Disjunctive Multi-Level Digital Forgetting Scheme

Marwan Adnan Darwish
Delft University of Technology

Georgios Smaragdakis
Delft University of Technology

ABSTRACT

The virtue of data forgetting has become a substantial demand in the digital era. Once online content has served its purpose, the concept of forgetting arises to ensure that data remains private between data owners and service providers. Despite significant advancements in supporting data forgetting through approaches like access heuristics, elastic expiration times, and manual revocation, the existing research falls short in addressing the demand for a multi-level forgetting structure that can cater to diverse audience-based expiration requirements while considering additional criteria. To the best of our knowledge, no prior works have investigated this gap, emphasizing the need for a comprehensive solution that can effectively accommodate the varying expiration needs of different audience groups. In this paper, we introduce a novel disjunctive multi-level forgetting scheme designed to meet the aforementioned demand for data forgetting. Our scheme introduces unique expiration periods for the encrypted data the service provider stores, called levels. Users are grouped into different levels based on priorities assigned by the data owners. Each level corresponds to a specific expiration threshold, enabling designated user groups to access the content within its validity period before it is forgotten. This approach enables selective data forgetting for one group while enabling concurrent access and retention for other user groups until the stipulated expiration period elapses. To achieve this, we have devised a cutting-edge system that integrates a hierarchical and dynamic scheme utilizing a key decay for managing expiration periods. Moreover, we introduce an innovative approach that harnesses smart contracts on a local Ethereum blockchain to enforce regulations and streamline the secure and efficient expiration and deletion of data. Finally, we thoroughly evaluate our proposed scheme, focusing on decay sensitivity, computational complexity, and rigorous security analysis.

CCS CONCEPTS

• **Security and privacy** → **Privacy-preserving protocols; Privacy protection**;

KEYWORDS

Digital Forgetting; Audience-Based Expiration; Multi-Level Scheme

ACM Reference Format:
Marwan Adnan Darwish and Georgios Smaragdakis. 2024. Disjunctive Multi-Level Digital Forgetting Scheme. In *The 39th ACM/SIGAPP Symposium on Applied Computing (SAC '24), April 8–12, 2024, Avila, Spain.* ACM, New York, NY, USA, Article 4, 10 pages. https://doi.org/10.1145/3605098.3635904

1 INTRODUCTION

In our information-driven world, data storage and access have become incredibly efficient and affordable. This era of "big data" has seen a remarkable surge in global data generation. According to the Internet Data Center (IDC) [16] report, global digital data jumped from 4.4 zettabytes in 2013 to an astounding 40 trillion gigabytes by the end of 2020. This exponential growth is projected to reach an estimated 175 zettabytes by 2025. While this abundance of data offers numerous advantages, it also poses real challenges. However, such data storage and utilization raise concerns regarding privacy and data protection [11, 18]. As a response to these concerns, *General Data Protection Regulation* (GDPR) was enacted by the European Union, emphasizing the *Right to Erasure*, also known as the *Right to be Forgotten* [14, 20]. This concept aims to give individuals control over their data, allowing them to request the removal or deletion of their data from various platforms and databases [2]. To facilitate the process of digital data forgetting, several techniques have been proposed, including Vanish [9], EphPub [4], and Neuralyzer [27]. These approaches primarily focus on uploading online data to service providers in a manner that obscures its content, along with providing instructions on how to retrieve the private keys (i.e., decryption keys) during static or flexible expiration periods. They rely on ephemeral storage solutions like *Domain Name System* (DNS) [12] and *Distributed Hash Table* (DHT) [22] to store and forget the private keys securely.

Unfortunately, previous studies have predominantly focused on a single level of forgetting, where the decryption keys are simultaneously accessible to all users during the designated expiration period. Moreover, in these approaches, the decryption keys are entirely forgotten (i.e., destroyed) for all users once the validity period ends. However, in order to cater to the diverse requirements of different user groups, it is essential to implement privacy controls that allow for customized exposure management of ephemeral data. This means that decisions regarding data exposure should not uniformly impact all readers but should instead consider individual users or groups of users and their specific needs [18]. The need for more control in forgetting the private keys within a single-level scheme results in inflexibility and imposes challenges when disseminating content to multiple levels. By employing the same expiration periods for all recipients, the ability to effectively manage the level of forgetting for stored data becomes limited [19].

Given these limitations, our research endeavors to overcome existing approaches' shortcomings by introducing an innovative ***Disjunctive Multi-Level Forgetting Scheme***. Our primary focus is presenting a multi-level structure that caters to diverse groups. Our novel scheme is constructed upon the foundation of the *key decay* framework [6], which ephemeral key embodies the concept of gradual deterioration from a state of integrity to complete corruption.

SAC '24, April 8–12, 2024, Avila, Spain

ACM ISBN 979-8-4007-0243-3/24/04.
https://doi.org/10.1145/3605098.3635904

In our approach, the multi-level scheme leverages a shared key, precisely the decryption key, associated with the same content but with distinct expiration times determined by owners. By allowing variations in the validity period across different groups, we aim to enhance flexibility and control over online content. Importantly, our proposed scheme achieves its objectives without needing central ephemeral storage for the keys or expiration periods. Our approach involves encrypting and uploading online content using the key decay concept. This encrypted content is then stored on the service provider's platform, where recipients with valid access privileges can download it. Our scheme categorizes recipients into distinct levels, each corresponding to a specific user group within the system. The levels in our scheme serve as discrete categories that include all users and are defined by specific rules, expiration periods, and validity periods to reconstruct the ephemeral keys. As a result, each user is assigned to a particular group based on the decision of the data owner. The data owner maintains the authority to determine the expiration periods by directly setting them or delegating this responsibility to the providers. Access to and downloading the content are contingent upon meeting the specified criteria for each group. This variation in expiration periods of the ephemeral key enables the same content to remain visible in one group while being forgotten simultaneously in another. As the decay progresses, all groups will forget the key, rendering the content useless.
In summary, our research makes the following contributions:

- We introduce the concept of multi-level key decay as a novel approach for achieving digital forgetting, showcasing its ability to provide flexibility and dynamism in data management.
- We develop a comprehensive scheme that ensures the key's partial and complete ephemerality, addressing the specific needs of different user groups.
- We evaluate the effectiveness of our disjunctive multi-level approach by implementing a prototype and conducting analyses on decay sensitivity, computational complexity, and security aspects.

2 CONCEPT

This section introduces terminology, design goals and provides a high-level view of the proposed scheme (see Figure 1).

2.1 Terminology

We define the fundamental notions used in this study, which include:
Disjunctive Multi-Level Scheme. A hierarchical organization of audience (i.e., user groups), each with unique data expiration and access control. Users belong to a single group, unlike the conjunctive scheme, where they can be in multiple groups.
Sender. The entity responsible for encrypting and transmitting data, setting expiration periods and generating the initial encryption key.
Receiver. The intended recipient with access to encrypted data is responsible for decryption and access based on expiration and group settings.
***Encrypted Data Object* (EDO).** Data protected by the scheme, including encrypted data, key generation points (i.e., random sources), checksum, and predefined group rules.

