# Toward an Extensible Framework for Redaction

Soteris Demetriou*, Nathaniel D. Kaufman*, Jonah Baim*, Adam J. Goldsher* and Carl A. Gunter*
*University of Illinois at Urbana-Champaign
{sdemetr2, nkaufma2, baim2, goldshe2, cgunter}@illinois.edu

***Abstract*—**
**Data is being created at an increasing rate by sources like the IoT devices, social media, and camera monitors. This data frequently includes sensitive information that parties must redact to adhere to laws and user privacy policies. At the same time, there is steady progress on *recognizers* that find latent information within rich data streams, and thereby create fresh privacy risks. In this work, we advocate the idea of developing a modular, extensible toolkit based on *decognizers* which are information hiding functions derived from recognizers that redact sensitive information. We offer steps towards an abstract conceptual framework and compositional techniques and discuss requirements for such a toolkit.**

## I. Introduction

With the advent of IoT, digital information is being generated and collected at an unprecedented pace. These rich streams of data entail challenges for dissemination that respects the privacy of individuals. For example, in social media there is a threat of privacy leakage caused by uploaded images because individuals' faces within the image are automatically recognized and shared as the image is disseminated across friends, and friends of friends [1]. Similarly, government data—collected through cameras in police vehicle dashboards (dashcams), and, increasingly, body cameras (bodycams) mounted on the uniforms of police officers—is subjected to public disclosure requests as indicated in the Freedom of Information Act (FOIA).

Controlling access to such rich media streams becomes challenging because they carry incidental information which can be hard to predict *a priori*. As the technology evolves, *latent information* becomes identifiable by an adversary within primitive objects. For example, face recognition technology allows one to infer the presence of targeted people in an otherwise innocuous image or video, while speech recognition advances, allow one to fingerprint a person from an audio stream. A good example of latent information concerns the genome of James Watson, who asked that the value of his ApoE gene be redacted because of its connection to dementia. Subsequently, research in genomics advanced to the point that this information could be inferred from the values of other (exposed) genes [2]. Previous works rely on empirically combining recognition technologies based on current knowledge and expectations. However, some might lead to significant information leakage [3], while others are specific to a media stream [4], [5], [6], [7], [1]. Other works focused on designing frameworks for controlling access to such data by third-party apps [8], [9]. While these might offer practical solutions on their application domains, we observe a lack of theoretical foundations upon which we can confidently built and combine redaction technology in a highly evolving IoT space.

To keep pace with rapid recognition advancements, alternatively one could choose a *close follower* approach: observe when a leakage incidence happens and then try to prevent future instantiations of it. For example, Google Street View added a licence plate redaction technology to its implementation after a reported incidence revealed that licence plates of vehicles can be exposed. In this work, we make a proposal and the first step towards such a *close follower* solution. This will provide the theory and tools to construct redaction functions—which we call decognizers—directly stemming from newly introduced recognizer technology. Towards this end, we develop an abstract conceptual framework to formally describe such transformations from recognizers to decognizers. We further define basic techniques to combine recognizers in ways which allow the construction of correct decognizers (Section II). We envision this to be translated in practice in the form of a modular, extensible, open-sourced toolkit of functions to recognize and redact sensitive data in rich media systems. We illustrate this with a prototype implementation of the toolkit which brings forth interesting redaction scenarios (Section III). Finally, we highlight key issues of both the theoretical framework and the toolkit (Section IV).

## II. Concept

Let us consider a simple conceptual model of a modular and extensible toolkit for managing the redaction of sensitive information from diverse media types. The key unit of information is a *record*. We denote records with $r$. Records can be of many types, including documents, images, audio recordings, videos, tables, and so on. To build the conceptual model we view records at two levels. At a *concrete* level they have common representations on computers like PDF or MPEG files; at an *abstract* level they can be represented as a matrix of values $V$ together with a distinguished element $\perp$. We write $V_{\perp}^{mn}$ for the $m$ by $n$ matrix space whose entries are values $v \in V$ or $\perp$. In this case, an abstract document might be a two dimensional array $V_{\perp}^{mn}$ where $V$ is a space of characters, $m$ is the number of lines and $n$ is the width of the text column. An image is a similar array but $V$ is a space of RGB triples. A video might be a three dimensional array consisting of a collection of images. Let us refer to spaces like $V_{\perp}^{mn}$ as *matrix spaces* and denote them with the character $M$ (so that records $r$ are from $M$). We assume a *concretion* function $C$ that maps an abstract representation (matrix) to a corresponding concrete representation (ASCII document, WAV recording, *etc.*). To keep things simple let's assume that this can be done so that there is an inverse *abstraction function* $A$ so that $A \circ C$ and $C \circ A$ are identity functions.

