'judicial_mechanisms': [
                'judicial', 'court', 'judge', 'legal', 'prosecution', 'prosecutor',
                'evidence', 'procedural', 'warrant', 'order', 'jurisdiction', 'extradition'
            ],

            'compliance_monitoring': [
                'compliance', 'monitor', 'monitoring', 'audit', 'review', 'assessment',
                'evaluate', 'evaluation', 'verify', 'verification', 'check', 'inspection'
            ]
        }

        # Results
        self.tech_evolution_results = {}
        self.implementation_results = {}
        self.articles_tech_references = {}
        self.articles_implementation_references = {}
        self.temporal_evolution = {}
        self.cross_document_comparison = {}

    def load_document(self, doc_id, text):
        """Load document text into the analyzer."""
        if doc_id in self.documents:
            self.documents[doc_id]['text'] = text
            print(f"Loaded {self.documents[doc_id]['title']} ({self.documents[doc_id]['year']})")
            return True
        else:
            print(f"Unknown document ID: {doc_id}")
            return False

    def extract_articles(self, doc_id):
        """Extract articles from a document."""
        if doc_id not in self.documents or not self.documents[doc_id]['text']:
            print(f"Document {doc_id} not loaded")
            return {}

        text = self.documents[doc_id]['text']
        pattern = self.article_patterns.get(doc_id)

        if not pattern:
            return {}

        articles = {}
        matches = re.finditer(pattern, text)

        for match in matches:
            if len(match.groups()) >= 2:
                article_num = match.group(1)
                article_title = match.group(2).strip()

                # Find the article content
                start_pos = match.end()

                # Find the next article or the end of text
                next_match = re.search(r'Article\s+\d+', text[start_pos:])
                if next_match:
                    end_pos = start_pos + next_match.start()
                else:
                    end_pos = len(text)

                article_content = text[start_pos:end_pos].strip()
                articles[article_num] = {
                    'title': article_title,
                    'content': article_content
                }

        print(f"Extracted {len(articles)} articles from {self.documents[doc_id]['title']}")
        return articles

        # Analyze presence of tech evolution categories
        for category, terms in self.tech_evolution_categories.items():
            category_count = 0
            category_matches = []

            for term in terms:
                pattern = r'\b' + re.escape(term) + r'[a-z]*\b'
                matches = re.finditer(pattern, text)

                for match in matches:
                    category_count += 1
                    category_matches.append(match.group(0))

            results[category] = {
                'count': category_count,
                'matches': Counter(category_matches).most_common()
            }

        self.tech_evolution_results[doc_id] = results
        return results

    def analyze_implementation(self, doc_id):
        """Analyze how a document addresses national implementation mechanisms."""
        if doc_id not in self.documents or not self.documents[doc_id]['text']:
            print(f"Document {doc_id} not loaded")
            return {}

        text = self.documents[doc_id]['text'].lower()
        results = {}

        # Analyze presence of implementation categories
        for category, terms in self.implementation_categories.items():
            category_count = 0
            category_matches = []

            for term in terms:
                pattern = r'\b' + re.escape(term) + r'[a-z]*\b'
                matches = re.finditer(pattern, text)

                for match in matches:
                    category_count += 1
                    category_matches.append(match.group(0))

            results[category] = {
                'count': category_count,
                'matches': Counter(category_matches).most_common()
            }

        self.implementation_results[doc_id] = results
        return results

            # Check for tech evolution terms
            tech_mentions = []

            for category, terms in self.tech_evolution_categories.items():
                for term in terms:
                    pattern = r'\b' + re.escape(term) + r'[a-z]*\b'
                    matches = re.finditer(pattern, content)
                    for match in matches:
                        tech_mentions.append({
                            'category': category,
                            'term': match.group(0),
                            'context': self._get_context(content, match.start(), 50)
                        })


            # Check for implementation terms
            implementation_mentions = []

            for category, terms in self.implementation_categories.items():
                for term in terms:
                    pattern = r'\b' + re.escape(term) + r'[a-z]*\b'
                    matches = re.finditer(pattern, content)

                    for match in matches:
                        implementation_mentions.append({
                            'category': category,
                            'term': match.group(0),
                            'context': self._get_context(content, match.start(), 50)
                        })

                    # Also check the title
                    matches = re.finditer(pattern, title)
                    for match in matches:
                        implementation_mentions.append({
                            'category': category,
                            'term': match.group(0),
                            'context': f"TITLE: {title}"
                        })

            if implementation_mentions:
                implementation_articles[article_num] = {
                    'title': article_data['title'],
                    'mentions': implementation_mentions
                }

        self.articles_implementation_references[doc_id] = implementation_articles
        return implementation_articles

    def _get_context(self, text, position, context_size):
        """Helper method to get context around a match position."""
        start = max(0, position - context_size)
        end = min(len(text), position + context_size)

        context = text[start:end]

        # Add ellipsis if context is truncated
        if start > 0:
            context = "..." + context
        if end < len(text):
            context = context + "..."

        return context
```

PART 2: Ransomware Technological Evolution Analysis - (Analysis Methods)

```
# This script contains analysis methods and execution code

# Add methods to the RansomwareTechEvolutionAnalyzer class
def analyze_temporal_evolution(self):
    """Analyze how technological adaptation and national implementation approaches have evolved
over time."""
    if not self.tech_evolution_results or not self.implementation_results:
        print("Run analysis methods for all documents first")
        return {}

    evolution = {
        'tech_evolution': {
            'by_category': {},
            'by_document': {}
        },
        'implementation': {
            'by_category': {},
            'by_document': {}
        }
    }

    # Sort documents by year
    sorted_docs = sorted(self.documents.items(), key=lambda x: x[1]['year'])

    # Analyze technology evolution by category
    for category in self.tech_evolution_categories:
        category_evolution = []

        for doc_id, doc_info in sorted_docs:
            if doc_id in self.tech_evolution_results:
                doc_result = self.tech_evolution_results[doc_id]

                if category in doc_result:
                    category_evolution.append({
                        'document': doc_id,
                        'year': doc_info['year'],
                        'count': doc_result[category]['count'],
                        'top_terms': doc_result[category]['matches'][:5] if doc_result[category]['matches'] else
[]
                    })

        evolution['tech_evolution']['by_category'][category] = category_evolution
    # Analyze implementation by category
    for category in self.implementation_categories:
        category_evolution = []

        for doc_id, doc_info in sorted_docs:
            if doc_id in self.implementation_results:
                doc_result = self.implementation_results[doc_id]

                if category in doc_result:
                    category_evolution.append({
                        'document': doc_id,
                        'year': doc_info['year'],
                        'count': doc_result[category]['count'],
                        'top_terms': doc_result[category]['matches'][:5] if doc_result[category]['matches'] else
[]
                    })

        evolution['implementation']['by_category'][category] = category_evolution

    # Analyze by document and year
    for doc_id, doc_info in sorted_docs:
        # Tech evolution by document
        if doc_id in self.tech_evolution_results:
            doc_result = self.tech_evolution_results[doc_id]
            doc_evolution = {}

            for category in self.tech_evolution_categories:
                if category in doc_result:
                    doc_evolution[category] = {
                        'count': doc_result[category]['count'],
                        'top_terms': doc_result[category]['matches'][:5] if doc_result[category]['matches'] else
[]
                    }

            evolution['tech_evolution']['by_document'][doc_id] = {
                'year': doc_info['year'],
                'title': doc_info['title'],
                'categories': doc_evolution
            }