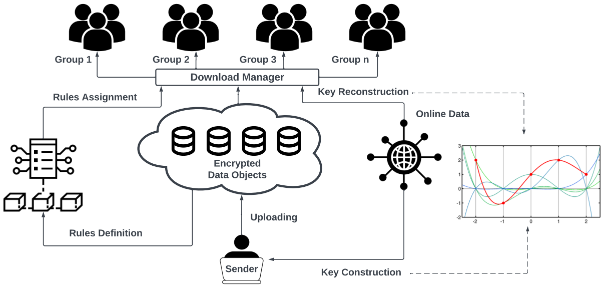


Figure 1: Proposed system model.

Platform Provider. The service that hosts encrypted data provides infrastructure and tools for data management (e.g., Google Drive, Dropbox).

2.2 Security Goals

Numerous security factors must be considered to make a suggested solution reliable and feasible. Consequently, we formalize our security objectives in the following:
Goal 1 – Guarantee a Multi-Level Forgetting Structure. The scheme should support the flexible configuration of expiration periods for different groups based on the preferences of the data owners.
Goal 2 – Achieve Retrospective Privacy. The scheme should prevent unauthorized access to the data only after the expiration of the decryption key, ensuring that expired content remains private[1].
Goal 3 – Ensure the Ephemerality of the Key. The scheme should ensure that decryption keys are wholly forgotten at a specific point. Achieving *Goal 1* results in varying key expiration times across different levels, enhancing system security and flexibility. *Goal 2*, retrospective privacy prevents attackers from retroactively exploiting data, maintaining privacy even if they gain access to expired data. Finally, *Goal 3* ensures secure key destruction across all groups, minimizing the risk of unauthorized access or decryption of data after key decay, ensuring long-term security.

2.3 Overall Architecture

Our scheme provides multi-level forgetting periods built over a key decay scheme [6]. The data owner encrypts online content using a key generated from the decay scheme. Key generation involves fitting online data from random sources onto a Lagrange-basis polynomial [7]. After uploading the content, the download manager divides users into groups, each with its threshold for key reconstruction. This distribution process is facilitated through smart contracts, determining the key's lifespan. Each group can reconstruct the key during its designated period, decrypting the content. Our system allows data owners to dynamically set expiration preferences and easily assign or reassign receivers based on these preferences to different groups. As the decay period approaches, some groups experience key decay while others can still access it. Eventually,

[1] While this assumption may seem idealistic, it serves to demonstrate the concept of retrospective privacy. We recognize that real-world attackers may not always follow such protocols (e.g., not persistently storing unencrypted copies of the data) [18].

the key decays for all groups after a final period, achieving digital forgetting. This approach offers tailored audience-dependent expiration periods, enhancing data forgetting control.

3 SCHEME DESCRIPTION

This section thoroughly examines the proposed system model, covering four key aspects: (i) Key decay scheme background, (ii) Multi-level forgetting structure, (iii) Smart contract integration, and (iv) Expiration and group factors. Table 1 summarizes the notation used in this study.

3.1 Background on Key Decay Scheme

This section introduces the core framework of our proposed system, the key decay scheme. This scheme utilizes *Single-Level Forgetting*, where the level represents a distinct group of users without subdivisions or overlapping. Users follow the same forgetting criterion for uniform decay. The implementation of the key decay scheme encompasses the following phases:

3.1.1 Key Generation Phase. To guarantee the encryption of the content prior to uploading, the decay mechanism relies on random sources to generate a seed for key creation. This creation seed utilizes diverse web sources, including platforms like *Facebook, Twitter, and YouTube*, to obtain specific values such as *views, likes,* and *tweets*. To accomplish this, the system employs a technique that generates a diverse set of online values (i.e., a set of points) capable of fitting into a Lagrange polynomial. The key generation process involves constructing a Lagrange basis polynomial, where each polynomial corresponds to a particular threshold K and the total number of points N that determine the decay rate. Once the ephemeral key is derived, it is employed in the AES 512-bit algorithm to facilitate the encryption phase [5]. This process involves compiling and securely storing the resulting EDO in the cloud by the service provider. The system incorporates several techniques to ensure irreversible decay, including a consensus mechanism that mitigates predictability and increases computational complexity to deter brute force attacks (Proof of Work [3]). In addition, the system utilizes alternating sum and modulo operations to obfuscate the plain values obtained (random source points). The alternating sum operation denoted as $\sum_{i=0}^{k}(-1)^{i}\widehat{\mu_i}$, incorporates a series of switched sign values that contribute to the obfuscation process along with modulo operations represented as $F(x)\ mod\ p$, where $p \in \mathbb{P}$ is greater than N to produce disjointed polynomial [6].

3.1.2 Key Reconstruction Phase. To decrypt the online content for the intended recipients, the Lagrange polynomial is used to reconstruct the key out of EDO structure. The same sources used during the key generation phase are utilized to interpolate the identical polynomial by meeting the threshold K (i.e., minimum points) requirement. Through this scheme, the key is reconstructed, enabling the execution of the content in its original form.

3.1.3 Key Decay Phase. To enable and streamline the process of digital forgetting, the key undergoes gradual modifications over time until it reaches a point of irrecoverability. As time progresses, the online values will change, consequently impacting the coordinates of the Lagrange curve used for key generation. These changes will introduce new corrupted points (i.e., N polynomial points) that differ from the previous ones. The decay scheme detects these alterations and generates a distinct combination of K out of N a ***fixed*** maximum number of points to obtain the seed. In the event that a more significant number of K combination sets are affected by the corruption, surpassing the predefined threshold, the key will decay. This decay will result in all recipients losing the ability to retrieve the key. As a result, the data will become inaccessible, meaning the encrypted data will still exist but will be rendered useless without a retrievable decryption key.

Table 1: Summary of notation.

Notion	Description
x, y	X, Y coordinates used in the Lagrange-basis polynomial
K	Threshold of the polynomial
N	Total number of the polynomial points
l	Lagrange-basis polynomial
$C(N,K)$	Key recovery combination
V	Validity period
D	Decay period
G	Group of level
L	Single level
μ	Stability mean
p	Prime modulus

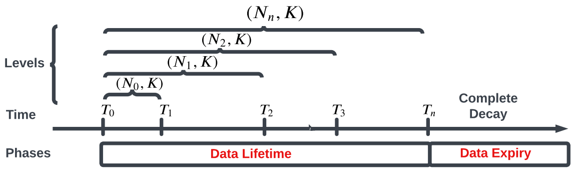


Figure 2: An overview of the multi-level forgetting scheme.