Now, we suppose that sensitive parts of a collection of records can be found with a function. A *recognizer* is a function $\Phi : M \to \mathcal{P}$ where $\mathcal{P}$ is the set of sets of indices $p$ in the matrix space $M$. For example, if $M$ is $V_\perp^{mn}$, then an element $P \in \mathcal{P}$ is a set of pairs $(i, j)$ where $1 \leq i \leq m$ and $1 \leq j \leq n$. For example, a function on ASCII characters in a document might be $\Phi : V_\perp^{mn} \to \mathcal{P}$ where the output of $\Phi$ on an input document is the set of digits considered to be parts of social security numbers (SSNs). The goal is to use this to redact the SSNs by replacing the recognized digits with the distinguished value $\perp$. We use a special term for the corresponding function that replaces recognized matrix values with $\perp$. This is called a *decognizer* and has the type $\Psi : M \to M$. The *simple* decognizer induced by a recognizer $\Psi$ is defined as follows: $\Psi(r)_p$ is equal to $\perp$ if $p \in \Phi(r)$ and it is equal to $r_p$ otherwise. We'll discuss later the idea of (non-simple) decognizers that produce values other than $\perp$; an example is a video redaction scheme that hides faces with blurring or pixelation.

The next steps to building a conceptual toolkit for managing redaction is to develop a library of recognizers and decognizers and with ways to compose and review them. The composition of decognizers needs to be carefully conducted. For example, if $\Phi$ is a recognizer for SSNs and $\Phi'$ is a recognizer for credit card numbers, one would reasonably expect nice properties, like an assurance that the order in which the induced decognizers are used does not make a difference. To assure the right results we propose a multiary merge function $\nu$ that takes a sequence of recognizers as arguments and produces a recognizer that is independent of the order of its arguments. We define $\nu(\Phi, \Phi')(r) = \Phi(r) \cup \Phi'(r)$. Then we define $\Psi = \mu(\Phi, \Phi')$ to be the decognizer induced by $\nu(\Phi, \Phi')$. Similar definitions are used for lists of arguments $\mu(\Phi_1, \ldots, \Phi_n)$.

Putting all of this together, if we have an ASCII file $r$ and want to use recognizers $\Phi$ and $\Phi'$ to redact SSNs and credit card numbers from it, then the value $r' = C(\mu(\Phi, \Phi')(A(r)))$ has the desired property. In practice it will not be efficient to literally convert concrete into abstract values to apply an abstract recognizer. However, we know that a proposed concrete decognizer $\Psi$ is correct if $\Psi = C \circ \mu(\Phi, \Phi') \circ A$.

## III. Illustration

To explore the idea of a modular and extensible toolkit based on our conceptual model, we built a prototype case study, which explores, within a common implementation framework, a collection of recognizers and decognizers for a diversity of media types including audio, text, and video (Figure 1).

**Audio.** A FOIA request might dictate the disclosure of an audio stream from a conversation between a police officer and a civilian. However, it might be desirable to redact *names* of minors/victims or *phone numbers*. To illustrate this we

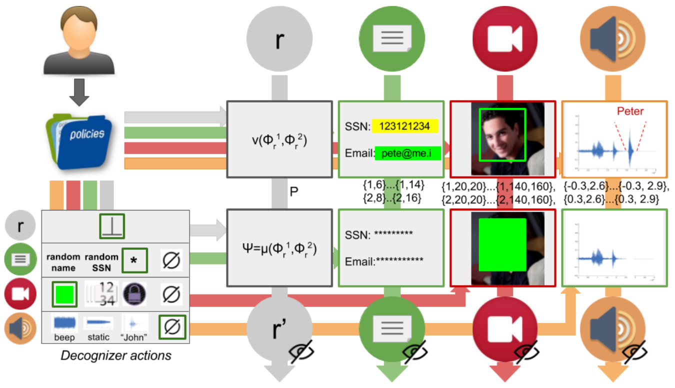

Fig. 1: Transparency high-level architecture for an abstract record *r* and examples for document, video and audio streams. The selected decognizer actions illustrate simple decognizers.

built two respective audio recognizers. We further built a decognizer which, given the original input and the output of the recognizers, replaces the sensitive information with a pre-defined sound (empty sound or beep sound). The recognizers use Google's speech recognition to transcribe the audio into half second intervals which they process to detect the presence of a target name or phone number using a pre-built library. Information regarding the intervals is fed to the decognizer which replaces the sensitive time intervals of the original audio stream with the pre-defined sound. In particular, our framework composes (merges) the recognizers to assure that the order of their application does not matter.