        # Implementation by document
        if doc_id in self.implementation_results:
            doc_result = self.implementation_results[doc_id]
            doc_evolution = {}

            for category in self.implementation_categories:
                if category in doc_result:
                    doc_evolution[category] = {
                        'count': doc_result[category]['count'],
                        'top_terms': doc_result[category]['matches'][:5] if doc_result[category]['matches'] else
[]
                    }
            evolution['implementation']['by_document'][doc_id] = {
                'year': doc_info['year'],
                'title': doc_info['title'],
                'categories': doc_evolution
            }

    self.temporal_evolution = evolution
    return evolution

def compare_documents(self):
    """Compare technological adaptation and national implementation approaches across
documents."""
    if not self.tech_evolution_results or not self.implementation_results:
        print("Run analysis methods for all documents first")
        return {}

    comparison = {
        'tech_evolution_matrix': {},
        'implementation_matrix': {},
        'term_overlap': {'tech_evolution': {}, 'implementation': {}},
        'unique_approaches': {'tech_evolution': {}, 'implementation': {}}
    }

    # Create comparison matrix for tech evolution categories
    tech_categories = list(self.tech_evolution_categories.keys())
    documents = list(self.documents.keys())

    tech_matrix = pd.DataFrame(0, index=documents, columns=tech_categories)

    for doc_id in documents:
        if doc_id in self.tech_evolution_results:
            for category in tech_categories:
                if category in self.tech_evolution_results[doc_id]:
                    value = self.tech_evolution_results[doc_id][category].get('count', 0)
                    try:
                        tech_matrix.loc[doc_id, category] = float(value)
                    except (ValueError, TypeError):
                        tech_matrix.loc[doc_id, category] = 0  # Default to 0 if conversion fails

    # Debug: Inspect the matrix
    print("Tech Matrix before conversion:")
    print(tech_matrix)
    print("Tech Matrix dtypes before conversion:")
    print(tech_matrix.dtypes)

    tech_matrix = tech_matrix.astype(float)

    print("Tech Matrix after conversion:")
    print(tech_matrix)
    print("Tech Matrix dtypes after conversion:")
    print(tech_matrix.dtypes)
```

```
        comparison['tech_evolution_matrix'] = tech_matrix

        # Create comparison matrix for implementation categories
        impl_categories = list(self.implementation_categories.keys())
        impl_matrix = pd.DataFrame(0, index=documents, columns=impl_categories)

        for doc_id in documents:
            if doc_id in self.implementation_results:
                for category in impl_categories:
                    if category in self.implementation_results[doc_id]:
                        value = self.implementation_results[doc_id][category].get('count', 0)
                        try:
                            impl_matrix.loc[doc_id, category] = float(value)
                        except (ValueError, TypeError):
                            impl_matrix.loc[doc_id, category] = 0  # Default to 0 if conversion fails

        # Debug: Inspect the matrix
        print("Implementation Matrix before conversion:")
        print(impl_matrix)
        print("Implementation Matrix dtypes before conversion:")
        print(impl_matrix.dtypes)

        impl_matrix = impl_matrix.astype(float)

        print("Implementation Matrix after conversion:")
        print(impl_matrix)
        print("Implementation Matrix dtypes after conversion:")
        print(impl_matrix.dtypes)

        comparison['implementation_matrix'] = impl_matrix

        # ... rest of the method (term_overlap and unique_approaches sections) ...

        self.cross_document_comparison = comparison
        return comparison

    def visualize_tech_evolution(self):
        """Visualize the distribution and evolution of technological adaptation across documents."""
        if not self.cross_document_comparison or 'tech_evolution_matrix' not in
    self.cross_document_comparison:
            print("Run compare_documents() first")
            return

        tech_matrix = self.cross_document_comparison['tech_evolution_matrix']

        print("Tech Matrix:")
        print(tech_matrix)
        print("Data types:")
        print(tech_matrix.dtypes)
```

```
# Convert to numeric type
tech_matrix = tech_matrix.apply(pd.to_numeric, errors='coerce')

print("Tech Matrix after conversion:")
print(tech_matrix)
print("Data types after conversion:")
print(tech_matrix.dtypes)

if tech_matrix.isnull().all().all():
   print("Error: Tech matrix contains no valid numeric data")
   return

# Inside visualize_tech_evolution method (around line 200-220)
plt.figure(figsize=(16, 10))  # Adjusted to match second graph's dimensions

# Create the heatmap with improved styling
ax = sns.heatmap(tech_matrix,
            annot=True,            # Show values in cells
            fmt='g',               # Format as integers
            cmap='Blues',          # Use Blues colormap like second graph
            linewidths=.5,         # Add thin lines between cells
            cbar_kws={'label': 'Reference Count'})  # Label the colorbar

# Enhance title and labels with better fonts and positioning
ax.set_title('References to Ransomware Technological Evolution Across Documents',
         fontsize=14, pad=20)
ax.set_xlabel('Technology Category', fontsize=14)
ax.set_ylabel('Document', fontsize=14)

# Adjust tick labels to match second graph
plt.xticks(rotation=45, ha='right', fontsize=12)
plt.yticks(rotation=0, fontsize=12)

# Make the plot look tighter
plt.tight_layout()
plt.show()

# Create a grouped bar chart
tech_matrix.plot(kind='bar', figsize=(16, 8))
plt.title('References to Ransomware Technological Evolution Across Documents')
plt.ylabel('Frequency')
plt.xlabel('Document')
plt.legend(title='Technology Category', bbox_to_anchor=(1.05, 1), loc='upper left')
plt.tight_layout()
plt.show()

# Create a normalized stacked bar chart
tech_matrix_norm = tech_matrix.div(tech_matrix.sum(axis=1), axis=0) * 100
tech_matrix_norm.plot(kind='bar', stacked=True, figsize=(16, 8), colormap='viridis')
plt.title('Relative Focus on Ransomware Technology Categories Across Documents')
plt.ylabel('Percentage (%)')
plt.xlabel('Document')
```

```
  plt.legend(title='Technology Category', bbox_to_anchor=(1.05, 1), loc='upper left')
  plt.tight_layout()
  plt.show()

def visualize_implementation(self):
  """Visualize the distribution and evolution of national implementation approaches across
documents."""
  if not self.cross_document_comparison or 'implementation_matrix' not in
self.cross_document_comparison:
    print("Run compare_documents() first")
    return

  impl_matrix = self.cross_document_comparison['implementation_matrix']

  print("Implementation Matrix in visualize_implementation:")
  print(impl_matrix)
  print("Implementation Matrix dtypes:")
  print(impl_matrix.dtypes)

  # Ensure numeric type
  impl_matrix = impl_matrix.apply(pd.to_numeric, errors='coerce')

  print("Implementation Matrix after pd.to_numeric:")
  print(impl_matrix)
  print("Implementation Matrix dtypes after pd.to_numeric:")
  print(impl_matrix.dtypes)

  if impl_matrix.isnull().all().all():
    print("Error: Implementation matrix contains no valid numeric data")
    return

  # Create a heatmap
  plt.figure(figsize=(16, 10))
  ax = sns.heatmap(impl_matrix, annot=True, fmt='g', cmap='Blues', linewidths=0.5,
linecolor='white', cbar_kws={"shrink": 0.8, "label": "Reference Count"})
  ax.set_title('References to National Implementation Approaches Across Documents', fontsize=16,
pad=20)
  ax.set_xlabel('Implementation Category', fontsize=14)
  ax.set_ylabel('Document', fontsize=14)
  plt.xticks(rotation=45, ha='right', fontsize=12)
  plt.yticks(rotation=0, fontsize=12)
  plt.tight_layout()
  plt.show()