3.2 Multi-Level Forgetting Structure

The proposed scheme includes *Disjunctive Multi-Level Forgetting*, wherein hierarchical levels are utilized to combine users into distinct groups. Unlike uniform decay, this scheme incorporates non-uniform decay, allowing for different rates of decay or expiration for various data or information within each group. The scheme expands the scope of the decay mechanism by serving two primary objectives: (i) **Partial Decay**, enabling selective forgetting for each group, and (ii) **Complete Decay**, representing irreversible decay of the data for all groups. The overall framework of this multi-level scheme throughout the content's lifespan is depicted in Figure 2. The scheme comprises two main phases, outlined as follows:

3.2.1 Uploading Phase. Our system incorporates a key decay scheme to generate the encryption key, also known as the seed. The key generation process relies on a Lagrange-basis polynomial, determined by two key parameters: K minimum number of points and N maximum number of points. To construct the polynomial, we require a set of points, denoted as $P_i(x_i, y_i)$. Specifically, we are given a set of required points K: $(x_1, y_1), (x_2, y_2), ..., (x_k, y_k)$, where no two points have the same x_i within the set. The Lagrange-basis polynomial with a specified threshold is computed using a product formula:

$$l_j(x) = \prod_{i \neq j, 0<i<k} \frac{x - x_i}{x_j - x_i} = \frac{(x - x_0)}{(x_j - x_0)} .... \frac{(x - x_k)}{(x_j - x_k)} \tag{1}$$

The resulting polynomial is then used to interpolate the corresponding y values.

$$F(x) = \sum_{j=0}^{k} y_j l_j(x) \tag{2}$$

The value of $F(0)$ obtained from this interpolation is utilized to generate the required private key. The scheme will generate different levels of N to be fitted within the achieved polynomial, meaning that the total number of points will acquire several polynomial subsets. Using the Formula 2 above, $F(x)$ and a new set of online links (i.e., values), $P_n(x_1, x_2, ..., x_n)$, generate more points N belonging to the exact Lagrange polynomial curve. Consequently, the achieved polynomial will contain a ***dynamic*** total number of points; any K out of N will be sufficient to get the exact polynomial. The main reason for this circumstance is that different N will be distributed and divided between recipients later in the decryption phase. Each sample will result in a unique key recovery period, leading to a different decay rate and a distinct expiration time for that particular combination. Each level contains one group (G) $Levels = \{L_0(G_0), L_1(G_1), \ldots, L_n(G_n)\}$. Once the key is derived, the online content will be encrypted using AES 512-bit before being uploaded to the provider.

3.2.2 Downloading Phase. During this phase, authorized recipients are able to decrypt the online content. This process begins by reconstructing the key using the used points during content uploading. The download manager organizes users into groups based on their expiration periods, which are determined by user preferences. The combinations of key points, represented by the levels $L = C(N, K)$, are then distributed among the corresponding user groups. Each group must fulfill the requirement of having at least K points out of the assigned N_i (i.e., the subset of related polynomial points). Consequently, if one group has more points than another group, it increases the probability of key recovery, as indicated by the binomial coefficient of $\binom{N_i}{K}$ (permutations when $N = K$). To illustrate the range of key recovery possibilities across different levels, random combinations are generated. This process helps showcase the variations in the likelihood of successfully reconstructing the key within each level:

$$Possibilities = \begin{cases} C_0(N_0, K) \\ C_1(N_1, K) \\ \vdots \\ C_n(N_n, K) \end{cases}, N_i \geq K \geq 0 \tag{3}$$

This indicates that the multi-level structure offers the flexibility to define varying expiration thresholds for each group based on specific preferences and the number of possible key recovery combinations. The concept of ephemeral key bits revolves around the decay of particular values within the system after a certain period. This decay process relies on changing random sources to generate these values. Each group in the system has its own unique decay and validity interval, which distinguishes it from other groups. The inequality shown below illustrates the variation in these intervals:

$$\begin{cases} K \leqslant V_0 \leqslant N_0, & \text{if } 0 \leqslant D_0 \leqslant K - 1 \\ K \leqslant V_1 \leqslant N_1, & \text{if } 0 \leqslant D_1 \leqslant K - 1 \\ & \vdots \\ K \leqslant V_n \leqslant N_n, & \text{if } 0 \leqslant D_n \leqslant K - 1 \end{cases} \tag{4}$$

Each level corresponds to a distinct combination of validity (V) and decay rate (D).Online content can be decrypted as long as the threshold for a particular level is not exceeded. However, decay occurs once the threshold is surpassed, rendering the online content invisible. This decay is referred to as *partial decay* for that specific level. Finally, when decay affects all sharing schemes (i.e., all subsets of the Lagrange polynomial $C_n(N_n, K)$), it results in *complete decay* for all recipients. Therefore, the key becomes irretrievable from the original Lagrange-basis polynomial. The central idea here is that the multi-level forgetting scheme offers flexible and dynamic control over distinct levels. Each group is independently isolated through the download manager to determine its specific validity and decay rate, resulting in unique expiration periods.

3.3 Smart Contracts

Using Ethereum-based smart contracts, we streamline data expiration management, automate processes, enforce rules, and set expiration times for groups [23, 26]. This enhances sensitive data protection, reduces complexity and costs, and eliminates the need to store key traces or data fragments. Key management is off-chain, with only group-specific rules stored in Ethereum contracts. The formula below represents parameters for polynomial total points at each level ($L(G)$) with the same threshold:

$$Parameters = \begin{cases} P_0 = (L_0(G_0), K) \\ P_1 = (L_1(G_1), K) \\ \vdots \\ P_n = (L_n(G_n), K) \end{cases} \tag{5}$$

Furthermore, the smart contracts will enforce the shared parameters (i.e., the total number of points and user classification) for each group level to ensure the allowed validity is achieved. Each level can form the combination upon expiration by retaining the respective parameters plus the same threshold shared between all the levels. The validity for each level (i.e., group of users) is described below:

$$V = C_i(N_i, K), where \binom{N_i}{K} \tag{6}$$

$$C_i(N_i, K) = \frac{N_i!}{K!(N_i - K)!} \tag{7}$$

In summary, our proposed approach involves a customized group rule assignment derived from the principles of EDO. This assignment includes the total number of points, and the level of access to a portion of these points is contingent upon user preferences. While a unified threshold is common across all groups, the potential for key reconstruction is based on the specific combination provided.

Algorithm 1 provides a detailed overview of parameter assignment (i denotes levels, j represents groups within a level, and x signifies point assignment variations within those groups). It begins by categorizing users into G groups based on specific criteria (data

Algorithm 1: Smart contract for rule-based key assignment and digital forgetting

```
Require : $G$ (Groups per level)
Ensure  : $V$ (Validity periods for each level)
function
    Rule-Based Key Assignment()
Input: Divide users into $G$ groups based on certain criteria by data
  owners;
for each level $i$ do
    Define groups $G_{ij}$ within level $i$; for each group $j$ in level $i$ do
        Assign point assignments $N_{xij}$ for group $G_{ij}$; Define
          validity period $V_{ij}$ for group $G_{ij}$;
    Assign a common threshold $K_i$ for all groups in level $i$;
for each level $i$ do
    for each group $j$ in level $i$ do
        Provide visibility to the content based on assigned
          combinations $K_i$ and $N_{xij}$;
for each level $i$ do
    for each group $j$ in level $i$ do
        if current decay > $V_{ij}$ expiration time then
            PartialForgetContent($N_{xij}, K_i$);
        else
            ($N_{xij}, K_i,$ ) is still valid;
CompleteForgetContent();
```

owners). For each group i, it defines levels L_{ij} and sets threshold values K_i, validity assignments N_{xij}, and validity periods V_{ij} for each level. The algorithm grants content visibility based on the assigned combinations K_i and N_{xij} for each group and level. When these combinations are determined, points are generated from an EDO and securely shared according to predefined rules, ensuring authorized user access. The contract then checks if the current decay has exceeded the expiration time V_{ij} for each group and level. Upon expiration, it triggers the **PartialForgetContent** function, which removes specific key combinations according to thresholds K_i and validity assignments N_{xij}. Afterward, the **CompleteForgetContent** function initiates comprehensive forgetting, deleting all associated combinations (ephemeral keys), offering a systematic approach for key assignment, validity management, and rule-based digital forgetting in a multi-level data framework.

