Preserving Privacy versus Data Retention

Markus Hinkelmann and Andreas Jakoby

Institut für Theoretische Informatik,
Universität zu Lübeck, Germany

{hinkelma, jakoby}@tcs.uni-luebeck.de

SIIM-TR-A-08-04
Report Series of the Institutes for Computer Science and Mathematics,
Universität zu Lübeck

September 18, 2008

Abstract

The retention of communication data has recently attracted much public interest, mostly because of the possibility of its misuse. In this paper, we present protocols that address the privacy concerns of the communication partners. Our data retention protocols store streams of encrypted data items, some of which may be flagged as critical (representing misbehavior). The frequent occurrence of critical data items justifies the self-decryption of all recently stored data items, critical or not. Our first protocol allows the party gathering the retained data to decrypt all data items collected within, say, the last half year whenever the number of critical data items reaches some threshold within, say, the last month. The protocol ensures that the senders of data remain anonymous but may reveal that different critical data items came from the same sender. We call this the affiliation of critical data. Our second, computationally more complex scheme obscures the affiliation of critical data with high probability.

1 Introduction

Recently, governments all over the world have increased their surveillance efforts. In 2006 the European Union adopted directive 2006/24/EC [10], on "the retention of data generated or processed in connection with the provision of publicly available electronic communications services or of public communications networks". Member states have to implement the directive by 2009 as national law. By then, communication service providers must retain data that identify the source and the destination of communication, its type, date and duration for at least six months. Additionally, information about the location of mobile communication equipment has to be recorded. Officials want to use retained data to detect and investigate critical activities.

There already exist several data retention applications outside the realm of communication data, e. g., the tracking of traffic offenses. If a traffic participant misconducts,

the incident is recorded. If too many incidents occur, a driver loses her or his license. After some amount of time this data should be deleted.

Certainly, these issues mean a conflict between investigational interests and preserving the sphere of personal privacy. A simple solution may be as follows: The communication providers encrypt the retained data of an user. If some suspicious facts justify a judicial order to open the stored data, the private decryption key is disclosed to the officials. Thus, the provider has always access to the retained data, at least to decryption keys of the users. The goal of this paper is to present protocols that implement such encryption and decryption processes and feature advanced properties. Our protocols allow the self-decryption of retained data if a threshold of critical activities is passed. Encrypted data that are stored before a prescribed period cannot be decrypted. We also care about the anonymity of users, i.e. the stored data are only related to the encrypted identity of a user. Furthermore, the provider is not responsible to store any user related data except for data needed for the encryption and classification of the actual message. I.e. the provider is not responsible to retain any information about the users' behaviors. For enhanced data privacy third parties (the providers) should not be allowed to store private data longer than necessary. This includes the knowledge about the number of critical activities of a user, too.
Additionally, we propose techniques that data can only be associated with user if it has been retained in the prescribed period and that critical activities cannot be traced for a longer period of time. Up to our knowledge our protocols are the first that ensure this kind of privacy.

Related Work: With the introduction of the Internet data retention, surveillance and privacy have drawn a lot of attention in the fields of sociology and computer science. Marx [19] identified four conditions under that data retention raises ethical concerns: Collecting data involves physical or psychological harm, produce invalid results, crosses a personal boundary without notice, or violates trust. Having access to the data the temptation to misusing them is great. Thus, one should stay alert and minimize this risk. Blanchette and Johnson [4] argue that the important value of social forgetfulness is slipping away since the introduction of electronic data retention. Cryptography provides some hope to counter this threat, e.g. by introducing digital pseudonyms [7, 8]. Nevertheless, electronic wiretapping means an architectured security breach and it is necessary to limit its use to appropriate scenarios [18]. Several papers deal with the technical implementation of data retention [1, 25, 21]. But to our knowledge no scheme proposes solutions for an increased level of privacy in data retention.

Our scheme has the feature that the retained data automatically allow their decryption if a threshold of misbehavior is reached. Therefore, secret sharing will be one important tool. Shamir [24] and Blakley [3] independently introduced secret sharing schemes. These schemes use threshold functions as access structures. Subsequently several schemes using general access structures have been presented (e.g. [2, 15]). Since the original shares are as large as the secret, one might ask to reduce the size of the shares. Czimraz [9] showed that this is not possible for every access structure. Using an information dispersal algorithm [22] and cryptographic encryption Krawczyk [17] proposes a scheme to reduce the share size.

Using Shamir shares our protocols allow that a secret can be decrypted only if a threshold is reached within a determined period of time. If the messages are too old they become useless for the decryption. Rivest et al. [23] introduced the notion of time-release crypto. They propose to use computational puzzles as time-locks to schedule the first point in time when it is possible to decrypt data. In a similar way timed commitments and signatures are implemented by Boneh and Naor [6]. Another aspect of the relationship between time and cryptography are cryptographical timestamps. Haber and Stornetta [13] presented protocols for timestamping that ensure the privacy of the data and deny attacks to change the timestamp.

As second technique, we use pseudorandom number generators (PRNG) to create keys and identification information. Blum and Micali [5], and Yao [26] introduced the notion for cryptographically robust PRNG. Goldreich et al. [12] proved that a PRNG can be used to generate a pseudorandom collection of functions. One may also use a PRNG to implement bit commitment schemes [20]. Given any one-way function Håstad et al. [14] showed that a PRNG can be constructed. Furthermore, they proved that PRNG exist if and only if one-way functions exist.

In Sections 3 and 4 we describe the scenario and present basic protocols. We discuss these protocols and identify a new problem type: the so called history of messages. Our main scheme, presented in section 5, obscures the message history of the users with high probability (w.h.p.). Throughout this work, obvious proofs are omitted, others can be found in the appendix.

2 Preliminaries

Let X be a discrete random variable that takes values from the set of real numbers or strings over the alphabet Σ. If X is uniformly distributed, we also write $X \in_R \Sigma^*$. Using a function symbol f we write $\Pr[f(X) = y]$ for $\sum_{x:\, f(x)=y} \Pr[X = x]$.

Pseudorandom Number Generators: Pseudorandom number generators (PRNG) are functions having special properties which make them very suitable for cryptography. If the input (the so called *seed*) of a PRNG is unknown, the output is indistinguishable from random strings for computationally bounded adversaries. On the other hand, PRNGs are deterministic functions. Thus, if the seed is known, we are able to reproduce the output of a PRNG.

Definition 1 *Let $h : \mathbb{N} \to \mathbb{N}$ be a polynomial stretch function such that $h(\ell) > \ell$ for all ℓ. Let S and Y be uniformly distributed discrete random variables taking values in Σ^ℓ and $\Sigma^{h(\ell)}$, respectively. We call a function $G\colon \Sigma^\ell \to \Sigma^{h(\ell)}$ a PRNG if for all probabilistic polynomial-time bounded algorithms $\mathcal{A}$, for all polynomials p and for all sufficiently large ℓ it holds that*

$$|\Pr[\mathcal{A}(Y) = 1] - \Pr[\mathcal{A}(G(S)) = 1]| < \frac{1}{p(\ell)}$$

The following proposition describes how to stretch the output of a pseudorandom number generator. It appears in [14] and is due to an observation made by Goldreich and

Micali. For a string $w = w_1 \dots w_\ell$ and $1 \leq a \leq b \leq \ell$ let $w_{\{a,\dots,b\}}$ be the substring $w_a \dots w_b$. The operator $\circ$ denotes the concatenation of strings.