  # Create a grouped bar chart
  impl_matrix.plot(kind='bar', figsize=(16, 8))
  plt.title('References to National Implementation Approaches Across Documents')
  plt.ylabel('Frequency')
  plt.xlabel('Document')
  plt.legend(title='Implementation Category', bbox_to_anchor=(1.05, 1), loc='upper left')
  plt.tight_layout()
  plt.show()
```

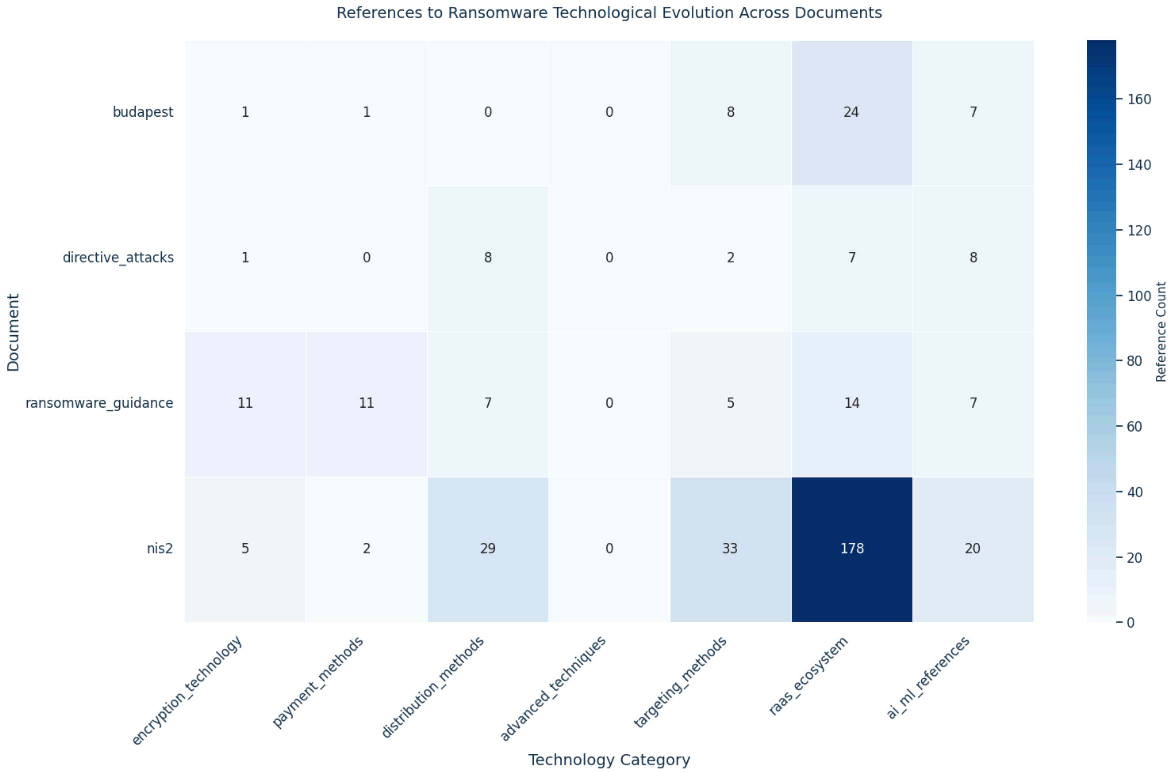
```
def analyze_adaptation_implementation_relationship(self):
    """Analyze the relationship between technological adaptation and national implementation
approaches."""
    if not self.cross_document_comparison:
        print("Run compare_documents() first")
        return

    tech_matrix = self.cross_document_comparison['tech_evolution_matrix']
    impl_matrix = self.cross_document_comparison['implementation_matrix']

    # Calculate correlation between tech evolution and implementation categories
    correlation = pd.DataFrame(index=tech_matrix.columns, columns=impl_matrix.columns)

    for tech_cat in tech_matrix.columns:
        for impl_cat in impl_matrix.columns:
            tech_values = pd.to_numeric(tech_matrix[tech_cat], errors='coerce')
            impl_values = pd.to_numeric(impl_matrix[impl_cat], errors='coerce')
            if len(tech_values.dropna()) > 1 and len(impl_values.dropna()) > 1:
                corr = tech_values.corr(impl_values)
                correlation.loc[tech_cat, impl_cat] = corr if not pd.isna(corr) else 0

    # Debug: Inspect correlation
    print("Correlation Matrix:")
    print(correlation)
    print("Correlation dtypes:")
    print(correlation.dtypes)

    correlation = correlation.astype(float)

    print("Correlation Matrix after conversion:")
    print(correlation)
    print("Correlation dtypes after conversion:")
    print(correlation.dtypes)

    # Visualize the correlation matrix
    plt.figure(figsize=(12, 10))
    sns.heatmap(correlation, annot=True, cmap='Blues', center=0, fmt='.2f')
    plt.title('Correlation Between Technological Adaptation and National Implementation Approaches')
    plt.ylabel('Technology Category')
    plt.xlabel('Implementation Category')
    plt.tight_layout()
    plt.show()

    # Calculate and visualize overall focus
    tech_totals = tech_matrix.sum(axis=1)
    impl_totals = impl_matrix.sum(axis=1)

    focus_ratio = pd.DataFrame({
        'Technology Focus': tech_totals,
        'Implementation Focus': impl_totals
    })
    plt.figure(figsize=(10, 6))
    focus_ratio.plot(kind='bar')
    plt.title('Balance Between Technological Adaptation and National Implementation Approaches')
    plt.ylabel('Number of References')
    plt.xlabel('Document')
    plt.grid(True, axis='y', linestyle='--', alpha=0.7)
    plt.tight_layout()
    plt.show()

    return correlation, focus_ratio

# Assign the new methods to the RansomwareTechEvolutionAnalyzer class
RansomwareTechEvolutionAnalyzer.analyze_temporal_evolution = analyze_temporal_evolution
RansomwareTechEvolutionAnalyzer.compare_documents = compare_documents
RansomwareTechEvolutionAnalyzer.visualize_tech_evolution = visualize_tech_evolution
RansomwareTechEvolutionAnalyzer.visualize_implementation = visualize_implementation
RansomwareTechEvolutionAnalyzer.analyze_adaptation_implementation_relationship =
analyze_adaptation_implementation_relationship

# Define execution function
def run_adaptation_implementation_analysis(budapest_text, attacks_directive_text,
ransomware_guidance_text, nis2_text):
    """Run a complete analysis on how the frameworks adapt to ransomware's technological evolution
and national implementation."""

    # Initialize the analyzer
    analyzer = RansomwareTechEvolutionAnalyzer()

    # Load documents
    analyzer.load_document('budapest', budapest_text)
    analyzer.load_document('directive_attacks', attacks_directive_text)
    analyzer.load_document('ransomware_guidance', ransomware_guidance_text)
    analyzer.load_document('nis2', nis2_text)

    # Extract articles and recitals
    results = {}

    for doc_id in analyzer.documents:
        # Extract articles
        articles = analyzer.extract_articles(doc_id)
        results[f'{doc_id}_articles'] = articles

        # Analyze technological evolution
        tech_evolution = analyzer.analyze_tech_evolution(doc_id)
        results[f'{doc_id}_tech_evolution'] = tech_evolution

        # Analyze implementation mechanisms
        implementation = analyzer.analyze_implementation(doc_id)
        results[f'{doc_id}_implementation'] = implementation

        # Find tech evolution references in articles
        tech_articles = analyzer.find_tech_evolution_in_articles(doc_id, articles)
        results[f'{doc_id}_tech_articles'] = tech_articles

        # Find implementation references in articles
        implementation_articles = analyzer.find_implementation_in_articles(doc_id, articles)
        results[f'{doc_id}_implementation_articles'] = implementation_articles

    # Analyze temporal evolution
    evolution = analyzer.analyze_temporal_evolution()
    results['temporal_evolution'] = evolution

    # Compare documents
    comparison = analyzer.compare_documents()
    results['cross_document_comparison'] = comparison

    # Return the analyzer for further use
    return analyzer, results

# Main execution function for Google Colab
def main():
    # Set up necessary imports and utilities
    !pip install PyPDF2

    from google.colab import files
    import io
    import PyPDF2

    def extract_text_from_pdf(file_path):
        """Extract text from a PDF file."""
        with open(file_path, 'rb') as file:
            pdf_reader = PyPDF2.PdfReader(file)
            text = ""
            for page_num in range(len(pdf_reader.pages)):
                text += pdf_reader.pages[page_num].extract_text()
        return text

    print("1. CETS_185.pdf (Budapest Convention)")
    print("2. DIRECTIVE 2013 40 EU.pdf (Directive on Attacks Against Information Systems)")
    print("3. T-CY Ransomware guidance note.pdf (T-CY Guidance Note #12)")
    print("4. DIRECTIVE (EU) 2022 2555.pdf (NIS 2 Directive)")

    uploaded = files.upload()