3.4 Expiration and Group Factors

The proposed multi-level scheme integrates expiration factors and group classification for comprehensive key expiration [10, 21]. These factors and criteria regulate key validity and decay, categorized into levels assigned to user groups for customized expiration periods, enhancing key management and content control.

3.4.1 *Expiration Factors.*

Essential factors in determining expiration thresholds include:

Data Sensitivity. Influences the expiration period, shorter for highly sensitive data.

Security and Usability Balance. Ensures security and convenient access.

Data Nature. Varies based on data type, requiring tailored expiration.

Regulatory Compliance. Aligns with industry-specific regulations. In addition to the factors discussed earlier, our scheme incorporates the concept of key decay rate, known as "Evolution." This aspect is crucial for understanding how keys deteriorate over time, influencing our key management and content control strategies. Owners can customize evolution to align with group expiration periods stored and managed within smart contracts. We aimed to grasp how incremental values progressively corrupt keys until they decay completely. Our study encompasses three distinct decay rates(λ): (i) **Low Rate:** Implies minimal change over time, resulting in keys deteriorating very slowly, (ii) **Moderate Rate:** Involves an average increment over time, leading to a gradual decay of keys, and (iii) **High Rate:** Features a significant increment rate, causing rapid key decay. These decay rates enable effective key management, gradually obscuring and estimating approximate expiring keys based on chosen parameters and influencing factors.

3.4.2 *Group Classification.*

Users are categorized into groups based on criteria like membership or access rights, allowing allocation of expiration thresholds. Two approaches are:

Owner-Defined Expiration: Owners set unique expiration periods for each group.

Membership Level-Based Expiration: Duration based on membership level or privileges.

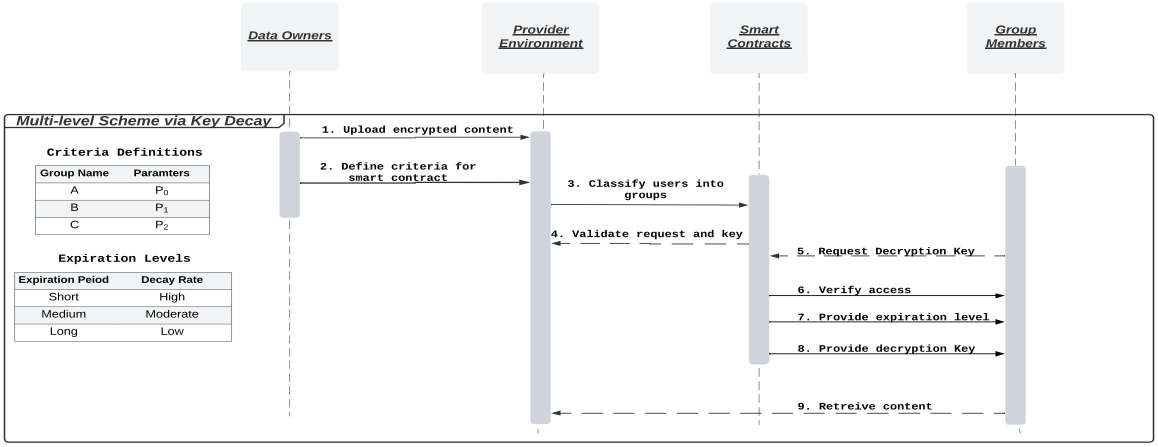
The scheme offers fine-grained control over content expiration, enabling explicit rules or data-driven insights.

4 SYSTEM DESIGN AND ARCHITECTURE

With a focus on the multi-level forgetting scheme, the suggested system design, shown in Figure 3, offers a thorough overview of our framework. The system design begins with the data owners encrypting their data before uploading it to the provider environment. This encryption process ensures the confidentiality and integrity of the content. To enable the ephemerality of the encryption keys, the system incorporates a decay scheme based on Lagrange polynomials (Formulas 1 and 2). This scheme generates the keys randomly and determines various decay rates (i.e., low, moderate, and high) defined in smart contracts, allowing for controlled and gradual partial key corruption until the complete deterioration. Upon EDO upload to the provider by compiling all points needed for reconstruction, owners can establish criteria for categorizing users into distinct levels and degrees of forgetting (see Section 3.4). This classification process enables personalized and finely-grained control over the assigned expiration levels for each group. The multi-level forgetting structure provides varying expiration thresholds tailored to user preferences or administrator decisions, ensuring a customized approach to key expiration. The group classification ($G_i, \quad 1 \leq i \leq G$) and corresponding expiration periods (E_i) for the G groups can be represented as:

$$E_i = \begin{cases} E_i, & \textit{if Owner-assigned} \\ f(m_i), & \textit{if Provider-assigned} \end{cases} \tag{8}$$

These formulas capture the group classification and expiration periods, respectively. Each group is allocated a unique classification and expiration period, enabling personalized control over expiration


Data Owners
Provider Environment
Smart Contracts
Group Members
Multi-level Scheme via Key Decay
Criteria Definitions
Expiration Levels
1. Upload encrypted content
2. Define criteria for smart contract
3. Classify users into groups
4. Validate request and key
5. Request Decryption Key
6. Verify access
7. Provide expiration level
8. Provide decryption Key
9. Retreive content

Group Name	Paramters
A	P_0
B	P_1
C	P_2

Expiration Peiod	Decay Rate
Short	High
Medium	Moderate
Long	Low

Figure 3: System design and architecture: Multi-level forgetting scheme, smart contract integration, and expiration control.