Proposition 2 *Let $G\colon \{0,1\}^\ell \to \{0,1\}^{\ell+1}$ be a PRNG. We define $G^{(1)}(S) = G(S)$, and inductively, for all $i \geq 1$,*

$$G^{(i+1)}(S) = G(G^{(i)}(S)_{\{1,\dots,\ell\}}) \circ G^{(i)}(S)_{\{\ell+1,\dots,\ell+i\}}.$$

Then, for every polynomial q and sufficiently large ℓ it holds that

$$G^{(q(\ell))}\colon \{0,1\}^\ell \to \{0,1\}^{\ell+q(\ell)}$$

is a PRNG.

Thus, for every polynomial h the function $\widehat{G} = G^{(h(l))}$ is a PRNG. As seen above, we inductively define the PRNGs $\widehat{G}^{(1)}(S) = \widehat{G}(S)$ and

$$\widehat{G}^{(i+1)}(S) = \widehat{G}(\widehat{G}^{(i)}(S)_{\{1,\dots,\ell\}}) \circ \widehat{G}^{(i)}(S)_{\{\ell+1,\dots,\ell+i\cdot h(\ell)\}}$$

for $i \leq q(\ell)$.

Definition 3 *Let $S \in_R \{0,1\}^\ell$ be a random string and $i \in \mathbb{N}$. We define the seed generating function $\mathrm{seed}^{(i)}\colon \{0,1\}^\ell \to \{0,1\}^\ell$ and the pseudorandom string function $\mathrm{rand}^{(i)}\colon \{0,1\}^\ell \to \{0,1\}^{h(\ell)}$ such that $\mathrm{seed}^{(0)}(S) = S$, and for $i \geq 1$ it holds that*

$$\mathrm{seed}^{(i)}(S) = \widehat{G}^{(i)}(S)_{\{1,\dots,\ell\}} \text{ and } \mathrm{rand}^{(i-1)}(S) = \widehat{G}^{(i)}(S)_{\{\ell+1,\dots,\ell+h(\ell)\}}.$$

The following observation is straightforward and we omit its proof.

Observation 4 *For $i \in \mathbb{N}$ we can use $\mathrm{seed}^{(i)}(S)$ to calculate the values $\mathrm{seed}^{(i+1)}(S)$ and $\mathrm{rand}^{(i)}(S)$ by*

$$\widehat{G}(\mathrm{seed}^{(i)}(S)) = \mathrm{seed}^{(i+1)}(S) \circ \mathrm{rand}^{(i)}(S).$$

Starting with a seed S we can generate sequences $\{\mathrm{seed}^{(i)}(S)\}_{i=0}^{\tau}$ and $\{\mathrm{rand}^{(i)}(S)\}_{i=0}^{\tau}$ efficiently. For a time step t we will use $\mathrm{rand}^{(t)}(S)$ to encrypt some data.

Note that if τ gets large, the probability to distinguish between random strings and pseudorandom strings increases. To work against this we may increase ℓ. But to inhibit this threat we can additionally substitute such a seed $\mathrm{seed}^{(t)}(S)$ by a new random string from time to time. In the following we will focus only on the way how we use the two sequences.

Although we can efficiently compute $\mathrm{seed}^{(j)}(S)$ and $\mathrm{rand}^{(j)}(S)$ for $j \geq i$ on input $\mathrm{seed}^{(i)}(S)$, no probabilistic polynomial time algorithm on input $\mathrm{seed}^{(i)}(S)$ or $\mathrm{rand}^{(i)}(S)$ is able to deduce more than negligible information about the strings $\mathrm{seed}^{(k)}(S)$ and $\mathrm{rand}^{(k-1)}(S)$ for $k < i$ due to the cryptographical robustness of the PRNG G. For further information about PRNGs we refer the interested reader to [11].

Shamir's Secret Sharing: Let $\mathcal{F}$ be a field with more than n elements. In [24] Shamir presented a threshold scheme to divide some data $D \in \mathcal{F}$ into n shares $D_1, \dots D_n$ such

that D can only be reconstructed if one knows at least k shares D_i. I.e. let p be a random polynomial over the field $\mathcal{F}$ of degree $k-1$ with $p(0) = D$. For $1 \leq i \leq n$ we can choose $D_i := p(i)$. Having access to k shares we can compute the polynomial p by Lagrange interpolation. If we have obtained less than k shares, then for each value D' we can generate a polynomial p' with $p'(0) = D'$ that is consistent with the obtained shares. Thus, this scheme provides information-theoretical privacy.

Anonymity: Let $\text{subj}(M)$ be the decrypted *subject* of interest about a message M. For example, this may be the information about its sender, receiver, or contents. We assume that $\text{subj}(M)$ is encrypted in the message.

Since our protocols are randomized we regard $\text{subj}(M)$ as a discrete random variable. The set $\mathcal{SU} = \{\text{subj}(M') \mid M' \text{ is a possible message}\}$ describes the set of all possible subjects of interest from all messages.

Definition 5 *Let I be a uniformly distributed discrete random variable that takes values from $\mathcal{SU}$. For a message M we call the information about a message* $\text{subj}(M)$ anonymous *for a party A if for all probabilistic polynomial-time bounded algorithms $\mathcal{A}$, for every polynomial p, for all communication strings c and any content r of the random string of A, for all sufficiently large security parameters k for encryption of* $\text{subj}(M)$ *it holds that*

$$|\Pr[\mathcal{A}(\text{subj}(M), M, c, r) = 1] - \Pr[\mathcal{A}(I, M, c, r) = 1]| \;\leq\; \frac{1}{p(k)} \;.$$

In the following we usually encrypt the subject of a message using a bitwise XOR with a pseudorandom string. Then, we regard the security parameter k as the length of the seed for the PRNG.

3 Basic Structure and Types of Messages

We divide the parties that participate in the process of communication into three groups.

1. Users interact with a system, e.g. surfing in the Internet or sending emails. They want to use these services privately. We also refer to users as senders or receivers.
2. Communication service providers offer and control access to the system. Providers are corporations that want to maximize their profit and minimize their costs and responsibilities.
3. The officials (government, administration, police, ...) ensure that other parties respect the law. In the context of data retention we call them gathering party.

To control the senders some governments have already prescribed data retention. I. e. providers are responsible for collection and storage of information about the communication of the senders. If the officials lawfully demand the retained data for a certain

user, the providers have to disclose them. This approach of data retention rises severe concerns about privacy and massively increasing storage costs of the providers [27].

In the following we present protocols that ensures the privacy of the senders, liberates the providers from storing the retained data items and allows officials to inspect all data items from a predetermined period if a party has recently committed too many critical actions. A critical interaction might occur if a user sends an email to a party that is already a subject of investigation. Depending on the type of interaction we distinguish between critical and non-critical interactions and, thus, between critical and non-critical messages.

The gathering party supplies the provider with a blacklist of critical actions. Whenever a sender interacts with the system the provider classifies this action as critical or non-critical. The provider prepares an encrypted data retention message for this interaction and sends the message to the gathering party.

We will investigate the problem of permitting a gathering party the decryption of recorded messages only if it has received enough critical messages within a predetermined period of time. This means the number of messages for a sender has passed a threshold. We assume that transmissions of critical and non-critical messages occur independently. To simplify the following analysis we assume that in each round there exists at most one message (critical or non-critical) initiated by the same sender.

Basically a message M can be described as a four tuple

$$M = \langle \text{time},\ \text{id},\ \text{share},\ \text{load} \rangle\ .$$

$\text{time}(M)$ denotes the initiating time (the time the message was sent), $\text{id}(M)$ denotes some kind of message ID or sender pseudonym, and $\text{share}(M)$ denotes the shares corresponding to that message. $\text{load}(M)$ consists of further information associated with M, in the following we assume that this part of a message includes the subject $\text{subj}(M)$ of the message. If M is non-critical we choose $\text{id}(M) = \text{share}(M) = 0$.

