A Privacy Preserving System for AI-assisted Video Analytics

1st Clemens Lachner
Distributed Systems Group
TU Wien
Vienna, Austria
c.lachner@dsg.tuwien.ac.at

2nd Thomas Rausch
Distributed Systems Group
TU Wien
Vienna, Austria
t.rausch@dsg.tuwien.ac.at

3rd Schahram Dustdar
Distributed Systems Group
TU Wien
Vienna, Austria
dustdar@dsg.tuwien.ac.at

***Abstract*—The emerging Edge computing paradigm facilitates the deployment of distributed AI-applications and hardware, capable of processing video data in real time. AI-assisted video analytics can provide valuable information and benefits for parties in various domains. Face recognition, object detection, or movement tracing are prominent examples enabled by this technology. However, the widespread deployment of such mechanism in public areas are a growing cause of privacy and security concerns. Data protection strategies need to be appropriately designed and correctly implemented in order to mitigate the associated risks. Most existing approaches focus on privacy and security related operations of the video stream itself or protecting its transmission. In this paper, we propose a privacy preserving system for AI-assisted video analytics, that extracts relevant information from video data and governs the secure access to that information. The system ensures that applications leveraging extracted data have no access to the video stream. An attribute-based authorization scheme allows applications to only query a predefined subset of extracted data. We demonstrate the feasibility of our approach by evaluating an application motivated by the recent COVID-19 pandemic, deployed on typical edge computing infrastructure.**

***Index Terms*—privacy, artificial-intelligence, edge-computing, information-extraction, video-processing, attribute based authentication**

I. Introduction

Real-time video feeds from urban areas in combination with AI-based processing techniques provide exciting opportunities for novel smart-city applications [1], [2]. The emerging edge computing paradigm empowers these applications even more, where high-resolution cameras deployed in public spaces are complemented by specialized edge computing devices that can detect, process, and interpret various features of video streams. Such features include, e.g., motion detection, object detection, or face recognition. Besides potentially improving security in specific domains, this information extraction process can also act as an enabler for application developers to provide valuable services to potential customers. Applications leveraging such information include e.g., biometric authentication (smartphone unlocking), locomotive systems (autonomous driving), fitness related applications (detecting and correcting movements during an exercise), traffic monitoring, law enforcement (tracking of fugitives e.g., drivers escape), and many more.

However, the increasing number of cameras in public spaces cause growing concerns about the abuse of mass surveillance systems and the implications on personal privacy and freedom [3]. Therefore, adequate protection of private data is an increasing concern in all kind of domains making use of public video streams, such as health, financial, and social security. The most common straight forward generalized approach to protect sensitive data is the installation of access control mechanisms alongside with various encryption techniques, in order to protect data at rest and in transit. An exemplary video analytics implementation at the edge might incorporate a computing unit, connected to a camera, encrypting and transmitting a video feed via Transport Layer Security (TLS) to a cloud server, where some form of e.g., Role Based Access Control (RBAC) ensures that the decrypted video feed may only be processed by a entity with adequate permission or role.

In this paper, we follow an orthogonal approach to the provided example. Instead of applying encryption or privacy preserving image transformation techniques to a recorded video feed, we focus on the extraction of relevant information from the feed with the help of sophisticated machine learning techniques. This information only is then transmitted and made available, and, of course, also protected by similar mechanism as described in the previous example. The (raw) video is never transmitted or persisted/distributed permanently.

We propose a secure system design, that leverages state-of-the-art access control mechanisms featuring a Key-Policy Attribute Based Encryption (KP-ABE) scheme. Furthermore, we present in more detail a use case for analyzing the use of protective face masks in public areas, which, given the global pandemic situation due to the coronavirus disease 2019 (COVID-19), is highly plausible and relevant. The performance of a sample use case implementation is evaluated, aiming to demonstrate the feasibility of such a system running on typical edge computing hardware.

This work was partially supported by the European Union's Horizon 2020 research and innovation programme under grant 871525 (FogProtect). We thank Sebastian Kaindl, Christoph Doppelhammer, and Johannes Braitenthaller for their great support in the implementation of our evaluation software.