levels based on user preferences. In cases where the administrator determines the classification (i.e., owners assign this role), the membership level m_i defines the rule or criterion utilized to ascertain the expiration period. Subsequently, the group classification information is transmitted to the smart contract, which acts as a decentralized authority responsible for managing and enforcing the defined criteria in EDO. Upon receiving the group classification information, the smart contract commences the distribution of expiration levels to the group members, as denoted by Formulas 5 and 7. Each user or group within the system is assigned a unique identifier or profile. The smart contract initiates a confirmation request to the cloud provider to ensure accurate implementation and enforcement of the expiration thresholds. This confirmation step bolsters the security and validates the correct application of expiration thresholds. Upon receiving a request for decryption keys from a recipient with valid access, the smart contract verifies their group membership and eligibility. We have a set of recipients, denoted as R, and a set of groups, denoted as G. Each recipient, represented by $r \in R$, has a membership status, m_r, which can be binary: $m_r = 1$ for group members and $m_r = 0$ for non-members. Eligibility for decryption keys depends on this status and potential additional criteria. A smart contract handles verification: if a recipient, m_r, equals 1 and meets specified criteria, the contract distributes necessary expiration periods from EDO. It also provides key recovery parameters to eligible recipients $r \in R$, including P_r (Lagrange polynomial formula) and C_r (specific key combinations). With these parameters (i.e., combination, rates, and expiration time), recipients can reconstruct decryption keys and access encrypted content. Expiration levels vary, resulting in different decay rates and durations. Short expiration levels (high decay rate) lead to faster key expiration, while prolonged expiration levels offer extended access (low decay rate). Following this stage, the partial decay process varies across distinct user groups, adapting to their specific requirements and access privileges. However, once the final time threshold is reached, all groups across different levels will collectively forget the random sources responsible for generating the polynomial. Consequently, this collective forgetting prevents them from recalling the key, particularly since the key is not stored; it is solely generated from the EDO to create and obscure the origin of the points. Implementing this scheme facilitates selective forgetting, allowing for managing decay rates associated with the expiration period to formalize group access. Simultaneously, it ensures complete decay occurs to promote digital oblivion.

Table 2: Parameters used in the simulation study.

Parameters	Description	Variable Range
K	Threshold of the polynomial	$[3, 5, \ldots, 60]$
N	Total number of the polynomial points	$[10, \ldots, 120]$
λ	Decay rate	$[0.1 \ldots, 100]$
μ	Stability mean	$\in \mathbb{Z}, \quad \widehat{\mu_i} \in [-50, 50]$
m	Total number of entities	$[1, \ldots, 20]$

5 RESULTS

We created a prototype of our multi-level scheme with key decay on the local Ethereum blockchain using a YouTube dataset and pagination-supported APIs for key seed generation [8]. This seed was employed to fit a Lagrange-basis polynomial, accommodating static and exponential decay rates for efficient data expiration. Our rigorous experiments, conducted on a MacBook with a 2.3 GHz Dual-Core Intel Core $i5$ processor and 8 GB of memory, explored various parameters (listed in Table 2), including K, N, λ, μ, and m, to comprehensively assess their impact on system performance. Our experiments covered three key areas: (i) Decay sensitivity: This term evaluates how different decay fashions impact key expiration and data security, (ii) Computational complexity: This concept assesses the system's efficiency in managing key expiration, and (iii) Security analysis: This term refers to the assessment of the system's security measures.

5.1 Decay Sensitivity

In the evaluation of decay sensitivity, we examine the effects of the decay process on key expiration in various scenarios. Our study emphasizes four fashions, namely (i) Time-based decay, (ii) Granularity decay, (iii) Contextual decay, and (iv) Adaptive decay.

5.1.1 Time-based decay. Focuses on understanding how the decay process influences key expiration over time. By analyzing the decay rate and observing the expiration of keys at different time intervals using a multi-level scheme, we can evaluate the effectiveness of the time-based decay mechanism in managing key expiration. Our evaluation examines two main aspects: partial decay and complete decay.

Partial Decay. To evaluate partial decay per group, we consider a scenario where our system comprises three groups: *short, medium,* and *long* expiration thresholds. It is important to note that, for the sake of explanation, we specifically focus on these three groups (containing a random μ and m) in this instance. We track the decay rate and observe the corresponding key expiration for each group over a specific period of time. This analysis allows us to assess the sensitivity of decay by investigating how different expiration thresholds affect the decay process within each group. Figure 4 provides a visual comparison of the decay rates and key expiration among the groups. In our experiment, the decay level represented the remaining key value using the formula: $DecayLevel = 1 - \lambda$, with λ denoting the decay rate over time. We used exponential decay functions, a common model for natural decay, where the rate is proportional to the current value. Group 1, with a short expiration and high decay rate, saw a rapid key value decline. Group 2, with a medium expiration and moderate decay rate, exhibited slower decay. Group 3, with a long expiration and low decay rate, experienced a gradual decline. This illustrates how expiration thresholds and decay rates impact each group's decay process.

The results showed decay levels over time for the three groups. Group 1 starts with an initial decay level of 1, decays to 50% in about 3 minutes, and to 90% in approximately 6 minutes with a 10-minute expiration. Group 2 starts with a similar decay level, reaches 50% in around 5 minutes, and 90% in about 12 minutes with a 15-minute expiration. Group 3, with a 20-minute expiration, takes about 7 minutes to reach 50% decay and approximately 17 minutes to hit 90%. We deliberately selected different decay rates to create variations in group expiration (Table 2). The plotted variance represents outcomes from multiple experiments with varying decay rates, introducing uncertainty and variability. Adding noise simulates variations that might occur when repeating experiments, with a range of variance around ±0.1 from the original decay curves. The magnitude of added noise dictates the level of variability, underscoring the need for robust experimental design and replication to ensure reliable results. Multiple experiments with different decay rates reveal the range of possible outcomes, assess consistency, and identify outliers or unexpected patterns.

Complete Decay. The complete decay evaluation analyzes the total decay rate and key expiration across all groups, complementing the partial group-specific analysis. Figure 5 illustrates the combined effect of partial and complete decay. To represent the decay process for each group, we use decay intensity functions: $I_1(t)$ for Group 1, $I_2(t)$ for Group 2, and $I_3(t)$ for Group 3. These functions capture the gradual decrease in decay level over time t for each group, expressed as $I(t) = \sum_{i=1}^{3} e^{-\lambda_i t}$. In these formulas, λ_1, λ_2, and λ_3 represent the decay rates for each group. A higher value of λ corresponds to a faster decay rate (i.e., less μ values), resulting in a more rapid decrease in the decay level over time. Each group is represented in the experiment by a unique decay process characterized by its specific decay intensity function. These functions for the three groups are given by $\exp\left(-\frac{\lambda}{5}t\right)$, $\exp\left(-\frac{\lambda}{8}t\right)$, and $\exp\left(-\frac{\lambda}{12}t\right)$, respectively. These values are selected to represent experimental decay rates λ in three distinct groups, showcasing the varying speeds at which keys degrade over time in the experiment. Comparing these rates, Group 1 has the highest λ, indicating the fastest decay, followed by Group 2 and Group 3. Thus, Group 1 experiences the quickest partial decay, decreasing in around 10 minutes, while Group 2 and 3 decay more slowly, taking roughly 15 and 20 minutes, respectively. In Figure 5, the top surface in each plot represents "complete decay" with a constant intensity value of 0.9 throughout the experiment. This intensity value serves as the threshold, indicating that a key is fully decayed and no longer usable for decryption or accessing information when its decay level reaches or surpasses this value. The final decay time is set at 22 minutes, signifying that after this point, all keys in all groups have irreversibly decayed, highlighting the loss of associated information. These visualizations illustrate key decay from their initial values to the final level (typically represented as 0), highlighting the irreversible nature of the process.