The sequence of messages belonging to the same sender may reveal information about his (critical) activities if the sender can be identified. We denote the sequence of messages belonging to the same sender as the history of the sender. Given a message M we also call the sequence of messages M' belonging to the sender of M and with $\text{time}(M) > \text{time}(M')$ the history of M.

3.1 Encryption of the Load

We assume there are n different senders. The identity of a sender is a unique code word of a binary blockcode $\mathcal{I}$ with Hamming distance δ and more than n code words. If a sender $\mathcal{I}_i$ commits an activity, the gathering party wants to retain data corresponding to that activity. Depending on the type of the activity, critical or not, the provider prepares a message M, critical or not, including the encrypted retained data (the encrypted subject) in $\text{load}(M)$. Then, the provider sends the message to the gathering party. The gathering party stores all retained data in a pool of messages. If we have obtained

a decryption key from critical messages, we must identify the messages that can be decrypted with this key. Therefore, we introduce an indicator string R, a pseudorandom seed L, and the (encrypted) fingerprint $\mathrm{fp}_{R,L}(\mathcal{I}_i)$ that corresponds to the identity of the sender. The implementation of the fingerprint is based on Naor's commitment scheme [20]. The structure of $\mathrm{load}(M)$ is

$$\mathrm{load}(M) = \langle R,\ \mathrm{fp}_{R,L}(\mathcal{I}_i),\ \mathrm{enc}_K(\mathrm{subj}(M))\rangle\ .$$

Let t be the time step when M is sent. For the random string $S \in \{0,1\}^\ell$ we generate the keys K and $L \in \{0,1\}^\ell$ from $\mathrm{seed}^{(t)}(S)$ by $\mathrm{rand}^{(t)}(S) = K \circ L$. Let $G'\colon \{0,1\}^\ell \to \{0,1\}^{|\mathrm{subj}(M)|}$ be a PRNG. Then, we define the encryption function as follows

$$\mathrm{enc}_K(\mathrm{subj}(M)) = G'(K) \oplus \mathrm{subj}(M)$$

where $\oplus$ is the bitwise XOR. Note that this crypto-system is symmetric with decryption function $\mathrm{dec}_K = \mathrm{enc}_K$. Instead of using XORing we may also use any other symmetric crypto-system like DES or AES.

Let $m \in \mathbb{N}$ be the security parameter for the fingerprint and $G''\colon \{0,1\}^\ell \to \{0,1\}^{m\cdot|\mathcal{I}_i|}$ be a PRNG. Let $G''(L) = B_1 \circ \ldots \circ B_{|\mathcal{I}_i|}$ where $B_j \in \{0,1\}^m$. Next, let $R = R_1 \circ \ldots \circ R_{|\mathcal{I}_i|} \in \{0,1\}^{m\cdot|\mathcal{I}_i|}$ where each block $R_i \in_R \{0,1\}^m$ is a random string containing at least one 1. We call R the *indicator string* of the fingerprint. Then we define

$$\mathrm{fp}_{R,L}(\mathcal{I}_i) = \tilde{B}_1 \circ \ldots \circ \tilde{B}_{|\mathcal{I}_i|}$$

as follows: If the jth bit b_j of $\mathcal{I}_i$ is 0 then we choose $\tilde{B}_j := B_j$. Otherwise, if $b_j = 1$ choose $\tilde{B}_j := B_j \oplus R_j$, i.e. we negate the bits of B_j where the corresponding bits in R_j are 1.

Having access to the message M and the keys K and L, in particular by expanding the correct $\mathrm{seed}^{(t)}(S)$, we can decrypt $\mathcal{I}_i$ and check whether it matches a fingerprint $\mathrm{fp}_{R,L}(\mathcal{I}_i)$. Recall that the values S are chosen independently for each sender. Let K' and L' denote two keys generated by a different seed $\mathrm{seed}^{(t)}(S')$ with $S \neq S'$, then, $\mathrm{fp}_{K,L}(\mathcal{I}_i)$ should be different to $\mathrm{fp}_{K',L'}(\mathcal{I}_j)$ for every different identity $\mathcal{I}_j \in \mathcal{I}$. Note that otherwise we would possibly associate M with the wrong sender. If the unwanted case happens, i.e. we associate the message M to the wrong identity $\mathcal{I}_j$, we say that a collision occurs. We will now analyze the probability of such a collision.

As described before we assume that each sender initiates at most one message per round. Furthermore, we assume that with probability $(1-q)$ all strings K and $G'(K)$, L, and $G''(L)$, respectively, are different for all messages. We can ensure that q is very small by increasing the seed length.

Lemma 6 *If for the security parameter it holds that $m \geq \ell + 1$, then that the collision probability is at most $q + (1-q)\cdot 2^{-(\delta-1)\ell}$.*

Proof: [Lemma 6] Let M be a message that is initiated by $\mathcal{I}_i$ at time step t, and $\mathrm{load}(M)$ is encrypted using $\mathrm{seed}^{(t)}(S)$ and $\mathrm{rand}^{(t)}(S) = K \circ L$. Then, it holds that $\mathrm{load}(M) = \langle R, \mathrm{fp}_{R,L}(\mathcal{I}_i), \mathrm{enc}_K(\mathrm{subj}(M))\rangle$. Recall that with probability q we have a

collision of K or L and we can decrypt subj(M) or obtain $\mathcal{I}_i$ even in the case that the gathering party knows a seed S' different from S.

The remaining part of the proof follows proof for the non-cheating property of the bit commitment protocol in [20]. Now, we assume that the gathering party has access to $K' \neq K$ and $L' \neq L$ obtained from a seed seed$^{(t)}(S')$. Let $G''(L) = B_1 \circ \ldots \circ B_{|\mathcal{I}|}$ and $G''(L') = B'_1 \circ \ldots \circ B'_{|\mathcal{I}|}$. We will estimate the probability of collision in this case, i.e., that M is associated with any (possibly wrong) identity.

Since the code $\mathcal{I}$ has got a Hamming distance δ, for all pairs of code words $\mathcal{I}_i \neq \mathcal{I}_j$ there exist at least δ positions $\kappa = \{k_1, \ldots, k_\delta\}$ where these two code words are different. There only exists a collision if for all $k \in \kappa$ it holds that $\tilde{B}_k = \tilde{B}'_k$ and therefore $B_k = R_k \oplus B'_k$. Note that the bits from B_k and B'_k differs exactly at the positions where R_k is 1 and they have to be identical at all remaining positions. Hence, if we had chosen a different value for R_k this collision had been prevented.

Since any pseudorandom string L is dependent on the original seed S there are at most 2^ℓ different choices for L'. For each R_k with $k \in \kappa$ there are $2^m - 1 > 2^{m-1}$ possible strings. Thus, there are more than $2^{\delta(m-1)}$ different choices for $R_{k_1}, \ldots, R_{k_\delta}$.

Consequently, the probability that there exists a collision is at most $q + (1-q)\frac{2^\ell}{2^{\delta(m-1)}}$. For $m \geq \ell + 1$ it holds that the probability for a collision is at most $q + (1-q)2^{-(\delta-1)\ell}$. ∎

Hence, it is very unlikely that a message M is associated with the wrong sender. Next, we analyze the information about $\mathcal{I}_i$ and subj(M) that can be deduced from load(M).