II. Related Work

Privacy and security are critical non-functional aspects of video analytics systems, and remain active areas of research. Recent research in particular has identified edge computing as a key enabler for privacy-sensitive systems that deal with real-time video processing [1], [4]. We discuss both frameworks for privacy for video analytics and surveillance in general, as well as specific methods of edge computing that enable our approach.

In the broad context of privacy in video-based media spaces, Boyle et al. [5] proposed a framework – a descriptive theory – that defines how one can think of privacy while analyzing media spaces and their expected or actual use. The framework explains three normative controls: solitude, confidentiality and autonomy, yielding a vocabulary related to the subtle meaning of *privacy*. A more technical introduction to video surveillance in general is given by Senior in [6]. The paper briefly summarizes the elements in an automatic video surveillance system, including architectures, followed by the steps in video analysis, from preprocessing to object detection, tracking, classification and behaviour analysis. Our proposed system builds on the high-level architecture described in this paper. We improve this architecture by considering AI-based video processing capabilities, and incorporate advanced security mechanisms. Furthermore, we suggest concrete hardware and software, proven to run with adequate performance in edge computing scenarios. Previous research on privacy mechanisms of video analytics systems often focuses on protecting the source video streams. For example, Upmanyu et al. proposed a privacy preserving video surveillance framework [7]. They split each frame into a set of random images, where each image by itself does not convey any meaningful information about the original frame. A blockchain-based approach was introduced by [8]. Chattopadhyay et al. demonstrate how the practical problem of privacy invasion can be successfully addressed through DSP hardware in terms of smallness in size and cost optimization [9]. This is particularly useful for edge computing scenarios, where computational resources may be scarce. Their access control scheme is based on a asymmetric key exchange mechanism, while regions of interest in the image are encrypted via AES. The work of [10] also focuses on encryption of an individual images.

Other, more application-specific approaches, often involve the preprocessing of video streams to anonymize or obscure specific parts of a frame, i.e. *denaturing*. An example is the work of Schiff et al. [3] that proposes *Respectful Cameras*, i.e., cameras that respect the privacy preferences of individuals. Their practical real-time approach preserves the ability to monitor activity while obscuring individual identities. This is achieved by identifying colored markers such as hats or vests, which are automatically tracked by their system. The identities of people wearing, e.g., a colored vest, are obscured by adding a solid overlay over the face on every image frame. Satyanarayanan et al. [1] proposed GigaSight, an Internet-scale repository of crowd-sourced video that also enforces privacy preferences and access control, and leverages edge computing technology. Closer related to our approach is the work done by [4]. They focus on camera-based digital manhunts of law enforcement agencies. Their approach leverages the inherent geo-distribution of fog computing systems to preserve privacy of citizens. If a camera system, mounted on the edge device, detects a face it sends a notification to the cloud. Though the authors state that an authorization mechanism is implemented, in order to access manhunt related data, they do not provide any specific details.

The previously presented approaches all focus on protecting or denaturing the source video stream. Our system is different in that it ensures that no frames are ever transmitted, therefore requiring new system design considerations. This design is motivated by the fact that, many applications do not require analyzing or recording the raw video feed, but instead only require filtered frames or extracted metadata processed by other video analytics components.

III. Motivating Scenario

Due to the recent pandemic situation caused by COVID-19, many countries imposed an obligation for people to wear facial masks in certain (mostly public) areas. Detecting if people adhere to such obligations may not only be of interest to law enforcement but also for virus transmission research and medical analysis. Public surveillance distributed at the edge, supported by adequate machine learning techniques (models and prediction accuracy), is capable of aiding in the detection and provision of relevant information, i.e., identifying clusters or numbers of people not wearing facial masks at a given location. However, despite its potential beneficial use, privacy aspects still have to be considered and the protection of sensitive data ensured. Applications, whether law enforcement related or for academic or societal purposes, do not necessarily need to store (raw or compressed) video feed, neither must they have access at any time to (live) video data, in such given use case described above. By implementing our system, relevant extracted information could in the simplest case be the number of people without masks per area, e.g., three people per 10 square meters. Combined with geodata of the surveyed area, interested parties would be able to get valuable insights and knowledge of peoples' behavior and adherence to possible obligations, and may take appropriate countermeasures. However, this is just a simplified example to demonstrate that there are real world use cases, where (governing) parties do not need to access a video feed directly in order to extract valuable information.