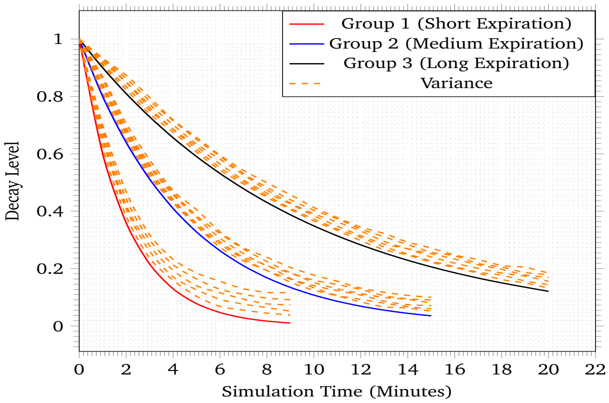


Figure 4: Partial decay evaluation for various groups.

5.1.2 *Granularity decay.*

Our research investigates key expiration across different granularity levels: fine-grained, medium-grained, and coarse-grained categorizations. We aim to understand the decay pattern by tracking key expiration rates over time using specific key combinations. Our investigation involves three distinct groups to align granularity levels and assess their characteristics.

Figure 6 illustrates the relationship between key expiration and granularity levels. Fine-grained keys, offering more key recovery possibilities, exhibit a slower decay rate and lower expiration percentage compared to medium-grained and coarse-grained keys. Fine-grained keys start at 20% expiration and reach 55%, while medium-grained keys begin at 55% and reach 75%, and coarse-grained keys start at 75% and progress to 95% expiration. These findings emphasize the importance of choosing the right granularity level for key categorization. More possibilities within a key combination lead to a slower decay process, offering insights for designing partial decay mechanisms that align with specific granularity requirements.

5.1.3 *Contextual decay.*

We also explore the impact of contextual factors on the decay process and key expiration. Contextual decay takes into account various factors that may influence the decay rate and the expiration of keys. These factors can include the usage

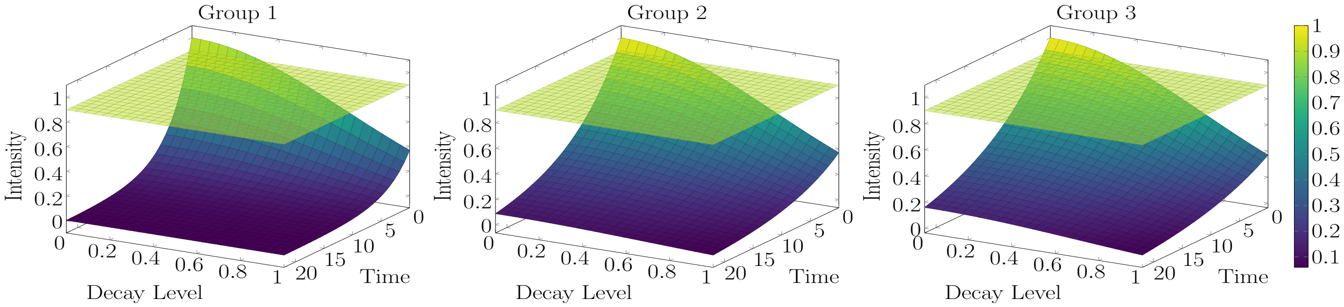


Figure 5: Complete decay evaluation across different groups.

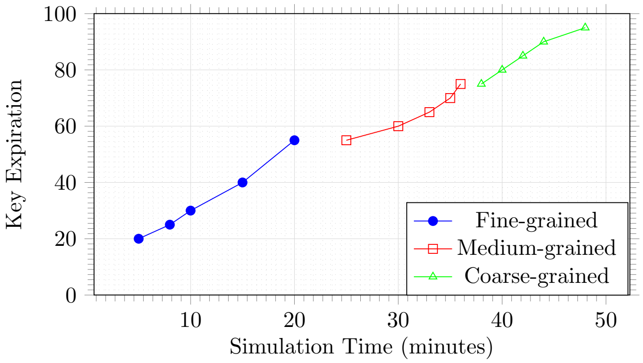


Figure 6: Key expiration and granularity levels.

patterns of keys, the type of data associated with the keys, or the sensitivity level of the data. For example, keys associated with frequently accessed data may have a different decay rate than those associated with rarely accessed data. Similarly, keys containing highly sensitive information may expire shorter than keys with less sensitive information. In our use case, data owners can define rules and parameters for each level, enabling them to customize the decay process according to their data's unique attributes and needs (see Section 3.4).

5.1.4 Adaptive decay. This concept refers to dynamically adjusting the decay process based on specific criteria or feedback. It enables fine-tuning decay parameters like decay rates and expiration thresholds in response to system performance or user needs. In our case, this adaptive decay empowers data owners to proactively manage data, allowing them to shorten or revoke it as needed. This flexibility is achieved by separating levels and rules, allowing for manual revocation or rule changes. Mathematically, adaptive decay can be expressed as follows:

$$\lambda = f(M) = \begin{cases} \lambda_{\text{high}} & \text{if } M > T \\ \lambda_{\text{low}} & \text{if } M < T' \\ \lambda_{\text{default}} & \text{otherwise} \end{cases} \tag{9}$$

In this formula, λ represents the decay rate and M denotes a certain measure or metric (e.g., average access frequency) that influences the adaptive decay process. The function f evaluates these parameters and determines the appropriate decay rate based on the user's preferences or system requirements. The decay rate can be dynamically adjusted by assigning different values to λ_{high}, λ_{low}, and λ_{default}. If the access metric M exceeds a certain threshold T, indicating a higher preference for key expiration, the decay rate is set to λ_{high}. Conversely, if M falls below a threshold T', indicating a lower preference for expiration, the decay rate is set to λ_{low}. For all other cases, where M is within an acceptable range, the decay rate defaults to λ_{default}. Figure 7 demonstrates the scalability of the scheme with four decay rates, although it can accommodate any number. The first rate exhibits exponential decay with a constant of 0.3 and includes sinusoidal oscillations (i.e., periodic fluctuations upon M) at a frequency of 150, leading to decay fluctuations. The second rate has a higher constant of 0.4, resulting in faster decay, with minor sinusoidal fluctuations at 120 frequency. The third rate, with the highest constant of 0.5, decays rapidly and displays noticeable fluctuations due to a 100-frequency sinusoidal component. The last rate, with a constant of 0.2, decays more slowly and includes a 100-frequency sinusoidal component, causing moderate fluctuations. Between 4 and 12 minutes, metrics ($M1 \longrightarrow M5$) introduce additional sinusoidal terms, further altering decay rates and enhancing complexity. Users can fine-tune parameters using the adaptive mechanism to align with their information retention goals and preferences.