**Text.** Another interesting scenario is redaction of sensitive information from text. In fact the majority of FOIA requests currently involve documents, where sensitive information such as email addresses, phone numbers and SSN numbers need to be redacted. To illustrate this scenario, we built three text recognizers: one for email addresses; one for phone numbers; and one for SSN numbers. We further built an induced decognizer which replaces the characters indicated by the recognizers with a special symbol (like an X or *). Note that, if we apply the recognizers serially, this might lead to privacy leakage. Just for purposes of illustration, consider the following scenario: a phone number (10 digits long) contains an SSN number (9 digits long). If we apply the SSN recognizer first which will redact the 9 digits, and then apply the phone recognizer on the result, the latter will fail, resulting in leaking 1 digit of the phone number. According to our analysis (see Section II), the combination of such recognizers needs to provide an assurance that the order of their application is insignificant. Instead, we apply each recognizer on the original input. Then, for each recognizer output, the decognizer is applied to replace the indicated characters, with a special symbol. Finally, we merge the results (either for each line or for each document as a whole) by maximizing the number of special symbols in the final output text. The toolkit further outputs a report, including whether characters where recognized as part of an object, by more than one recognizer. For example, in the case of the SSN number embedded in a phone number, the report will indicate which characters of the original text were identified as part of both an SSN and a phone number.

**Video.** Video can be modeled as a sequence of frames. Vision recognizers typically output the pixel coordinates where the objects are detected within a frame. We built three video recognizers: one for faces; one for eye detection; and one for mouth detection. In our implementation we used the opencv library with haar cascades. The recognizers identify the corresponding pixels in the original frame detected as part of a desired sensitive object. We also built an induced decognizer, which given the original input and the output of the combination of the recognizers, it replaces the detected pixels with green pixels (rgb(0,255,0)). The induced decognizer simply maximizes the redaction across recognizers. Thus, it wouldn't matter in which order the recognizers were applied; all pixels corresponding to a recognized mouth, eye or face will be replaced. However, in some cases redaction with simple decognizers can fail (leaking information). For example, if within a frame, part of the face is obstructed by a physical object, then the face recognizer would fail. The toolkit will still redact the mouth and eyes of the person but part of the face will be revealed in some frames. In our implementation, we remember the detected and redacted faces from previous frames. Thus when a mouth or eye is detected within the pixel coordinates of an old face, it automatically redacts the whole historical face region to ensure no information leakage. Video demonstrations can be found on our project's website [10].

While this is preliminary, since more complicated scenarios exist in reality, it demonstrates that while the *commutative* property of the $\nu$ function can ensure non information leakage in the simple scenarios we described in Section II, it might not be enough for more complex cases. Therefore, new properties need to be defined to describe more complex recognizer and decognizer functions, which in turn will allow the development of correct redaction application for more interesting scenarios.

## IV. Discussion

**Complex Recognizers.** In Section II we described "simple" abstract decognizers, which replace a matrix value in the original input as indicated by a recognizer, with a distinguished value. This is meaningful and useful in many applications. For example, on a par with our illustration (see Section III), a redaction service could replace all characters belonging to an SSN number with an asterisk (*); in social media, if we were to perform access control on faces rather than whole images we could replace pixels recognized as being in the protected face with black pixels. However, in some cases, the replacement value is not simple. For example, in the audio case, we replaced time intervals with a given sound (empty or beep). Representing audio in the abstract space is more complicated: audio can be seen as a signal in the time domain or the frequency domain. In the former case we could use the root mean squared (RMS) amplitude value per time sample, while, in the former case, we could use the amplitude and frequency values. This would allow us to perceive audio signals as two-dimensional arrays. The recognizers must recognize the time intervals or frequencies belonging to sensitive sound. The replacement value used by the decognizer can be a constant amplitude value to represent the beep sound or empty sound.