IV. System Design

In this section we explain our proposed system design in detail, focusing on hardware and architecture. Section IV-A will provide a view on the software side of the system. The foundation of our system is the architecture described in Section I. Fig. 1 gives an overview of the involved components and mechanism incorporated. The proposed systems assumes a processing and sensing unit (a), mounted at the edge of the

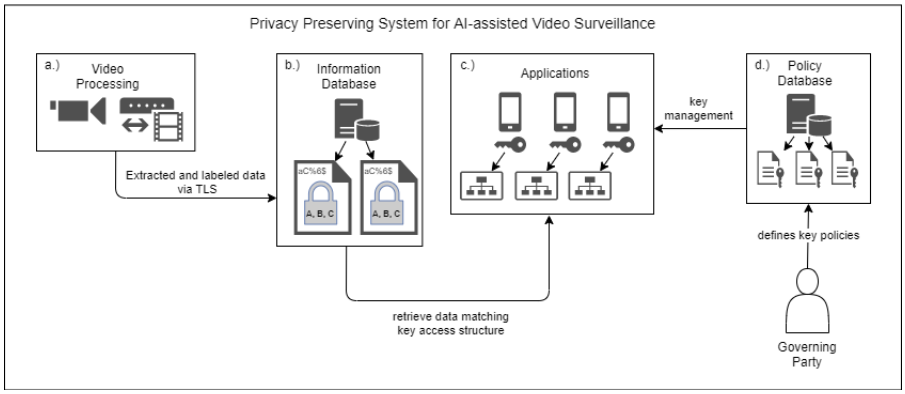

Fig. 1. Schematic overview of our proposed privacy preserving system for AI-assisted video analytics

network, e.g., on a smart lamppost. This unit comprises a high-resolution digital video camera, a computing device optimized for AI operations (e.g., NVIDIA Jetson TX2 [11]), and a computing device (from now on referred as caching device) that caches the extracted video information. Additionally, the latter can also persist video data if needed. As soon as the camera starts recording, video data is directly passed to the AI accelerated device. For simplicity reasons, we assume that one or more adequate pre-trained machine learning models are already deployed on this device.

After the information is extracted, subsets of this data are labeled according to a specification provided by, e.g., a service provider or governing entity. A simple example may be the number of people in a certain area not wearing a mask, according to the motivating use case described in Section III. The specific extracted information is an integer value, and the corresponding label could be `peopleWithoutMasks`. An adequate and sophisticated labeling procedure may play a more important role when dealing with more complex scenarios.

The extracted and labeled information is then passed to the caching device, where it gets transmitted further to the information database (b). The information database can run anywhere in the compute continuum, and facilitates both protected real-time access as well as access to historic data for batch analytics. The transmission of all extracted information is protected by the well established TLS standard, ensuring the integrity, authenticity, and confidentiality of data.

The information database, once extracted information is received, is then being encrypted using the Key-Policy Attribute-Based Encryption (KP-ABE) technique [12]. In this cryptosystem, ciphertexts are labeled with sets of attributes, i.e., our previously assigned labels. Furthermore, private keys are associated with access structures that control which ciphertexts a user is able to decrypt. Specific, fine grained access policies, define which user is allowed to access a certain labeled ciphertext for decryption. A user is able to decrypt a ciphertext if the attributes associated with a ciphertext satisfy the key's access structure. For instance, if Alice has the key associated with the access structure "X AND Y", and Bob has the key associated with the access structure "Y AND Z", they are not able to decrypt a ciphertext whose only attribute is Y by colluding. The KP-ABE system further allows deriving keys from other keys, based on their restriction hierarchy and access structure, i.e., each user's key is associated with a tree-access structure where the leaves are associated with attributes, allowing any user that has a key for access structure X to derive a key for access structure Y, if and only if Y is more restrictive than X.