5.2 Computational Complexity

We compared the computational requirements of two forgetting schemes, multi-level and key decay, as summarized in Table 3. Our analysis evaluated efficiency, scalability, security, flexibility, and key management aspects, providing insights into their computational characteristics and trade-offs. Our experiments analyzed the time and space complexities of the multi-level and decay schemes. The multi-level scheme exhibited a quadratic time complexity ($O(m^2)$, e.g., 100 ms for 10 data points, 400 ms for 20). Its space complexity $O(s(m))$ grew linearly with data points (e.g., 100 KB for 10 data points, 200 KB for 20). This reflects the impact of decentralization on computational requirements. Conversely, the decay scheme demonstrated linear time complexity ($O(m)$, e.g., 10 ms for 10 data points, 20 ms for 20), making it efficient with larger datasets. The decay scheme's space complexity $O(s(\lambda))$ varied with the decay rate. For instance, with a 0.5 decay rate and 5 KB storage per unit of the decay rate, it was 2.5 KB. Higher decay rates resulted in a faster reduction of required storage space. Both schemes prioritize security by gradually corrupting keys over time. The multi-level scheme provides customized expiration policies and access privileges at the group level, offering flexibility. In contrast, the decay scheme applies a uniform decay process to all keys. In terms of scalability, the multi-level scheme efficiently adapts to varying group numbers and data, accommodating expanding datasets. Conversely, the decay scheme

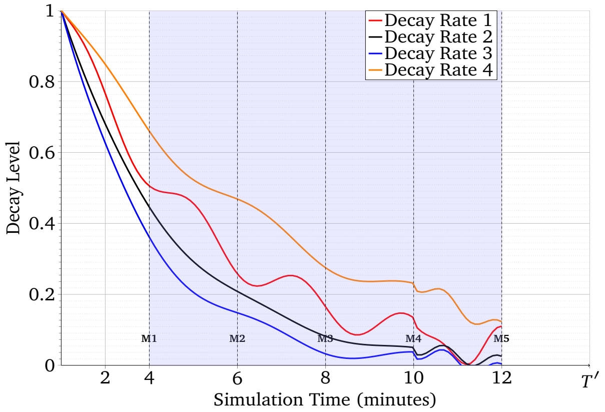


Figure 7: Comparison of partial decay rates in adaptive decay.

has fixed scalability, treating all groups uniformly without the ability to scale. Finally, Key management is a strength of the multi-level scheme, granting data owners control over distinct expiration rules and access privileges for each group. However, the decay scheme lacks such key management capabilities.

5.3 Security Analysis

In this section, we assess the security of the proposed approach regarding various security goals and threats.

Retrospective Privacy. The proposed scheme introduces a novel key management and expiration method that enhances security by ensuring complete key expiration. It incorporates the concept of key decay and utilizes distinct levels with associated contracts for partial decay. This approach significantly reduces the risk of data exposure and unauthorized decryption, even if an attacker gains access to the data later.

Brute Force Attack. The multi-level data forgetting scheme may face brute force attacks, where adversaries attempt to guess the key's origin. To counter this threat, the scheme employs different combinations for each level, making it more complex for attackers to determine the key's origin. Additionally, an obfuscation process conceals the key's sources, further thwarting brute-force attacks (see Section 3.1.1).

Dynamic expiration periods. This research extends the decay scheme to offer multiple key recovery options through partial and complete decay, enabling users to manage decay rates for diverse expiration periods. Smart contracts enhance this by enforcing rules based on user preferences, providing greater flexibility and control over data accessibility duration.

Extending Expiration Periods. The proposed approach enables data owners to update rules and extend expiration periods by publishing new rules to smart contracts, granting flexibility according to evolving circumstances (as discussed in adaptive decay 5.1.4).

Manual Revocation. The scheme supports manual revocation by assigning unique identifiers to user groups. Data owners can revoke access to specific groups by resetting shared parameters through smart contracts without waiting for the decay scheme to complete. Setting the parameters $P = N = 0$ (as in Section 5.1) prevents the reconstruction of the private key combination and the decryption of online content during the validity period.

Table 3: A comparison of Multi-Level and Key Decay schemes.

Aspect	Multi-Level Scheme	Decay Scheme
Time Complexity	$O(m^2)$	$O(m)$
Space Complexity	$O(s(m))$	$O(s(\lambda))$
Security	Key Decay	Key Decay
Flexibility	High	Low
Scalability	Scalable	Fixed
Key Management	Granular Control	Limited

6 RELATED WORK

This section summarizes recent developments in digital forgetting and online data revocation. These studies aim to address the need for secure and efficient deletion of digital information without considering audience-based expiration. Our proposed framework is based on the key decay scheme presented by Marwan et al. [6], allowing keys to exist temporarily without requiring centralized storage during their validity period. However, a notable limitation of this approach is the need for more flexibility in accommodating different user groups. Various techniques have been proposed to handle expiration periods, including flexible and fixed single-level expiration periods for all audiences. Examples include Ephemerizer [13], Vanish [9], Neuralyzer [27], EphPub [4], and Timed Revocation [15]. Another approach [17] uses smart contracts on a local Ethereum blockchain to enforce revocation conditions based on contractual agreements between cloud providers and data owners. In the realm of provable data deletion schemes, Yang et al. [25] introduced the number-rank-based Merkle hash tree (NR-MHT) for efficient data integrity auditing and dynamic data insertion while ensuring provable data deletion for the same audience. For secure deletion in IoT devices, Xiong et al. [24] proposed a key derivation encryption algorithm based on flash memory's hierarchical structure. This algorithm involves simultaneous key deletion for all users, encompassing both cipher form and key-related components when data validity ceases. The Forgits data structure strengthens online deletion by gradually dropping the lowest bits or pixels (least significant bits) from old to new data, enabling infinite retrievals by forgetting older stored data without impacting other user groups [1]. Despite significant progress in digital forgetting techniques, they still face limitations in dynamically revoking access to files across different user groups. Each reviewed technical proposal concentrates on particular functionalities within this context. In order to offer a comprehensive comparison, we thoroughly considered the key metrics within the digital forgetting domain in relation to our proposed scheme; we have summarized our findings in Table 4.

Delete Content. Enables content removal from the platform, giving users control over its accessibility and option to delete it.

Reduce Exposure. Provides selective content forgetting on the platform, allowing users to control access and share it with specific audiences.

Flexibility. Enables customizable expiration periods for digital content, allowing users to specify timeframes for automatic deletion or inaccessibility, providing flexible content lifespan management.

Manual Revocation. Empowers users to revoke access to their content anytime manually. This gives them control over who can access their content and allows them to change permissions or revoke access privileges as needed.

Table 4: Comparative analysis of research studies: Examining different approaches.

Research Name	Delete Content	Reduce Exposure	Flexibility	Manual Revocation	Scalability	User-Level Expiration
Ephemerizer [13]	✓	✗	✗	✗	✗	✗
Vanish [9]	✓	✗	✗	✗	✗	✗
EphPub [4]	✓	✗	✗	✗	✗	✗
Timed Revocation [15]	✓	✗	✗	✓	✗	✗
Neuralyzer [27]	✓	✗	✓	✓	✓	✗
Towards contractual agreements [17]	✓	✗	✓	✓	✓	✗
Provable data deletion scheme [25]	✓	✗	✓	✓	✓	✗
IoT data deletion scheme [24]	✓	✗	✓	✓	✓	✗
Forgits [1]	✓	✗	✗	✗	✗	✗
Key Decay [6]	✓	✗	✓	✗	✗	✗
Proposed Scheme	✓	✓	✓	✓	✓	✓

Scalability. Pertains to the system's ability to efficiently manage a large volume of users and data while maintaining optimal performance.