    # Extract text from uploaded documents
    file_names = list(uploaded.keys())
    print("\nUploaded files:", file_names)

    # Extract text from each file
    budapest_text = ""
    attacks_directive_text = ""
    ransomware_guidance_text = ""
    nis2_text = ""
    if budapest_file:
        budapest_text = extract_text_from_pdf(budapest_file)
        print(f"Budapest Convention: {len(budapest_text)} characters extracted")
    else:
        print("Budapest Convention file not found. Please check uploaded files.")

    if attacks_file:
        attacks_directive_text = extract_text_from_pdf(attacks_file)
        print(f"Directive on Attacks: {len(attacks_directive_text)} characters extracted")
    else:
        print("Directive on Attacks file not found. Please check uploaded files.")

    if guidance_file:
        ransomware_guidance_text = extract_text_from_pdf(guidance_file)
        print(f"Ransomware Guidance: {len(ransomware_guidance_text)} characters extracted")
    else:
        print("Ransomware Guidance file not found. Please check uploaded files.")

    if nis2_file:
        nis2_text = extract_text_from_pdf(nis2_file)
        print(f"NIS 2 Directive: {len(nis2_text)} characters extracted")
    else:
        print("NIS 2 Directive file not found. Please check uploaded files.")

    # Run the analysis
    print("\nRunning analysis on how frameworks adapt to ransomware's technological evolution and national implementation...")
    analyzer, results = run_adaptation_implementation_analysis(budapest_text, attacks_directive_text, ransomware_guidance_text, nis2_text)

    # Visualize the results
    print("\nGenerating visualizations of technological adaptation and implementation approaches...")

    # Visualize technological evolution
    analyzer.visualize_tech_evolution()

    # Visualize implementation approaches
    analyzer.visualize_implementation()

    # Analyze relationship between adaptation and implementation
    analyzer.analyze_adaptation_implementation_relationship()

    # Generate detailed tables for technological evolution
    print("\n\nDetailed Analysis of Technological Evolution Categories\n")

    # Create DataFrame for tech evolution categories
    tech_categories = list(analyzer.tech_evolution_categories.keys())
    tech_data = {}

    for doc_id, doc_info in analyzer.documents.items():
        if doc_id in analyzer.tech_evolution_results:
            doc_data = {}
            for category in tech_categories:
                if category in analyzer.tech_evolution_results[doc_id]:
                    doc_data[category] = analyzer.tech_evolution_results[doc_id][category]['count']
            tech_data[doc_id] = doc_data

    df_tech = pd.DataFrame(tech_data).T
    df_tech.index = [f"{analyzer.documents[doc_id]['title']} ({analyzer.documents[doc_id]['year']})" for
doc_id in df_tech.index]

    print(df_tech)

    # Generate detailed tables for implementation approaches
    print("\n\nDetailed Analysis of National Implementation Categories\n")

    # Create DataFrame for implementation categories
    impl_categories = list(analyzer.implementation_categories.keys())
    impl_data = {}

    for doc_id, doc_info in analyzer.documents.items():
        if doc_id in analyzer.implementation_results:
            doc_data = {}
            for category in impl_categories:
                if category in analyzer.implementation_results[doc_id]:
                    doc_data[category] = analyzer.implementation_results[doc_id][category]['count']
            impl_data[doc_id] = doc_data

    df_impl = pd.DataFrame(impl_data).T
    df_impl.index = [f"{analyzer.documents[doc_id]['title']} ({analyzer.documents[doc_id]['year']})" for
doc_id in df_impl.index]

    print(df_impl)

    return analyzer, results

# If this script is run directly, execute the main function
if __name__ == "__main__":
    analyzer, results = main()
```

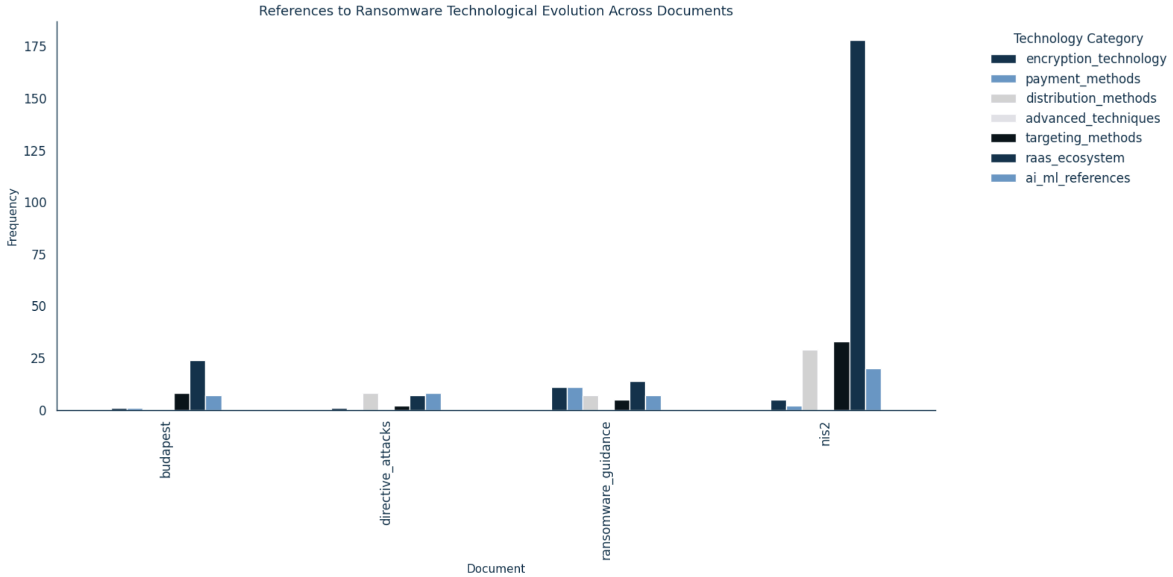
```
Requirement already satisfied: PyPDF2 in /usr/local/lib/python3.11/dist-packages (3.0.1)
1. CETS_185.pdf (Budapest Convention)
2. DIRECTIVE 2013 40 EU.pdf (Directive on Attacks Against Information Systems)
3. T-CY Ransomware guidance note.pdf (T-CY Guidance Note #12)
4. DIRECTIVE (EU) 2022 2555.pdf (NIS 2 Directive)

Saving T-CY Ransomware guidance note.pdf to T-CY Ransomware guidance note (10).pdf
Saving DIRECTIVE (EU) 2022 2555.pdf to DIRECTIVE (EU) 2022 2555 (10).pdf
Saving DIRECTIVE 2013 40 EU.pdf to DIRECTIVE 2013 40 EU (10).pdf
Saving CETS_185.pdf to CETS_185 (10).pdf