Lemma 7 *For any probabilistic polynomial algorithm $\mathcal{A}$, all polynomials p and for all sufficiently large ℓ it holds that*

$$\begin{array}{lrcll} 1. & \Pr[\mathcal{A}(\text{load}(M)) = \mathcal{I}_i] & < & \frac{1}{n} + \frac{1}{p(\ell)} & \textit{and} \\ 2. & \Pr[\mathcal{A}(\text{load}(M)) = \text{subj}(M)] & < & \frac{1}{|\mathcal{SU}|} + \frac{1}{p(\ell)} \ . & \end{array}$$

Proof: The first inequality follows analogously to Naor [20]: R and $\text{fp}_{R,L}(\mathcal{I}_i)$ do not reveal more than negligible information about $\mathcal{I}_i$.

From Proposition 2 it follows that K and $G'(F)$ are pseudorandom. Thus, also $G'(K) \oplus \text{subj}(M) = \text{enc}_K(\text{subj}(M))$ is pseudorandom. Otherwise, if we were able to deduce non-negligible information about subj(M) from load(M), then we would be able to construct a polynomial algorithm that distinguishes between pseudorandom and truly random strings. The second inequality follows directly. ∎

If we do not have any information about the decryption keys K and L then the advantage to guess $\mathcal{I}_i$ and subj(M) for a message M is negligible.

3.2 Allocation and Generation of the Keys

We choose the length of discrete time steps such that every sender commits at most one critical activity in every time step. We denote the time steps as natural numbers.

Usually, if the gathering party wants to decrypt the retained data, it may apply for a judicial order. If the petition is justified, the decryption key is disclosed by a trusted party. We propose that critical messages themselves contribute to obtaining the decryption key if a sender commits too many misbehaviors within a specific period of time. Let $\mathcal{T}_i \subset \mathbb{N}$ with $i \in \mathbb{N}$ denote the ith period. Note that several periods may overlap. We assume that $\mathcal{T}_i$ consists of Δ consecutive time steps. Note that with some modification to our schemes we may also allow arbitrary sets of natural numbers for $\mathcal{T}_i$. Let Π_t be the set of all periods i such that $t \in \mathcal{T}_i$ and let $t_{\min}(\mathcal{T}_i) = \min_{t \in \mathcal{T}_i} t$.

To implement the wanted behavior of self decryption we will assign a Shamir shares [24] of the key to each message: For each sender and each period $\mathcal{T}_i$ we generate a random polynomial p of degree $d-1$ over a field $\mathcal{F}$ that is sufficiently large such that $p(0) = \text{seed}^{(t_{\min}(\mathcal{T}_i))}(S)$. If we send a critical message M in time step $t \in \mathcal{T}_i$, then we attach the share $p(t - t_{\min}(\mathcal{T}_i) + 1)$ to M. Note that $1 \leq t - t_{\min}(\mathcal{T}_i) + 1 \leq \Delta$.

Thus, if we receive d messages within $\mathcal{T}_i$, we can reconstruct $\text{seed}^{(t_{\min}(\mathcal{T}_i))}(S)$ by Lagrange interpolation. Afterwards, we can generate the sequence of $\text{seed}^{(t)}(S)$ and $\text{rand}^{(t)}(S)$ for all $t \in \mathcal{T}_i$. According to the used encryption of the load of a message we can identify those (critical and non-critical) messages where we can correctly decrypt the fingerprint $\text{fp}_{R,L}(\mathcal{I}_i)$. For each of these identified messages M we can also decrypt $\text{subj}(M)$.

4 A Threshold Scheme for Critical Data

In this section we present a scheme to construct critical messages.

Scheme Initialization:

- The gathering party supplies the provider with a blacklist of critical activities.
- For each user i the provider performs the following steps: The provider generates an initial seed S and a unique random number u. For each period $\mathcal{T}_j$ we choose a random polynomial p_j with $p_j(0) = \text{seed}^{(t_{\min}(\mathcal{T}_j))}(S)$.

Sending a critical message: Assume that a user i performs a critical activity at round t. In order to identify a critical activity, the provider checks his blacklist. Using the user-specific seed S and random number u the provider generates the message M with $\text{time}(M) = t$, $\text{id}(M) = u$, $\text{share}(M) = \langle p_{j_1}(x_{j_1}), p_{j_2}(x_{j_2}), \ldots, p_{j_\ell}(x_{j_\ell})\rangle$ where $\Pi_t = \{j_1, \ldots, j_\ell\}$ and $x_j = t - t_{\min}(\mathcal{T}_j) + 1$, and $\text{load}(M) = \langle R, \text{fp}_{R,L}(\mathcal{I}_i), \text{enc}_K(\text{subj}(M))\rangle$. Then, the provider sends M to the gathering party.

Identification and Decryption: If the gathering party has received d critical messages with the same id u within $\mathcal{T}_j$ it reconstructs $\text{seed}^{(t_{\min}(\mathcal{T}_j))}(S)$ by Lagrange in-

terpolation. Note that this reconstruction can be done efficiently if the critical messages are sorted according to their $\mathrm{id}(M)$. Afterwards, it can generate all subsequent values $\mathrm{seed}^{(t)}(S)$ and $\mathrm{rand}^{(t)}(S)$. Using these values the gathering party is able to identify all (critical and non-critical) messages where it can correctly decrypt the fingerprint $\mathrm{fp}_{R,L}(\mathcal{I}_i)$. For each of these identified messages M the gathering party decrypts $\mathrm{subj}(M)$.

Let $\mathcal{IU}_{\mathrm{unident}}$ be the set of all identities that the gathering party has not been able to identify, i.e., for any $\mathcal{I}_i \in \mathcal{IU}_{\mathrm{unident}}$ and all periods $\mathcal{T}_j$ the gathering party has not received d critical messages associated with $\mathcal{I}_i$ within period $\mathcal{T}_j$.

Theorem 8 *Let M be a message that is associated with $\mathcal{I}_i \in \mathcal{IU}_{\mathrm{unident}}$. Then, M is anonymous to the gathering party with respect to $\mathcal{IU}_{\mathrm{unident}}$.*

Proof: If $\mathcal{I}_i \in \mathcal{IU}_{\mathrm{unident}}$, then $\mathrm{seed}^{(t)}(S)$ cannot be deduced from $\mathrm{share}(M)$. Furthermore, for all messages M' (even for $M = M'$)the values $\mathrm{time}(M')$, $\mathrm{id}(M')$, and $\mathrm{share}(M')$ are independently chosen from $\mathrm{subj}(M)$ and $\mathrm{seed}^{(t)}(S)$. Thus, we cannot get access to $\mathrm{seed}^{(t)}(S)$ and the keys K and L of M from all received values $\mathrm{time}(M')$, $\mathrm{id}(M')$, and $\mathrm{share}(M')$. Hence, for all probabilistic polynomial algorithms $\mathcal{A}$ there exists a probabilistic polynomial algorithm $\mathcal{B}$ such that for all communication strings c and contents r of the random strings it holds that

$$\begin{aligned} Q &= |\Pr[\mathcal{A}(\mathrm{subj}(M), M, c, r) = 1] - \Pr[\mathcal{A}(I, M, c, r) = 1]| \\ &= |\Pr[\mathcal{B}(\mathrm{subj}(M), \mathrm{load}(M), c, r) = 1] - \Pr[\mathcal{B}(I, \mathrm{load}(M), c, r) = 1]| \ . \end{aligned}$$

Since $\mathcal{I}_i \in \mathcal{IU}_{\mathrm{unident}}$, Lemma 7 implies that $\mathrm{load}(M)$ is pseudorandom. Moreover, c consists of sequence of messages that are independent from M except of M itself. Thus, for all polynomials p and all sufficiently large ℓ it holds that

$$Q = |\Pr[\mathcal{C}(\mathrm{subj}(M), \mathrm{load}(M), r) = 1] - \Pr[\mathcal{C}(I, \mathrm{load}(M), r) = 1]| \leq \frac{1}{p(\ell)}$$

for every probabilistic polynomial algorithm $\mathcal{C}$. Thus, M fulfills the requirement of anonymity. ∎

If we restrict ourselves to use only one period $\mathcal{T}_i = \mathcal{T}_0$, then we can also use the protocols proposed by Jarecki and Shmatikov [16] since they may only encrypt a constant number of messages of a tag (user). If the messages can be decrypted, all messages with the same tag can be decrypted. A PRNG as key generator allows us to encrypt a polynomial number of messages with the same tag and also prevents the decryption of messages with the same tag that were encrypted long ago.