User-Level Expiration. Enables disjunctive personalized expiration periods for user content, offering individual control over accessibility and deletion timelines.

7 CONCLUSION

We introduced a novel multi-level data forgetting scheme using key decay techniques. The ubiquity of digital data demands solutions for audience-dependent data transience and ephemerality. Our scheme empowers data owners to manage expiration periods for different user groups, providing better content control and customization. This flexibility enhances content management and personalized forgetting, creating a comprehensive framework for optimizing content management while ensuring improved accessibility and privacy across user groups. We have implemented and evaluated a prototype, yielding promising results for our scheme's effectiveness.

ACKNOWLEDGMENTS

This work has received funding from the European Union's research and innovations programmes Horizon 2020 under grant agreement #883275 (HEIR) and Horizon Europe under grant agreements #101092912 (MLSysOps), #101094901 (SEPTON), and #101070052 (TANGO), and from the European Research Council (ERC) under Starting Grant ResolutioNet (679158).

REFERENCES

[1] Azza Abouzied and Jay Chen. 2015. Harnessing data loss with forgetful data structures. In *Proceedings of the Sixth ACM Symposium on Cloud Computing*. ACM, USA, 168–173.

[2] Matt Bishop, Emily Rine Butler, Kevin Butler, Carrie Gates, and Steven Greenspan. 2013. Forgive and forget: return to obscurity. In *Proceedings of the 2013 New Security Paradigms Workshop*. ACM, USA, 1–10.

[3] The Chain Bulletin. 2019. The Chain Bulletin. https://chainbulletin.com/proof-of-work-explained-in-simple-terms.

[4] Claude Castelluccia, Emiliano De Cristofaro, Aurélien Francillon, and Mohamed-Ali Kaafar. 2011. Ephpub: Toward robust ephemeral publishing. In *2011 19th IEEE International Conference on Network Protocols*. IEEE, Canada, 165–175.

[5] Joan Daemen. 2020. *The Design of Rijndael: The Advanced Encryption Standard (AES)*. Springer; 2nd edition, USA.

[6] Marwan Adnan Darwish and Apostolis Zarras. 2023. Digital Forgetting Using Key Decay. In *Proceedings of the 38th ACM/SIGAPP Symposium on Applied Computing*. ACM, USA, 34–41.

[7] Biswajit Das and Dhritikesh Chakrabarty. 2016. Lagrange's interpolation formula: representation of numerical data by a polynomial curve. *International Journal of Mathematics Trend and Technology* 34, 2 (2016), 23–31.

[8] Datasnaek. Accessed: June 29, 2023. YouTube Trending Video Statistics Dataset. https://www.kaggle.com/datasets/datasnaek/youtube-new

[9] Roxana Geambasu, Tadayoshi Kohno, Amit A Levy, and Henry M Levy. 2009. Vanish: Increasing Data Privacy with Self-Destructing Data.. In *USENIX Security*, Vol. 316. USENIX, USA, 10–5555.

[10] Farzane Karami, David Basin, and Einar Broch Johnsen. 2022. DPL: A Language for GDPR Enforcement. In *2022 IEEE 35th Computer Security Foundations Symposium (CSF)*. IEEE, IEEE, Haifa, 112–129.

[11] Viktor Mayer-Schönberger. 2011. *Delete: The virtue of forgetting in the digital age.* Princeton University Press, USA.

[12] Paul V Mockapetris. 1987. RFC1035: Domain Names -Implementation and Specification.

[13] Radia Perlman. 2005. The ephemerizer: Making data disappear.

[14] Eugenia Politou, Efthimios Alepis, and Constantinos Patsakis. 2018. Forgetting personal data and revoking consent under the GDPR: Challenges and proposed solutions. *Journal of cybersecurity* 4, 1 (2018), tyy001.

[15] Sirke Reimann and Markus Dürmuth. 2012. Timed revocation of user data: long expiration times from existing infrastructure. In *Proceedings of the 2012 ACM workshop on Privacy in the electronic society*. ACM, USA, 65–74.

[16] David Reinsel, John Gantz, and John Rydning. 2018. The Digitization of the World From Edge to Core. https://www.seagate.com/files/www-content/our-story/trends/files/idc-seagate-dataage-whitepaper.pdf

[17] Theodor Schnitzler, Markus Dürmuth, and Christina Pöpper. 2019. Towards contractual agreements for revocation of online data. In *IFIP International Conference on ICT Systems Security and Privacy Protection*. Springer, Lisbon, 374–387.

[18] Theodor Schnitzler, Shujaat Mirza, Markus Dürmuth, and Christina Pöpper. 2021. SOK: Managing Longitudinal Privacy of Publicly Shared Personal Online Data. *Proceedings on Privacy Enhancing Technologies* 2021, 1 (2021), 229–249.

[19] Theodor Schnitzler, Christine Utz, Florian M Farke, Christina Pöpper, and Markus Dürmuth. 2018. User perception and expectations on deleting instant messages—or—"what happens if i press this button?". *(GDPR)* 4 (2018), 5.

[20] Esther Shein. 2013. Ephemeral data. *Commun. ACM* 56, 9 (2013), 20–22.

[21] Yanjun Shen, Bin Yu, Shangqi Lai, Xingliang Yuan, Shi-Feng Sun, Joseph K Liu, and Surya Nepal. 2022. OblivSend: Secure and Ephemeral File Sharing Services with Oblivious Expiration Control. In *International Conference on Information Security*. Springer, Springer, Switzerland, 269–289.

[22] Ion Stoica, Robert Morris, David Karger, M Frans Kaashoek, and Hari Balakrishnan. 2001. Chord: A scalable peer-to-peer lookup service for internet applications. *ACM SIGCOMM computer communication review* 31, 4 (2001), 149–160.

[23] Weizheng Wang, Huakun Huang, Zhimeng Yin, Thippa Reddy Gadekallu, Mamoun Alazab, and Chunhua Su. 2023. Smart contract token-based privacy-preserving access control system for industrial Internet of Things. *Digital Communications and Networks* 9, 2 (2023), 337–346.

[24] Jinbo Xiong, Lei Chen, Md Zakirul Alam Bhuiyan, Chunjie Cao, Minshen Wang, Entao Luo, and Ximeng Liu. 2020. A secure data deletion scheme for IoT devices through key derivation encryption and data analysis. *Future Generation Computer Systems* 111 (2020), 741–753.

[25] Changsong Yang, Yueling Liu, Feng Zhao, and Shubin Zhang. 2022. Provable data deletion from efficient data integrity auditing and insertion in cloud storage. *Computer Standards & Interfaces* 82 (2022), 103629.

[26] Chuntang Yu, Yongzhao Zhan, and Muhammad Sohail. 2022. SDSM: Secure Data Sharing for Multilevel Partnerships in IoT Based Supply Chain. *Symmetry* 14, 12 (2022), 2656.

[27] Apostolis Zarras, Katharina Kohls, Markus Dürmuth, and Christina Pöpper. 2016. Neuralyzer: Flexible Expiration Times for the Revocation of Online Data. In *ACM Conference on Data and Application Security and Privacy*. ACM, USA, 14–25.