Uploaded files: ['T-CY Ransomware guidance note (10).pdf', 'DIRECTIVE (EU) 2022 2555 (10).pdf',
'DIRECTIVE 2013 40 EU (10).pdf', 'CETS_185 (10).pdf']
```

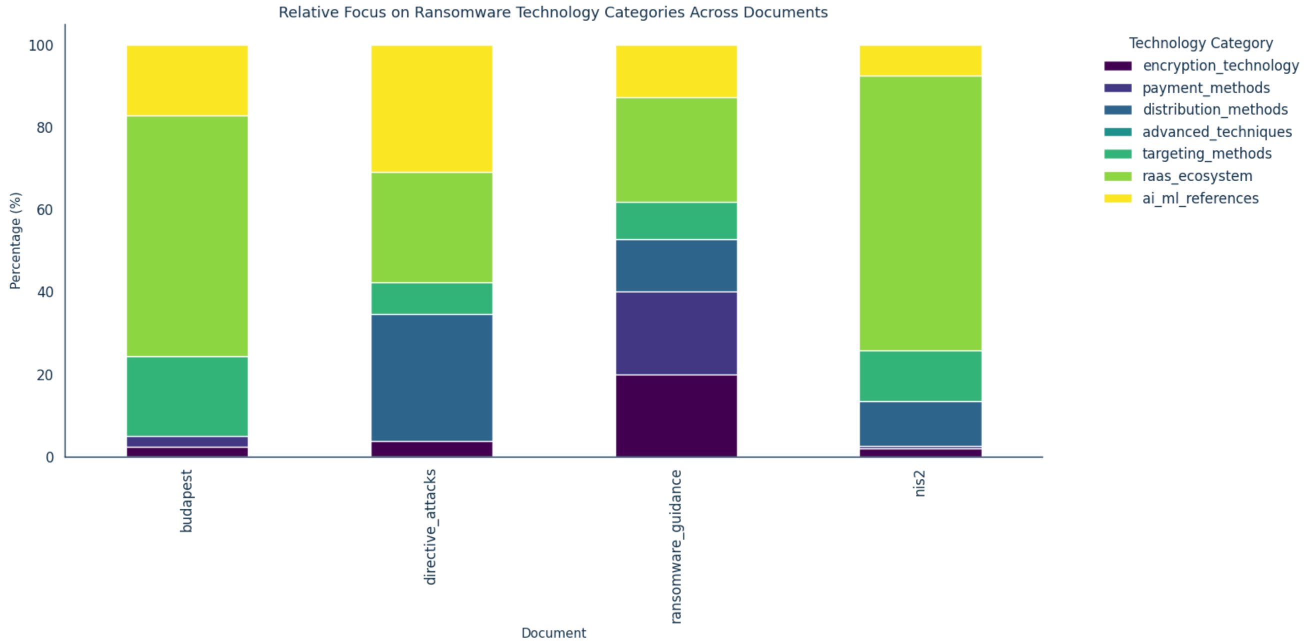
```
Budapest Convention: 66824 characters extracted
Directive on Attacks: 36435 characters extracted
Ransomware Guidance: 21759 characters extracted
NIS 2 Directive: 149627 characters extracted
```

PART 3: Running analysis and visualisations of data

```
Running analysis on how frameworks adapt to ransomware's technological evolution and national
implementation...
Loaded Convention on Cybercrime (Budapest Convention) (2001)
Loaded Directive on Attacks Against Information Systems (2013/40/EU) (2013)
Loaded T-CY Guidance Note #12: Aspects of Ransomware covered by the Budapest Convention
(2022)
Loaded Directive on measures for a high common level of cybersecurity across the Union (NIS 2
Directive) (2022)
Tech Matrix before conversion:

Generating visualizations of technological adaptation and implementation approaches...
Tech Matrix:
                    encryption_technology  payment_methods  \
budapest                              1.0              1.0
directive_attacks                     1.0              0.0
ransomware_guidance                  11.0             11.0
nis2                                  5.0              2.0

                    distribution_methods  advanced_techniques  \
budapest                             0.0                  0.0
directive_attacks                    8.0                  0.0
ransomware_guidance                  7.0                  0.0
nis2                                29.0                  0.0

                    targeting_methods  raas_ecosystem  ai_ml_references
budapest                          8.0            24.0               7.0
directive_attacks                 2.0             7.0               8.0
ransomware_guidance               5.0            14.0               7.0
nis2                             33.0           178.0              20.0
Data types:
dtype: object
```

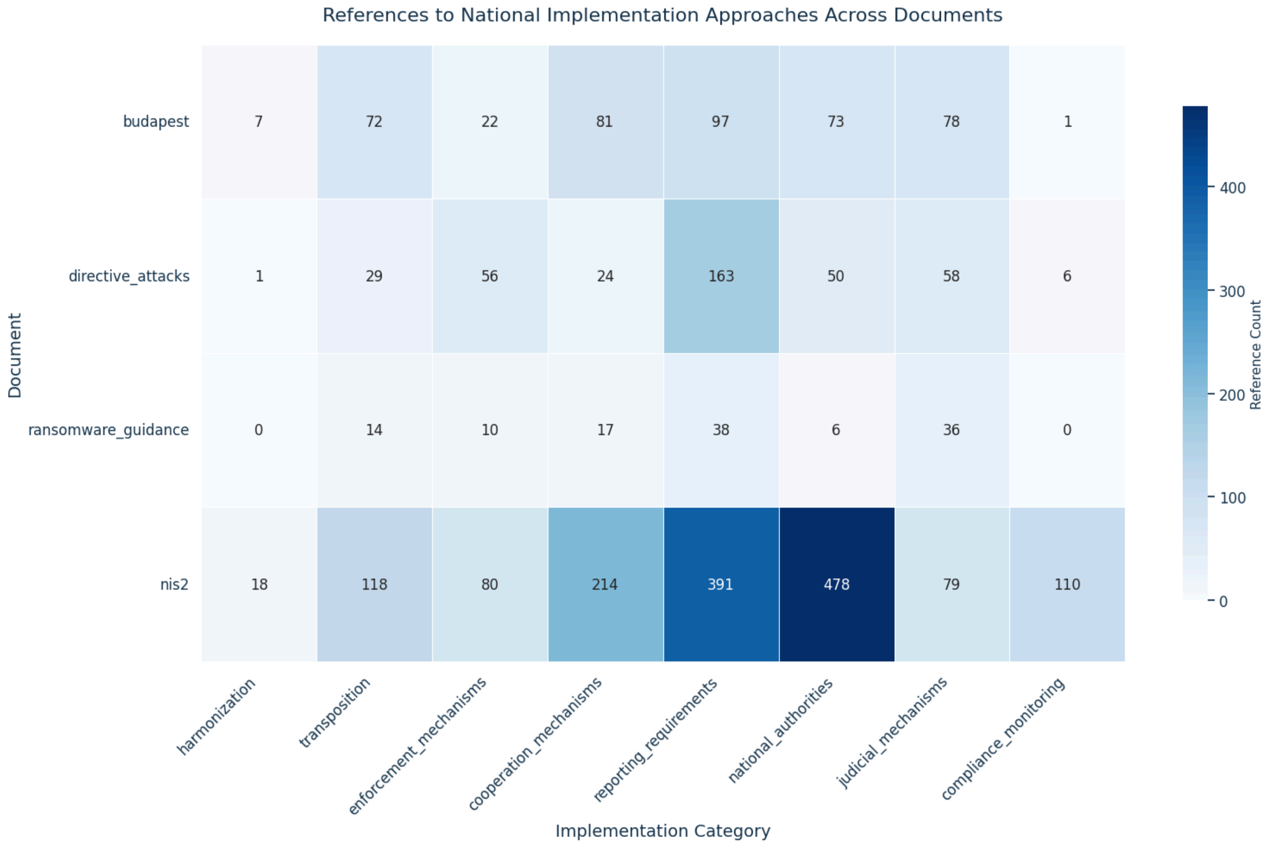

References to Ransomware Technological Evolution Across Documents
Document
Technology Category
Reference Count
encryption_technology
payment_methods
distribution_methods
advanced_techniques
targeting_methods
raas_ecosystem
ai_ml_references
budapest
1
1
0
0
8
24
7
directive_attacks
1
0
8
0
2
7
8
ransomware_guidance
11
11
7
0
5
14
7
nis2
5
2
29
0
33
178
20
0
20
40
60
80
100
120
140
160

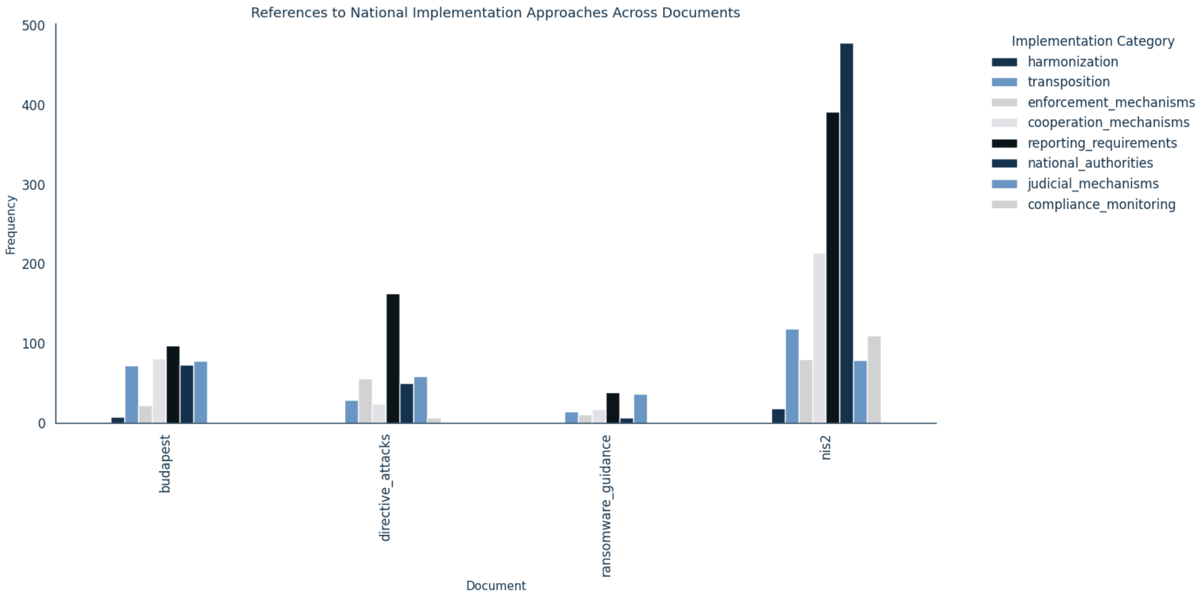

References to Ransomware Technological Evolution Across Documents
Frequency
0
25
50
75
100
125
150
175
budapest
directive_attacks
ransomware_guidance
nis2
Document
Technology Category
encryption_technology
payment_methods
distribution_methods
advanced_techniques
targeting_methods
raas_ecosystem
ai_ml_references