In a KP-ABE, the encryptor exerts no control over who has access to the data they encrypt, except by their choice of descriptive attributes for the data. Rather, they must trust that the key-issuer issues the appropriate keys to grant or deny access to the appropriate users. In our case, users would be applications that may only need a specific subset of the extracted video information. A simplified example of access to encrypted data via policy defined access structure in a KP-ABE system is shown in Fig.2. Accessing only this subset of data is reflected in the application's private key (c), determined by a policy. An application, firstly when deployed, is issued with such a key and is notified by the policy database if its key is modified. This allows for a seamless fine-grained management of access control for any application without the need of a re-deployment. The policies are stored and managed at a dedicated independent location at the edge or in the cloud (d). Managing those policies could be done by governing parties for example; but in this paper we do not further address this issue. Furthermore, we notice that if features not specified at design time are needed by an application, those unsupported feature extraction has to be re-implemented and deployed by e.g., a service provider or governing party. Once the extracted information is correctly encrypted, it is possible for applications to access this information via a well defined separate API, via a corresponding Attribute-Based Access Control (ABAC) mechanism. The API never allows by design for an application to directly communicate with a video processing unit at the edge, thereby prohibiting theoretical access to the video feed. An application is now able to further process the extracted information depending on their specific needs.

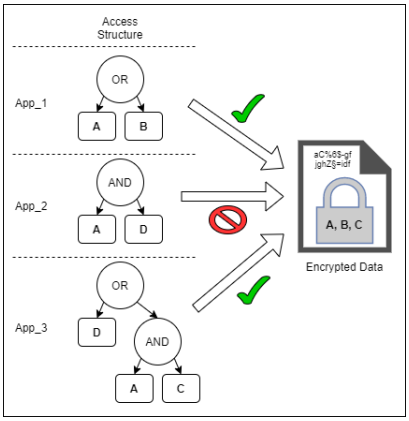

Fig. 2. Access to encrypted data via policy defined access structure in a KP-ABE system

A. Sample Implementation

In this section we showcase a concrete exemplary implementation of our proposed system based on the scenario described in Section III. Systems for face or object detection are well understood, as they have been broadly studied, implemented and evaluated over the recent years. Since the outbreak of COVID-19, machine learning models and their applications for mask detection have gained increased attention of researchers and developers world-wide.

The detection accuracy for real-time systems mostly depends on external factors like, lightning conditions, view angle and even skin color of observed persons. In our example, we adapted the code from [13] to run on the different hardware nodes of our evaluation testbed. The extracted information, i.e., probability of a mask being detected alongside with the count of people in a given frame, is labeled and transferred over the internet, encrypted via standard TLS, to the information database. On the information database runs a KP-ABE scheme [14], which is responsibility of the re-encryption of initial ciphertexts, i.e., incorporating the attribute groups into the ciphertext. The information database hosts a simple REST-API providing access to the extracted information, given proper authentication. A sample Android application is then able to query only information which labels are reflected in the private key deployed on the smartphone. A potential limiting factor in this application chain is the network latency, which depends on multiple environmental factors. Therefore, we did not include measurements regarding network related performance into our evaluation. This sample application aims to showcase the scenario executed on dedicated edge computing hardware and evaluate the core systems tasks, i.e., AI-assisted object detection and encryption.

B. Evaluation

In our experiments, we focused on the AI-assisted information extraction process and the corresponding encryption tasks. Therefore, we deliberately executed the video processing tasks and the information database on the same device.

Our testbed comprises a heterogeneous set of typical edge computing hardware. First, a laptop with an Intel i7-7700 CPU@4.2GHz and 16GB RAM. Second, a Nvidia Jetson TX2 Developer Board with a ARM Cortex-A57@2GHz CPU and Pascal GPU and 8GB RAM. Third, a Raspberry Pi4 with a ARM Cortex-A72@1.5GHz CPU and 4GB RAM. The AI-assisted information extraction process, i.e., detecting the number of people wearing a mask, is computational expensive. For the evaluation, we chose three short publicly available video sequences, showing varying numbers of people wearing a mask. The first video shows a single person putting a mask on an off. The second video footage (labeled Multiple Persons in Fig 3) constantly shows five people taking on and off their masks, while the third video (labeled Crowd in Fig 3) shows a large amount of people (>10; some wearing a mask, some do not) constantly varying in number. This videos are the input for our system, where the number of people wearing a mask is extracted and encrypted on a frame per second (FPS) basis. If the FPS processed by our system matches the FPS the input video is recorded with (e.g., 25 FPS), real-time performance is achieved, i.e., a user could potentially read the extracted information in real-time, but obviously this still also depends on the network conditions. We have to notice that our mask detection implementation is a chaining process of a face detection algorithm and a mask detection algorithm, each working with its own dedicated model. While the point of the paper is not to implement a performance optimized mask-detection framework, this chaining procedure obviously greatly affects the overall performance of the system. Fig. 3 shows the results of our experiments. The overall performance is mainly affected by the AI-specific tasks and furthermore on the conditions and specifics of the information that needs to be extracted, e.g., an increase in number of people leads to a massive decrease in performance. The encryption tasks are commonly CPU-bound, scaling linearly with CPU-speed and/or are also dependent on the available specific encryption algorithm based hardware instruction of a given CPU, like e.g., the AES instruction set which is integrated into many modern processors [15]. The extracted plain text information, concerning our scenario, is rather small (compared to e.g.,