Privacy of the Message History: Now, we are going to analyze the situation where the gathering party has received d or more critical messages with the same id u. Let t' be the earliest time step such that we can recover a seed $\mathrm{seed}^{(t')}(S)$ from these messages. Then, we can decrypt the identity $\mathcal{I}_i$ and all messages from the corresponding sender that are initiated at time step $t \geq t'$. In addition, we are able to identify the complete history of critical messages since all critical message have the same id u. Therefore, we can also determine partial knowledge about the history of the identified sender. Recall

that even for an identified user one of our goals is to ensure the anonymity of messages initiated at steps $t < t'$. We can guarantee this for non-critical messages since we cannot compute $\text{seed}^{(t)}(S)$ for $t < t'$ by construction. But the history of critical messages is still disclosed. In the following we present two approaches that can be used to obscure the history of critical messages.

First approach: Recall that the sharing information share(M) of a message consists of ℓ shares of the ℓ polynomials that are valid for the round t where the critical activity have been performed. To obscure the history the provider may generate individual messages for the shares and sends them in an arbitrary order mixed with the shares of other users. In more detail the scheme looks as follows: For each period $\mathcal{T}_j$ and each individual with identity $\mathcal{I}_i$ we assign a unique random number $u_{i,j}$ to the polynomial p_j. Furthermore, we divide the sharing information share(M) into the parts associated with the different polynomials p_j. For all $\mathcal{T}_j$ with $t \in \mathcal{T}_j$ we send a message $\langle t, u_{i,j}, p_j(x_j), \text{load}(M)_j \rangle$ in place of the original message M where every load $\text{load}(M)_j$ is encrypted with a different key. This strategy obscures the history if the number of messages that occur at each round is huge. If the gathering party receives a small number of messages for some rounds, it will be able to collect information about the relationship of numbers $u_{i,j}$ of the same sender and, thus, about the message history.

Second approach: We allow that every user can use a fixed number α of different ids. Hence, it is possible that a user can choose one of his IDs to be used in a message. If we assume that each id is only valid for a fixed period of time and if the user performs only a small number of critical activities he can hide the history that is associated with a specific id. Hence, in the worst case, the sender might be able to send $(2\alpha + 1)(d - 1)$ critical messages within Δ time steps such that he cannot be identified. However, in most cases it is desirable that a sender of critical messages is identified whenever the threshold d of critical messages is reached.

5 A Protocol for Obscuring the History

In the previous section we have presented a protocol that allows an observer to gain some knowledge on the history of the parties. This knowledge includes the appearance of critical data even if the content of the critical data remains decrypted. If the ID of a party is discovered the observer can assign such an history to the found ID. In the following protocol we will change the way how critical messages are generated. This allows us to mix and thereby obscure the histories of the critical messages. This protocol does not change the generation of non-critical messages.

More precise, we will replace the *polynomial* or *pseudo sender ID* of every critical message by an ambiguous randomly chosen message ID $\text{id}_{\text{act}} \in_R \{1, \ldots, N\}$, i.e. by an ID that may appear for several messages of several senders. To connect consecutive messages of the same sender we will include the message ID $\text{id}_{\text{pre}} \in \{1, \ldots, N\}$ of the preceding message (or a randomly chosen message ID if the actual message is the first message of the sender) in the actual message. Hence, we modify the structure of a

message M as follows:

$$M \;=\; \langle \text{time},\ \text{id}_{\text{act}},\ \text{id}_{\text{pre}},\ \text{share},\ \text{load} \rangle \ .$$

Recall, that we assume that the IDs are chosen randomly and are not unique, i.e. within a certain period of time several messages with the same ID will be used with a non negligible probability.

Let $\mathcal{M}_{[t]}$ denote the set of all messages collected by the gathering party until step t. Then, we can draw a *message graph* $G_{[t]} := (\mathcal{M}_{[t]}, E_{[t]})$ where for $M_1, M_2 \in \mathcal{M}_{[t]}$

$$(M_1, M_2) \in E_{[t]} \quad \Longleftrightarrow \quad \text{time}(M_1) > \text{time}(M_2) \ \text{ and } \ \text{id}_{\text{pre}}(M_1) = \text{id}_{\text{act}}(M_2) \ .$$

Note that a directed path from a source in $G_{[t]}$ to a sink denotes the possible sequence of all messages of a sender. Hence, we have to describe an algorithm that detects a correct sequence if the threshold of critical messages is reached.

Analogously to the identification mechanism, we add the encrypted sender ID to the load of each critical message and we assume that, as in the previous protocols, share(M) gives us a share of the seed of a pseudorandom number generator. Having the desired number of d consecutive critical messages $M_1, \ldots, M_d$ we can compute a corresponding seed and by using this seed we can determine (decrypt) a value for the sender ID of every message id_i on this sequence. If all values id_i are equal, then we assume that these messages were initiated by the sender with ID id_i. Note that if such a sequence is initiated by one sender the ID of this sender will be detected. On the other hand, following our analysis to identify the sender of non-critical data it follows that the probability of a false positive, i.e. that we claim that a sequence is initiated by the wrong sender, is negligible.

Lemma 9 *Let ℓ be the length of S and m be the block length of indicator string. If the security parameter $m \geq \ell + 1$ then probability of a false positive is at most $q + (1-q) \cdot 2^{-(\delta-1)\ell}$ where $(1-q)$ is the probability that all seeds and pseudo-random strings used for encrypting and fingerprinting are different.*

Now, we investigate the efficiency of our algorithm to detect a sequence of consecutive critical messages $M_1, \ldots, M_d$ that are initiated by the same sender within a time period $\mathcal{T}_i$ of length Δ. Let m_t denote the number of messages M with timestamp $t = \text{time}(M)$, let $m_{\max} := \max_t m_t$ and let $m_{\min} := \min_t m_t$. If we assume that at every round every party initiate a critical message with probability p_{cm}, then by some standard calculations one can show that $\Pr[m_t \leq \frac{2}{3} \cdot p_{cm} n] \leq e^{-2p_{cm}n/9}$ and $\Pr[m_t \geq \frac{4}{3} \cdot p_{cm} n] \leq e^{-4p_{cm}n/3}$ where n denotes the number of participating parties. Hence, if we choose N such that $N \in 2^{o(p_{cm}n)}$ then with probability $1 - N^{-z}$ we have $\frac{2}{3}p_{cm}n \leq m_t \leq \frac{4}{3}p_{cm}n$ for every constant z. Hence, we can assume that $m_{\min}$ and $m_{\max}$ only deviate form each other by a factor of 2. In the following we assume that $m_{\max} = N^{\varepsilon}$ for some appropriate chosen values $\varepsilon < 1$.