```
Implementation Matrix in visualize_implementation:
Implementation Matrix after pd.to_numeric:
                     harmonization  transposition  enforcement_mechanisms  \
budapest                       7.0           72.0                    22.0
directive_attacks              1.0           29.0                    56.0
ransomware_guidance            0.0           14.0                    10.0
nis2                          18.0          118.0                    80.0

                     cooperation_mechanisms  reporting_requirements  \
budapest                               81.0                    97.0
directive_attacks                      24.0                   163.0
ransomware_guidance                    17.0                    38.0
nis2                                  214.0                   391.0

                     national_authorities  judicial_mechanisms  \
budapest                             73.0                 78.0
directive_attacks                    50.0                 58.0
ransomware_guidance                   6.0                 36.0
nis2                                478.0                 79.0

                     compliance_monitoring
budapest                               1.0
directive_attacks                      6.0
ransomware_guidance                    0.0
nis2                                 110.0
Implementation Matrix dtypes after pd.to_numeric:
harmonization             float64
transposition             float64
enforcement_mechanisms    float64
cooperation_mechanisms    float64
reporting_requirements    float64
```

```
national_authorities      float64
judicial_mechanisms       float64
compliance_monitoring     float64
dtype: object
```





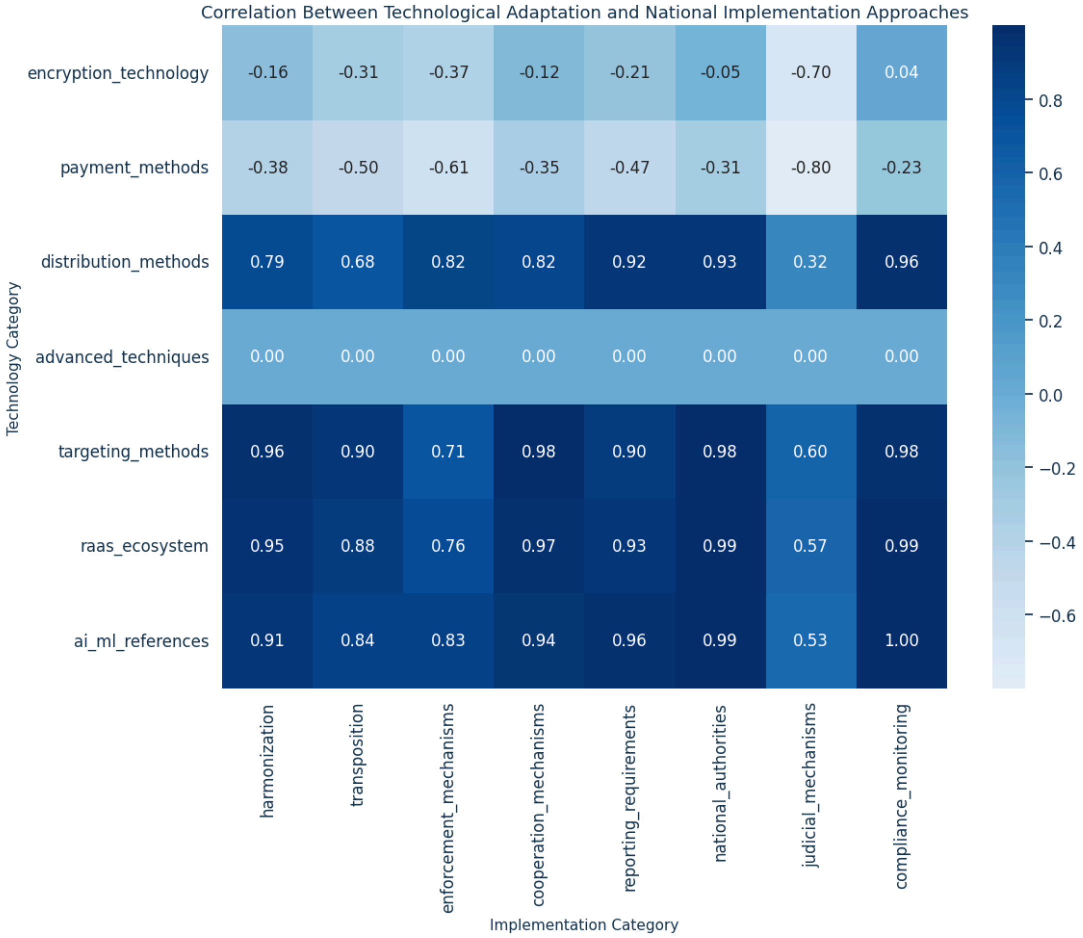
```
    # Document overview
    report.append("\n## 2. Document Overview")

    for doc_id, doc_info in sorted(self.documents.items(), key=lambda x: x[1]['year']):
        report.append(f"\n### {doc_info['title']} ({doc_info['year']})")
        report.append(f"- Type: {doc_info['type']}")
        report.append(f"- Issuing Body: {doc_info['body']}")

        if doc_id in self.tech_evolution_results:
            # Calculate total tech evolution references
            tech_total = sum(category_data['count'] for category_data in
self.tech_evolution_results[doc_id].values())
            report.append(f"- Technology Evolution References: {tech_total}")

        if doc_id in self.implementation_results:
            # Calculate total implementation references
            impl_total = sum(category_data['count'] for category_data in
self.implementation_results[doc_id].values())
            report.append(f"- National Implementation References: {impl_total}")