However, for some other applications, we want to go beyond the simple decognizer strategy and the issue is not just representation. A well-known example is hiding faces by *blurring* them. Using a simple *box blur* approach, pixels recognized as part of a sensitive face are replaced with the average value of their neighboring pixels in the original matrix. Obviously, our simple decognizer cannot describe this operation. Things become even more challenging with other redaction operations. For example, one might wish to *encrypt* the recognized values [11]. This would allow redacted faces to be replaced later using a key even without access to the original image. Moreover, previous work has shown that by *substituting* words with synonyms or by replacing characters within words, renders automatic identification of the original input more challenging [12]. A number of works also focus on strategies in *de-identifying* health information [13]. In general, the simple decognizer induced by a recognizer can describe useful scenarios, but this is just the tip of the iceberg. We need an extended formalization of the redaction algebra to describe more complicated functions that create decognizers from recognizers. Ideally such an algebra will support proofs of useful properties about potential information leakage.

**Recognizer Quality.** Non-information leakage guarantees are directly correlated with recognizers' detection performance. An *ideal* recognizer would be one that never misses a sensitive object and also all the objects that it detects are sensitive objects. However, most of the recognition technology (speech recognition, human/face detectors) are not ideal and use probabilistic models to make estimates. In a toolkit multiple recognizers that perform a similar function could be submitted by different developers and a user could find support for selecting the most appropriate one. We propose three different schemes for recognizer quality evaluation as follows: (a) manual; (b) assisting; (c) crowdsourced.

In the *manual* scheme, the user runs all candidate recognizers and decognizers on the input records and then manually evaluates their accuracy and selects how to apply them. This is feasible in applications or requests where the input size is tractable and the cost of an error is high. On the other hand, it can guarantee the least information leakage since the user explicitly selects the best redactions. Moreover, through this process, the user has the opportunity to manually redact information missed by all candidate recognizers. Thus it makes the scheme appropriate for sensitive applications like the declassification of documents for review by congressional committees. Many FOIA requests might have this standard as well. Note however, that here the amount of work the user needs to do may be significantly reduced since the framework will automatically find and redact many sensitive instances.

In the *assisting* scheme, the toolkit automates more of the previous process to further alleviate the user from laborious manual evaluations. For example, the tool can provide the administrator with a report and/or cues focusing on the differences between the candidate recognizers. This reduces the administrator effort but it might end up revealing more information than intended. For example, all candidate video recognizers might end up missing the same faces in the same frames. The user might might never be given the opportunity to catch and rectify this event.

Alternatively, the framework could utilize *crowdsourcing* for recognizer evaluation. For example, crowdsourcing platforms such as Amazon Mechanical Turk, Microworkers or similar platforms, can be utilized to enroll users. These will be queried to manually evaluate the accuracy of recognizers on predefined inputs. Of course, some care is needed to assure that sensitive information is not leaked to the crowdsource platform participants. This might be done by taking data out of context such as identifying fragments of names or pictures; or it could be done by labeling non-sensitive data and using this to train classifiers that are used on the sensitive data.

**Toolkit Utility.** Currently, an institution interested in redaction would need to either develop recognizer and redaction algorithms from scratch, or perform a wide scale search for suitable technologies. A toolkit could reduce this effort through a global, open-source repository offering a collection of libraries for recognizer and decognizer algorithms.

A redaction toolkit should support *policies*: that is, given a policy described in a suitable formalism, the correct combination of recognizers and decognizers should be chosen to be applied on the input records. Consider for example the following simple FOIA policy: *replace all SSN numbers from all input documents with character* *. In this case, the toolkit will present the user with all candidate text recognizers that detect SSN numbers and their induced decognizers which replace the characters belonging to SSN numbers with a star symbol. A Facebook policy applied when a person is not authorized to see a particular face in an image could be: *replace face having id='123' with rectangle having color='green'*. This would replace all the pixels belonging to the particular face with green pixels.

**Community Value.** Last but not least, such a toolkit could offer significant value to the community. We envision this to be analogous to the Weka [14] toolkit for data mining. The envisioned toolkit can be the equivalent for redaction where recognizers and decognizers can be integrated and extended by the community. The toolkit can offer support in combining such recognizers by enforcing correctness checks on their input and output arguments. Furthermore, it would be of value to researchers active in information redaction, dataset anonymity, and also to government and private institutions which are either legally bound to perform redactions (FOIA) or offer a relevant service (Google Street View, Facebook etc.).