Lemma 10 *With probability $1 - (e \cdot \Delta \cdot N^{\varepsilon-1})^{-(k-1)\cdot\Delta}$ for every constant $k > 1$ the number of different sequences of d consecutive critical messages ending with message M within a period of length Δ is bounded by $(e \cdot \Delta \cdot N^{\varepsilon-1})^d$.*

Proof: Let us first investigate the number of messages M' that may occur within $h = \Delta - d + 1$ consecutive rounds with a given ID. Recall that we need a sequence of d consecutive critical messages to decrypt the secret that appear within a period of length Δ. Let X_{id} be a discrete random variable describing the number of messages M' with $\text{id}_{\text{act}}(M') = \text{id}$ that occur within h consecutive rounds $t, \ldots, t + h - 1$ for a given t. If $X_{\text{id}} \leq c$ w.h.p. for some moderate values c, then the number of sequences of d consecutive critical messages ending with M within a period of length Δ is bounded by c^d w.h.p.

Our goal is to estimate the probability for $X_{\text{id}} \leq c$ and $c = (1 + \varepsilon_1) \cdot \frac{\Delta \cdot m_{\max}}{N}$ for $\varepsilon_1 > 0$. Recall that $\frac{m_t}{N}$ denotes the expected number of messages with the same ID in round t. Using Chernoff bounds we get

$$\Pr[X_{\text{id}} \geq c] \;=\; \Pr[X_{\text{id}} \geq (1+\varepsilon_1)\frac{\Delta \cdot m_{\max}}{N}] \;\leq\; \left(\frac{e^{-\varepsilon_1}}{(1+\varepsilon_1)^{1+\varepsilon_1}}\right)^{\Delta \cdot m_{\max}/N} .$$

Our goal is to find values for ε_1 such that this probability is smaller than $c^{-k \cdot \Delta}$ for some appropriate chosen values $k \in \mathbb{N}$. If we choose $\varepsilon_1 = e - 1$ then one can show that

$$\Pr[X_{\text{id}} \geq c] \;\leq\; c^{-k\Delta} \quad \Longleftrightarrow \quad m_{\max} \;\geq\; \frac{1 + \ln \Delta - (1-\varepsilon)\ln N}{2e - 1} \cdot kN$$

where $m_{\max} = N^{\varepsilon}$ for $\varepsilon < 1$. Note that the latter holds if $1 + \ln \Delta - (1 - \varepsilon) \ln N < 0$ and thus if $N > e^{\frac{1+\ln \Delta}{1-\varepsilon}}$.

Thus, with probability $1 - c^{-(k-1)\cdot\Delta}$ we have that the number of sequences of d consecutive critical messages ending with message M within a period of length Δ is bounded by

$$c^d = (e \cdot \Delta \cdot N^{\varepsilon - 1})^d \; .$$

∎

If we have $\varepsilon = \frac{1}{2}$, then $m_{\max} = \sqrt{N} \in \omega(\ln N)$ and $c^d = (e \cdot \Delta/\sqrt{N})^d$ where the polynomial degree d is a constant given by the system.

Note that whenever a new message arrives at the gathering party, it has to search in the message graph whether there exists a sequence of d consecutive critical messages in the actual period that ends with the recently received message. Thus, the lemma above gives us a time bound for our algorithm for detecting such a sequence.

Lemma 11 *With high probability our algorithm determines whether there exists a sequences of d consecutive critical messages ending with a given message M within a period of length Δ in time $O((e \cdot \Delta \cdot N^{\varepsilon-1})^d)$.*

Let us now focus on the question whether we can determine the history of critical messages of a sender i of a message M if non of his messages M' with $\text{time}(M) > \text{time}(M')$ can be used to determine the ID of i, i.e. for every period $\mathcal{T}_j$ that ends before time step $\text{time}(M)$ sender i has initiated less than d critical messages.

Let $\tilde{m}_t$ denote the number of messages M with timestamp $t = \text{time}(M)$ that do not belong to a sequence of d consecutive critical messages within a period of length Δ. Let

$\tilde{\mathcal{M}}_{[t]}$ denote the corresponding set of all of these messages with timestamp $t \leq \text{time}(M)$ and let $\tilde{G}_{[t]}$ denotes the subgraph of $G_{[t]}$ induced by the set $\tilde{\mathcal{M}}_{[t]}$. Define $\tilde{m}_{\min} := \min_t \tilde{m}_t$.

Note that, if for a message $M \in \tilde{\mathcal{M}}_{[t]}$ there exists two or more messages $M_1, M_2, \ldots \in \tilde{\mathcal{M}}_{[t]}$ with $t = \text{time}(M)$ and $\text{id}_{\text{pre}}(M) = \text{id}_{\text{act}}(M_1) = \text{id}_{\text{act}}(M_2) = \ldots$, then the predecessor of the message M within a message history of M cannot be uniquely determined. With high probability there exist at least two of these messages M_i, M_j with $\text{id}_{\text{pre}}(M_i) \neq \text{id}_{\text{pre}}(M_j)$. Hence, one can see that the message history of M gets more and more diffused. In the following we investigate the degree of diffusion as a function over the time. Let

$$I(t', M) := \{\text{id}_{\text{pre}}(M') \mid M' \in \tilde{\mathcal{M}}_{[t]} \text{ with } \text{time}(M') \geq t' \text{ and } M' \text{ is reachable from } M \text{ in } \tilde{G}_{[t]} \text{ where the time distance of any two consecutive messages on at least one path from } M \text{ to } M' \text{ is at least } \Delta/(d-1)\} .$$

The set $I(t', M)$ denotes a subset of IDs of messages that belong to potential predecessors M in the message history. In the following we analyze the cardinality of $I(t', M)$. Our goal is to prove an upper bound for t' such that with high probability we have $|I(t', M)| = N$. If this is true, then we can assume that all messages that are initiated before time t' are potential members of the message history of M. Note that

$$I(\text{time}(M), M) \;=\; \{\, \text{id}_{\text{pre}}(M) \,\} \quad \text{and} \quad \forall t' > \text{time}(M) \;:\; I(t', M) \;=\; \emptyset \; .$$

Furthermore, for all $t \leq \text{time}(M) - \Delta/(d-1)$ it holds that $\text{id} \in I(t, M) \setminus I(t+1, M)$ iff there exists at least one message $M' \in \tilde{\mathcal{M}}_{[t]}$ such that $\text{id} = \text{id}_{\text{pre}}(M')$ and $\text{id}_{\text{act}}(M') \in I(t + \Delta/(d-1), M)$.

Lemma 12 *Let*

$$\zeta(k) \;:=\; k \cdot \frac{N^2 \log_2 N}{N - \tilde{m}_{\min}} + \frac{\Delta}{d-1}$$

For any M, $k > 1$, and $t \leq \text{time}(M) - \zeta(k)$, it holds that $|I(t, M)| > \tilde{m}_{\min}$ with probability $1 - N^{-(k-1)}$.

Proof: Let $c := \Delta/(d-1)$. For $\ell \in \mathbb{N}$ we investigate the rounds $r \in \{\text{time}(M) - c, \ldots, \text{time}(M) - c - \ell\}$. Let $M_{r,1}, \ldots, M_{r,\tilde{m}_{\min}}$ denote a sequence of $\tilde{m}_{\min}$ messages with $\text{time}(M_{r,i}) = r$. Finally let $M_{\bullet,i} := M_{r,i}|_{r \in \{\text{time}(M)-c,\ldots,\text{time}(M)-c-\ell\}}$ denote the sequence of the i-th messages of these sequences. Our goal is to determine ℓ such that with high probability we can find in each sequence $M_{\bullet,i}$ one message M_i such that all of these messages have the same ID $\text{id}_{\text{act}}(M_i) = \text{id}_{\text{pre}}(M)$ and they have different values $\text{id}_{\text{pre}}(M_i)$ with $\text{id}_{\text{pre}}(M_i) \neq \text{id}_{\text{pre}}(M)$.