    # Technological adaptation analysis
    report.append("\n## 3. Adaptation to Ransomware's Technological Evolution")

    for category in self.tech_evolution_categories:
        report.append(f"\n### {category.replace('_', ' ').title()}")

        for doc_id, doc_info in sorted(self.documents.items(), key=lambda x: x[1]['year']):
            if doc_id in self.tech_evolution_results and category in self.tech_evolution_results[doc_id]:
                count = self.tech_evolution_results[doc_id][category]['count']
                report.append(f"\n#### {doc_info['title']} ({doc_info['year']})")
                report.append(f"- Frequency: {count}")

                if count > 0:
                    top_terms = self.tech_evolution_results[doc_id][category]['matches'][:5]
                    terms_str = ", ".join([f"{term} ({count})" for term, count in top_terms])
                    report.append(f"- Top terms: {terms_str}")

        # Add section for evolution over time for this category
        if category in self.temporal_evolution['tech_evolution']['by_category']:
            category_data = self.temporal_evolution['tech_evolution']['by_category'][category]
            if len(category_data) > 1:  # Only if we have multiple data points
                report.append("\n#### Evolution Over Time")

                for item in category_data:
                    doc_id = item['document']
                    report.append(f"- {self.documents[doc_id]['title']} ({item['year']}): {item['count']} mentions")

    # Articles referencing technological adaptation
    report.append("\n### Key Legal Provisions Addressing Technological Evolution")

    for doc_id, doc_info in sorted(self.documents.items(), key=lambda x: x[1]['year']):
        if doc_id in self.articles_tech_references and self.articles_tech_references[doc_id]:
            report.append(f"\n#### {doc_info['title']} ({doc_info['year']})")

            for article_num, article_data in self.articles_tech_references[doc_id].items():
                report.append(f"\n- Article {article_num}: {article_data['title']}")

        # Compare specific categories
        report.append("\n#### Evolution by Technology Category")

        for category in self.tech_evolution_categories:
            if (category in self.tech_evolution_results[earliest_id] and
                category in self.tech_evolution_results[latest_id]):

                earliest_count = self.tech_evolution_results[earliest_id][category]['count']
                latest_count = self.tech_evolution_results[latest_id][category]['count']

                category_display = category.replace('_', ' ').title()
                report.append(f"\n- **{category_display}**: {'Increased' if latest_count > earliest_count else
'Decreased'} from {earliest_count} to {latest_count} mentions")

    # Implementation evolution over time
    report.append("\n### National Implementation Evolution")

    if earliest_id in self.implementation_results and latest_id in self.implementation_results:
        # Calculate total implementation mentions for earliest and latest docs
        earliest_total = sum(category_data['count'] for category_data in
self.implementation_results[earliest_id].values())
        latest_total = sum(category_data['count'] for category_data in
self.implementation_results[latest_id].values())

        report.append(f"\nFrom {earliest_doc[1]['year']} to {latest_doc[1]['year']}, references to national
implementation mechanisms have {'increased' if latest_total > earliest_total else 'decreased'} from
{earliest_total} to {latest_total} mentions.")

        # Compare specific categories
        report.append("\n#### Evolution by Implementation Category")

        for category in self.implementation_categories:
            if (category in self.implementation_results[earliest_id] and
                category in self.implementation_results[latest_id]):

                earliest_count = self.implementation_results[earliest_id][category]['count']
                latest_count = self.implementation_results[latest_id][category]['count']

                category_display = category.replace('_', ' ').title()
                report.append(f"\n- **{category_display}**: {'Increased' if latest_count > earliest_count else
'Decreased'} from {earliest_count} to {latest_count} mentions")

    # Unique approaches
    report.append("\n## 6. Unique Approaches in Each Framework")

    if 'unique_approaches' in self.cross_document_comparison:
        # Unique tech approaches
        report.append("\n### Unique Technological Adaptation Approaches")

        for doc_id, unique_approaches in
self.cross_document_comparison['unique_approaches']['tech_evolution'].items():
            if unique_approaches:
                doc_info = self.documents[doc_id]
                report.append(f"\n#### {doc_info['title']} ({doc_info['year']})")

                for category, terms in unique_approaches.items():
                    category_display = category.replace('_', ' ').title()
                    report.append(f"- **{category_display}**: {', '.join(terms)}")

        # Unique implementation approaches
        report.append("\n### Unique National Implementation Approaches")

        for doc_id, unique_approaches in
self.cross_document_comparison['unique_approaches']['implementation'].items():
            if unique_approaches:
                doc_info = self.documents[doc_id]
                report.append(f"\n#### {doc_info['title']} ({doc_info['year']})")

                for category, terms in unique_approaches.items():
                    category_display = category.replace('_', ' ').title()
                    report.append(f"- **{category_display}**: {', '.join(terms)}")

    # Conclusion
    report.append("\n## Conclusions and Implications")

    # Technological adaptation conclusions
    report.append("\n### Ransomware's Technological Evolution: Framework Adaptations")
    report.append("\nBased on the analysis of the four legal instruments spanning from 2001 to 2022,
the following conclusions can be drawn about how these frameworks adapt to ransomware's
technological advancements:")

    # Get the key tech evolution findings
    if self.cross_document_comparison and 'tech_evolution_matrix' in
self.cross_document_comparison:
        tech_matrix = self.cross_document_comparison['tech_evolution_matrix']

        # Most addressed tech categories
        most_addressed_tech = tech_matrix.sum().sort_values(ascending=False).index[0]
        most_addressed_tech_display = most_addressed_tech.replace('_', ' ').title()

        # Least addressed tech categories
        least_addressed_tech = tech_matrix.sum().sort_values().index[0]
        least_addressed_tech_display = least_addressed_tech.replace('_', ' ').title()

        report.append(f"\n1. **Focus Areas**: The frameworks primarily focus on
{most_addressed_tech_display} aspects of ransomware, while {least_addressed_tech_display}
receives the least attention.")
```

```
        # Document with most tech references
        most_tech_doc_id = tech_matrix.sum(axis=1).idxmax()
        most_tech_doc = self.documents[most_tech_doc_id]

        report.append(f"\n2. **Most Adaptive Framework**: The {most_tech_doc['title']}
({most_tech_doc['year']}) provides the most comprehensive coverage of ransomware's technological
aspects.")

        # AI and advanced techniques
        ai_refs = tech_matrix['ai_ml_references'].sum()
        advanced_refs = tech_matrix['advanced_techniques'].sum()

        report.append(f"\n4. **Emerging Technologies**: References to AI/ML ({ai_refs} mentions) and
advanced techniques ({advanced_refs} mentions) are {'prominent' if ai_refs + advanced_refs > 20
else 'present but limited' if ai_refs + advanced_refs > 5 else 'minimal'}, suggesting {'strong' if ai_refs
+ advanced_refs > 20 else 'moderate' if ai_refs + advanced_refs > 5 else 'limited'} adaptation to
cutting-edge ransomware developments.")

    # Implementation conclusions
    report.append("\n### National Implementation Influence on Countering Ransomware")
    report.append("\nRegarding the influence of national implementation on the frameworks' ability to
counter ransomware, the analysis reveals:")

    # Get the key implementation findings
    if self.cross_document_comparison and 'implementation_matrix' in
self.cross_document_comparison:
        impl_matrix = self.cross_document_comparison['implementation_matrix']

        report.append(f"\n1. **Implementation Priorities**: The frameworks emphasize
{most_addressed_impl_display} as the primary mechanism for national implementation, while
{least_addressed_impl_display} receives less attention.")

        # Document with most implementation references
        most_impl_doc_id = impl_matrix.sum(axis=1).idxmax()
        most_impl_doc = self.documents[most_impl_doc_id]

        report.append(f"\n2. **Implementation Framework**: The {most_impl_doc['title']}
({most_impl_doc['year']}) provides the most comprehensive guidance on national implementation
mechanisms.")