## V. Related Work

There is a body of literature on solutions for controlled disclosure of specific media types such as images [7], [15], [11], [1], text [3], [16] and video streams [6], [5], [4], [17]. Other works focus on designing frameworks were third-party applications gain controlled access to all or parts of various media objects [8], [9]. All prior techniques can be complementary to our approach. Our system does not focus on developing the recognition technology but instead it uses it as a means to continuously update its sensitive information discovery capabilities. Moreover, we abstract away from a specific application domain and make the first step towards developing the theory associated with the description of recognizers and the development of respective decognizers and their compositions for redacting information from media streams.

## VI. Conclusion

To keep pace with the evolution of recognition technology we argued the value of a close follower modular extensible redaction toolkit. We made the first step towards the development of a theory which can be leveraged to express correctness properties in the utilization and composition of recognizers and decognizers. To showcase the application of the proposed theory in practice we developed a prototype performing redaction on a variety of media streams for different application scenarios and identified key points for further development.

## References

[1] P. Ilia, I. Polakis, E. Athanasopoulos, F. Maggi, and S. Ioannidis, "Face/off: Preventing privacy leakage from photos in social networks," in *CCS*. ACM, 2015.

[2] S. Sadigh-Eteghad, M. Talebi, and M. Farhoudi, "Association of apolipoprotein e epsilon 4 allele with sporadic late onset alzheimers disease," *A meta-analysis. Neurosciences (Riyadh)*, 2012.

[3] D. Lopresti and A. L. Spitz, "Quantifying information leakage in document redaction," in *HDP Workshop*. ACM, 2004.

[4] E. T. Hassan, R. Hasan, P. Shaffer, D. Crandall, and A. Kapadia, "Cartooning for enhanced privacy in lifelogging and streaming videos," *CVPRW*, 2017.

[5] N. Raval, A. Srivastava, A. Razeen, K. Lebeck, A. Machanavajjhala, and L. P. Cox, "What you mark is what apps see," in *MobiSys*. ACM, 2016.

[6] R. Templeman, M. Korayem, D. J. Crandall, and A. Kapadia, "Placeavoider: Steering first-person cameras away from sensitive spaces." in *NDSS*. ISOC, 2014.

[7] S. Jana, A. Narayanan, and V. Shmatikov, "A scanner darkly: Protecting user privacy from perceptual applications," in *S&P*. IEEE, 2013.

[8] S. Jana, D. Molnar, A. Moshchuk, A. M. Dunn, B. Livshits, H. J. Wang, and E. Ofek, "Enabling fine-grained permissions for augmented reality applications with recognizers." in *USENIX Security*, 2013.

[9] F. Roesner, D. Molnar, A. Moshchuk, T. Kohno, and H. J. Wang, "World-driven access control for continuous sensing," in *CCS*. ACM, 2014.

[10] "Project website," https://goo.gl/7HFzPX, 2017.

[11] P. Aditya, R. Sen, P. Druschel, S. J. Oh, R. Benenson, M. Fritz, B. Schiele, B. Bhattacharjee, and T. W. I-pic, "A platform for privacy-compliant image capture," in *MobiSys*. ACM, 2016.

[12] B. Li and Y. Vorobeychik, "Feature cross-substitution in adversarial classification," in *NIPS*, 2014.

[13] B. A. Malin, K. E. Emam, and C. M. O'keefe, "Biomedical data privacy: problems, perspectives, and recent advances," 2013.

[14] I. H. Witten, E. Frank, M. A. Hall, and C. J. Pal, *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, 2016.

[15] A. Frome, G. Cheung, A. Abdulkader, M. Zennaro, B. Wu, A. Bissacco, H. Adam, H. Neven, and L. Vincent, "Large-scale privacy protection in google street view," in *ICCV*. IEEE, 2009.

[16] C. M. Cumby and R. Ghani, "A machine learning based system for semi-automatically redacting documents." in *IAAI*, 2011.

[17] M. Korayem, R. Templeman, D. Chen, D. Crandall, and A. Kapadia, "Enhancing lifelogging privacy by detecting screens," in *CHI*. ACM, 2016.