Note that the probability that we have search the first j_1 elements of $M_{\bullet,1}$ before we find an adequate message M_1 with $\text{id}_{\text{act}}(M_i) = \text{id}_{\text{pre}}(M)$ and $\text{id}_{\text{pre}}(M_i) \neq \text{id}_{\text{pre}}(M)$ is

$$\frac{N-1}{N^2} \cdot \left(1 - \frac{N-1}{N^2}\right)^{j_1 - 1} \quad .$$

We analyze the sequences $M_{\bullet,i}$ one after another. Assume that we have already determined $M_1, \ldots, M_{i-1}$. Let J_i denote the random variable for the event that we have search the first J_i elements of $M_{\bullet,i}$ before we find an adequate message M_i with $\mathrm{id}_{\mathrm{act}}(M_i) = \mathrm{id}_{\mathrm{pre}}(M)$, $\mathrm{id}_{\mathrm{pre}}(M_i) \neq \mathrm{id}_{\mathrm{pre}}(M)$ and $\mathrm{id}_{\mathrm{pre}}(M_i) \notin \{\mathrm{id}_{\mathrm{pre}}(M_1), \ldots, \mathrm{id}_{\mathrm{pre}}(M_{i-1})\}$ then for every $j_i \in \mathbb{N}$ we have

$$\Pr[J_i = j_i] \;=\; \frac{N-i}{N^2} \cdot \left(1 - \frac{N-i}{N^2}\right)^{j_i - 1} \qquad \text{and} \qquad \Pr[J_i > j_i] \;=\; \left(\frac{i}{N^2} + \frac{N-1}{N}\right)^{j_i} \;.$$

Note that for every value $j \in \mathbb{N}$ the probabilities for $\Pr[J_i > j]$ increase with i. Hence, it suffices to find a value for $j_{\tilde{m}_{\min}}$ such that $\Pr[J_{\tilde{m}_{\min}} > j_{\tilde{m}_{\min}}]$ is sufficiently small. Note that

$$\Pr[J_i > j_i] \;=\; \left(1 - \frac{N-i}{N^2}\right)^{j_i} \;\leq\; 2^{-\frac{N-i}{N^2} j_i} \;.$$

Hence, if we choose $\ell = k \cdot \frac{N^2 \cdot \log_2 N}{N - \tilde{m}_{\min}}$ for any i, then we get $\Pr[J_i > \ell] \leq N^{-k}$. Hence with probability $1 - \frac{N}{N^k}$ we have added at least $\tilde{m}_{\min}$ IDs to $I(\mathrm{time}(M), M)$ in $\ell + \frac{\Delta}{d-1} = \zeta(k)$ rounds. ∎

We generalize the this lemma in the following. Assume that $N > m_{\min}$.

Lemma 13 *Let*

$$\xi(k) := k \cdot \frac{N^2 \cdot \log_2 N}{mN - m^2 - \tilde{m}_{\min} m + m} + \frac{\Delta}{d-1} \;.$$

For any M, $k > 1$, $s, m \in \mathbb{N}$, and $t \leq s - \xi(k)$, it holds that if $|I(s,M)| \geq m$ and $m + m_{\min} \leq N$ then $|I(t,M)| \geq m + \tilde{m}_{\min}$ with probability $1 - N^{-(k-1)}$.

Proof: Let $c := \Delta/(d-1)$ and let us assume that $m = |I(s,M)|$. For $\ell \in \mathbb{N}$ we investigate the rounds $r \in \{s-c, \ldots, s-c-\ell\}$. Let $M_{r,1}, \ldots, M_{r,\tilde{m}_{\min}}$ denote a sequence of $\tilde{m}_{\min}$ messages with $\mathrm{time}(M_{r,i}) = r$. Finally let $M_{\bullet,i} := M_{r,i}|_{r \in \{s-c,\ldots,s-c-\ell\}}$ denote the sequence of the i-th messages of these sequences. Our goal is to determine ℓ such that with high probability we can find in each sequence $M_{\bullet,i}$ one message M_i such for all of these messages M_i we have $\mathrm{id}_{\mathrm{act}}(M_i) \in I(s,M)$ and all of the predecessor IDs $\mathrm{id}_{\mathrm{pre}}(M_i)$ are different and it holds that $\mathrm{id}_{\mathrm{pre}}(M_i) \notin I(s,M)$.

Note that the probability that we have search the first j_1 elements of $M_{\bullet,1}$ before we find an adequate message M_1 with $\mathrm{id}_{\mathrm{act}}(M_i) \in I(s,M)$ and $\mathrm{id}_{\mathrm{pre}}(M_i) \notin I(s,M)$ is

$$\frac{mN - m^2}{N^2} \cdot \left(1 - \frac{mN - m^2}{N^2}\right)^{j_1 - 1} \;.$$

We analyze the sequences $M_{\bullet,i}$ one after another. Assume that we have already determined $M_1, \ldots, M_{i-1}$. Let J_i denote the random variable for the event that we have search the first J_i elements of $M_{\bullet,i}$ before we find an adequate message M_i with $\mathrm{id}_{\mathrm{act}}(M_i) \in I(s,M)$, $\mathrm{id}_{\mathrm{pre}}(M_i) \notin I(s,M)$, and $\mathrm{id}_{\mathrm{pre}}(M_i) \notin \{\mathrm{id}_{\mathrm{pre}}(M_1), \ldots, \mathrm{id}_{\mathrm{pre}}(M_{i-1})\}$ then for every $j_i \in \mathbb{N}$ we have

$$\Pr[J_i = j_i] \;=\; \frac{m(N-i-m+1)}{N^2} \cdot \left(1 - \frac{m(N-i-m+1)}{N^2}\right)^{j_i - 1}$$

and

$$\Pr[J_i > j_i] \;=\; \left(\frac{m(m+i-1)}{N^2} + \frac{N-m}{N}\right)^{j_i} \;.$$

For every value $j \in \mathbb{N}$ the probabilities for $\Pr[J_i > j]$ increases with i. Hence, it suffice to find a value for $j_{\tilde{m}_{\min}}$ such that $\Pr[J_{\tilde{m}_{\min}} > j_{\tilde{m}_{\min}}]$ is sufficiently small. Note that

$$\Pr[J_i > j_i] \;=\; \left(1 - \frac{mN - m^2 - mi + m}{N^2}\right)^{j_i} \;\leq\; 2^{-\frac{mN-m^2-mi+m}{N^2} j_i} \;.$$

Hence, if we choose $\ell = k \cdot \frac{N^2 \cdot \log_2 N}{mN - m^2 - \tilde{m}_{\min} m + m}$ for any i, then we get $\Pr[J_i > \ell] \leq N^{-k}$. Hence, with probability $1 - \frac{N}{N^k}$ we have added at least $\tilde{m}_{\min}$ IDs to $I(s, M)$ in $\ell + \frac{\Delta}{d-1} = \xi(k)$ rounds. ∎

From the lemma above we can conclude that the size of $I(t, M)$ increases by $\tilde{m}_{\min}$ with high probability in every $\xi(k)$ rounds. Hence, we need $\mathcal{O}(\frac{N}{\tilde{m}_{\min}})$ iterations of this strategy before we reach a set $I(t, M)$ that includes a constant fraction of all IDs $\{1, \ldots, N\}$. Note that $\tilde{m}_{\min}$ might be small. The time needed by such an iteration depends on the size m of the corresponding set $I(s, M)$, i.e. the required time is $\xi(k)$.