    # Download the report
    files.download('ransomware_adaptation_implementation_report.md')
    print("\nReport download initiated. Check your downloads folder for
'ransomware_adaptation_implementation_report.md'")

    return report

# If ran after part1 and part2, you can call this function with your analyzer instance
# generate_and_save_report(analyzer)
Requirement already satisfied: PyPDF2 in /usr/local/lib/python3.11/dist-packages (3.0.1)
```

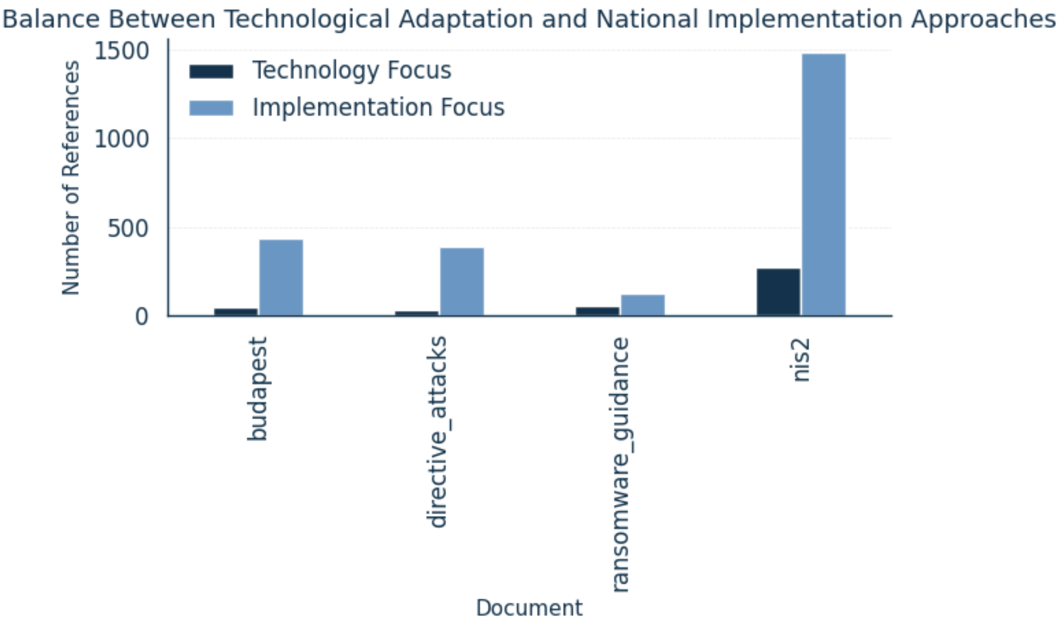
```
Saving T-CY Ransomware guidance note.pdf to T-CY Ransomware guidance note (11).pdf
Saving DIRECTIVE (EU) 2022 2555.pdf to DIRECTIVE (EU) 2022 2555 (11).pdf
Saving DIRECTIVE 2013 40 EU.pdf to DIRECTIVE 2013 40 EU (11).pdf
Saving CETS_185.pdf to CETS_185 (11).pdf

Uploaded files: ['T-CY Ransomware guidance note (11).pdf', 'DIRECTIVE (EU) 2022 2555 (11).pdf',
'DIRECTIVE 2013 40 EU (11).pdf', 'CETS_185 (11).pdf']
Budapest Convention: 66824 characters extracted
Directive on Attacks: 36435 characters extracted
Ransomware Guidance: 21759 characters extracted
NIS 2 Directive: 149627 characters extracted

Loaded Convention on Cybercrime (Budapest Convention) (2001)
Loaded Directive on Attacks Against Information Systems (2013/40/EU) (2013)
Loaded T-CY Guidance Note #12: Aspects of Ransomware covered by the Budapest Convention
(2022)
Loaded Directive on measures for a high common level of cybersecurity across the Union (NIS 2
Directive) (2022)

Correlation Matrix:

                      harmonization  transposition  enforcement_mechanisms  \
encryption_technology     -0.162121      -0.306657                -0.37079
payment_methods           -0.382049      -0.496902               -0.609399
distribution_methods       0.786075       0.676487                0.824977
advanced_techniques               0              0                       0
targeting_methods          0.964641       0.902911                0.705983
raas_ecosystem             0.951184       0.882906                0.756658
ai_ml_references           0.914309       0.838195                0.834308

                      cooperation_mechanisms  reporting_requirements  \
encryption_technology              -0.115905               -0.213412
payment_methods                    -0.345974                -0.47458
distribution_methods                0.823003                0.918656
advanced_techniques                        0                       0
targeting_methods                   0.979152                0.899649
raas_ecosystem                      0.969178                0.928018
ai_ml_references                    0.937222                0.962849

                      national_authorities  judicial_mechanisms  \
encryption_technology            -0.049056            -0.704041
payment_methods                  -0.308918            -0.799433
distribution_methods              0.926332             0.322921
advanced_techniques                      0                    0
targeting_methods                  0.98447             0.595362
raas_ecosystem                    0.992983              0.57262
ai_ml_references                  0.990871             0.534242

                      compliance_monitoring
encryption_technology              0.039915
payment_methods                   -0.230112
distribution_methods               0.964009
```

```
advanced_techniques           0
targeting_methods      0.977341
raas_ecosystem         0.992226
ai_ml_references       0.999629
Correlation dtypes:
harmonization             object
transposition             object
enforcement_mechanisms    object
cooperation_mechanisms    object
reporting_requirements    object
national_authorities      object
judicial_mechanisms       object
compliance_monitoring     object
dtype: object
Correlation Matrix after conversion:
                       harmonization  transposition  enforcement_mechanisms  \
encryption_technology      -0.162121      -0.306657               -0.370790
payment_methods            -0.382049      -0.496902               -0.609399
distribution_methods        0.786075       0.676487                0.824977
advanced_techniques         0.000000       0.000000                0.000000
targeting_methods           0.964641       0.902911                0.705983
raas_ecosystem              0.951184       0.882906                0.756658
ai_ml_references            0.914309       0.838195                0.834308

                       cooperation_mechanisms  reporting_requirements  \
encryption_technology               -0.115905               -0.213412
payment_methods                     -0.345974               -0.474580
distribution_methods                 0.823003                0.918656
advanced_techniques                  0.000000                0.000000
targeting_methods                    0.979152                0.899649
raas_ecosystem                       0.969178                0.928018
ai_ml_references                     0.937222                0.962849

                       national_authorities  judicial_mechanisms  \
encryption_technology             -0.049056            -0.704041
payment_methods                   -0.308918            -0.799433
distribution_methods               0.926332             0.322921
advanced_techniques                0.000000             0.000000
targeting_methods                  0.984470             0.595362
raas_ecosystem                     0.992983             0.572620
ai_ml_references                   0.990871             0.534242

                       compliance_monitoring
encryption_technology               0.039915
payment_methods                    -0.230112
distribution_methods                0.964009
advanced_techniques                 0.000000
targeting_methods                   0.977341
raas_ecosystem                      0.992226
ai_ml_references                    0.999629
Correlation dtypes after conversion:
harmonization             float64
transposition             float64
```

```
enforcement_mechanisms    float64
cooperation_mechanisms    float64
reporting_requirements    float64
national_authorities      float64
judicial_mechanisms       float64
compliance_monitoring     float64
dtype: object
```



	harmonization	transposition	enforcement_mechanisms	cooperation_mechanisms	reporting_requirements	national_authorities	judicial_mechanisms	compliance_monitoring
encryption_technology	-0.16	-0.31	-0.37	-0.12	-0.21	-0.05	-0.70	0.04
payment_methods	-0.38	-0.50	-0.61	-0.35	-0.47	-0.31	-0.80	-0.23
distribution_methods	0.79	0.68	0.82	0.82	0.92	0.93	0.32	0.96
advanced_techniques	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
targeting_methods	0.96	0.90	0.71	0.98	0.90	0.98	0.60	0.98
raas_ecosystem	0.95	0.88	0.76	0.97	0.93	0.99	0.57	0.99
ai_ml_references	0.91	0.84	0.83	0.94	0.96	0.99	0.53	1.00


Balance Between Technological Adaptation and National Implementation Approaches
Technology Focus
Implementation Focus
Number of References
budapest
directive_attacks
ransomware_guidance
nis2
Document