Note that m has to be from the range $m \in \{1, \ldots, N - m_{\min}\}$ and the bound for $|I(t, M)|$ has its maximum value for $m = 1$ and $m = N - \tilde{m}_{\min}$. For these values of m the required time is

$$t \;:=\; \zeta(k) \;=\; k \cdot \frac{N^2 \cdot \log_2 N}{N - \tilde{m}_{\min}} + \frac{\Delta}{d-1} \;.$$

The following two lemmata follow directly from our observations above.

Lemma 14 *For any M, $k > 2$, and*

$$t \leq \text{time}(M) - \frac{N}{\tilde{m}_{\min}} \cdot \zeta(k)$$

it holds that

$$|I(t, M)| = N \text{ with probability } 1 - N^{-(k-2)} \;.$$

Lemma 15 *For any sufficiently large time step T, $k > 3$, and*

$$t \leq T - \frac{N}{\tilde{m}_{\min}} \cdot \zeta(k)$$

it holds simultaneously for every message M with $T \leq \text{time}(M)$ that

$$|I(t, M)| = N \text{ with probability } 1 - N^{-(k-3)} \;.$$

We can conclude:

Theorem 16 *For every $k > 3$ and every message M with probability $1 - N^{-(k-3)}$ we cannot deduce any information about the history of M if we investigate messages that are initiated in a round*

$$t \leq \text{time}(M) - \frac{N}{\tilde{m}_{\min}} \cdot \zeta(k) \;.$$

If many senders have been identified, then the value of $\tilde{m}_{\min}$ is small. The number of iterations needed to reach $|I(t, M)| = N$ increases. Thus, it is desirable to have a value $\tilde{m}_{\min}$ that is close to $m_{\min}$. If we combine several rounds in our analysis, then we get a higher value for $\tilde{m}_{\min}$ and slightly better bounds. Asymptotically, the time needed until $|I(t, M)| = N$ is in $\mathcal{O}(N^2 \log(N)/\min\{\tilde{m}_{\min}, N - \tilde{m}_{\min}\})$. Hence, if $\tilde{m}_{\min} = \varepsilon \cdot N$ for $\varepsilon < 1$, then the required time is in $\mathcal{O}(N \log(N))$.

6 Conclusions

In this paper we presented a scheme for data retention that allows self-decryption if the number of critical messages reaches a threshold. As long as the messages cannot be decrypted the sender is anonymous to the gathering party. Furthermore, we introduced the history of messages as subject of privacy. Our scheme ensures the privacy of the history of non-critical messages. For critical messages we propose a protocol that obscures the history. The runtime of this protocol is polynomial in the parameters $N^{\varepsilon-1}$ and Δ but exponential in d. Since most nodes in $\mathcal{M}_{[t]}$ have degree at least 2, there is little hope to reduce the exponential behavior in d. But it is left open whether we can improve the runtime for the other parameters. In the protocols presented in this paper all messages of a user are encrypted by the same provider. An interesting question is whether we can extend our protocols such that one user can user several different providers that do not share information about their customers. Such a multi-provider protocol requires that the users have to encrypt their messages themselves and that the providers have to verify the correctness of the used encryption keys.

References

[1] A. Agarwal, H. Li, and K. Roy. Drg-cache: a data retention gated-ground cache for low power. In *DAC*, pages 473–478. ACM, 2002.

[2] J. C. Benaloh and J. Leichter. Generalized secret sharing and monotone functions. In *Proc. of the 8th Annual IACR Cryptology Conference (CRYPTO)*, volume 403 of *LNCS*, pages 27–35, 1988.

[3] G. Blakley. Safeguarding cryptographic keys. In *Proc. of the 1979 AFIPS*, 1979.

[4] J.-F. Blanchette and D. G. Johnson. Data retention and the panoptic society: The social benefits of forgetfulness.“. *The Information Society*, 18:33–45, 2002.

[5] M. Blum and S. Micali. How to generate cryptographically strong sequences of pseudo random bits. In *FOCS*, pages 112–117, 1982.

[6] D. Boneh and M. Naor. Timed commitments. In *Proc. of the 20th Annual IACR Crypto Conference (CRYPTO 2000)*, volume 1880 of *LNCS*, pages 236–254. Springer, 2000.

[7] D. Chaum. Untraceable electronic mail, return addresses, and digital pseudonyms. *Commun. ACM*, 24(2):84–88, 1981.

[8] D. Chaum. Security without identification: Transaction systems to make big brother obsolete. *Commun. ACM*, 28(10):1030–1044, 1985.

[9] L. Csirmaz. The size of a share must be large. *J. Cryptology*, 10(4):223–231, 1997.

[10] European Parliament and Council. Directive 2006/24/EC, Mar. 2006.

[11] O. Goldreich. Texts in computational complexity: Pseudorandom generators, Jan. 2006.

[12] O. Goldreich, S. Goldwasser, and S. Micali. How to construct random functions. *J. ACM*, 33(4):792–807, 1986.

[13] S. Haber and W. S. Stornetta. How to time-stamp a digital document. *J. Cryptology*, 3(2):99–111, 1991.

[14] J. Håstad, R. Impagliazzo, L. A. Levin, and M. Luby. A pseudorandom generator from any one-way function. *SIAM J. Comput.*, 28(4):1364–1396, 1999.

[15] M. Ito, A. Saito, and T. Nishizeki. Secret sharing scheme realizing general access structure. In *Proc. of the IEEE Global Telecommunication Conf., Globecom 87*, pages 99–102, 1987.

[16] S. Jarecki and V. Shmatikov. Handcuffing big brother: an abuse-resilient transaction escrow scheme. In *Proc. of the International Conference on the Theory and Applications of Cryptographic Techniques (EUROCRYPT)*, volume 3027 of *LNCS*, pages 590–608, 2004.

[17] H. Krawczyk. Distributed fingerprints and secure information dispersal. In *PODC*, pages 207–218, 1993.

[18] S. Landau. Security, liberty, and electronic communications. In *Proc. of the 24th Annual IACR Crypto Conference (CRYPTO)*, volume 3152 of *LNCS*, pages 355–372, 2004.

[19] G. T. Marx. An ethics for the new surveillance. *Inf. Soc.*, 14(3), 1998.

[20] M. Naor. Bit commitment using pseudorandomness. *J. Cryptology*, 4(2):151–158, 1991.

[21] K. Ng and H. Liu. Customer retention via data mining. *Artif. Intell. Rev*, 14(6):569–590, 2000.

[22] M. O. Rabin. Efficient dispersal of information for security, load balancing, and fault tolerance. *J. ACM*, 36(2):335–348, 1989.

[23] R. L. Rivest, A. Shamir, and D. A. Wagner. Time-lock puzzles and timed-release crypto. Technical report, Cambridge, MA, USA, 1996.

[24] A. Shamir. How to share a secret. *Commun. ACM*, 22(11):612–613, 1979.

[25] W. van Wanrooij and A. Pras. Data on retention. In *DSOM*, volume 3775 of *LNCS*, pages 60–71. Springer, 2005.

[26] A. C.-C. Yao. Theory and applications of trapdoor functions (extended abstract). In *FOCS*, pages 80–91, 1982.

[27] A. Zuccato and K. Rannenberg. Data retention has serious consequences. CEPIS Position Paper, LSI SIN (04)01, 2004.