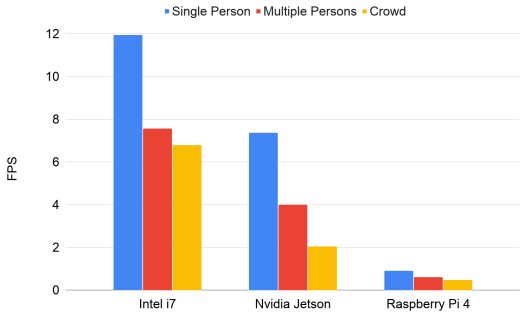

Fig. 3. Performance of information extraction and encryption of different video input

encrypting the whole raw video data), hence the computational effort to produce ciphertext minimal. Therefore, the overall contribution to the performance capabilities of our system is neglectable compared to the information extraction process. Research has shown that modern edge computing devices are capable of executing AI-assisted video processing, without significant loss in performance [16]–[19]. Hence, in order to achieve real-time performance, the suggested ABE-based encryption scheme is not a bottleneck in the system performance-wise, but rather the combination of used hardware and the nature of the AI-assisted feature extraction task. To address this problem, strategies like lowering the sampling rate of the video for feature extraction, reducing the input size, etc. could be incorporated.

V. Conclusion

Video cameras deployed in urban areas provide enormous value for novel smart-city applications, but at the same time, cause legitimate privacy concerns. These concerns are mostly related to the unrestricted access and recording of the raw video feed, and potential abuse for mass surveillance. We have found, however, that many applications do not require this access in the first place. Instead, we argue that video analytics should be pushed to the extreme edge, and direct access to the video feed should be avoided. To that end, we have presented a privacy preserving system for AI-assisted video analytics. It features a decoupling architecture that effectively hinders applications from directly accessing the underlying video feed, and instead allows them to advertise what type of information they require. Our system then extracts the information using existing AI-based video processing techniques, ensures that privacy preferences are met, and facilitates the secure access to the extracted information for both real-time and batch applications. A ciphertext (i.e., the encrypted information extracted from video data) is labeled with certain attributes, which only allows applications with a matching private key (i.e., the attributes corresponding to the labels of the ciphertext are encoded in the key) to decrypt and access the data. A KP-ABE security scheme ensures that only authorized parties have access to this extracted information. To allow for a more fine grained access control, security policies determine which application is able to decrypt specific subsets of the encrypted extracted data. The policies are stored and managed at a dedicated policy database, located at the edge or in the cloud. Furthermore, it is responsible for issuing keys to an application, as well as notifying applications if a key's attributes change. Hence, a seamless fine-grained management of access control for any application without the need of a re-deployment is achieved. We showed that our system is able to run on typical edge computing hardware, by implementing and evaluating a simple, yet due to the recent pandemic situation highly relevant scenario.

References

[1] M. Satyanarayanan, P. Simoens, Y. Xiao, P. Pillai, Z. Chen, K. Ha, W. Hu, and B. Amos, "Edge analytics in the internet of things," *IEEE Pervasive Computing*, vol. 14, no. 2, pp. 24–31, 2015.

[2] T. Rausch and S. Dustdar, "Edge intelligence: The convergence of humans, things, and ai," in *2019 IEEE International Conference on Cloud Engineering*, ser. IC2E'19. IEEE, 2019, pp. 86–96.

[3] J. Schiff, M. Meingast, D. K. Mulligan, S. Sastry, and K. Goldberg, "Respectful cameras: Detecting visual markers in real-time to address privacy concerns," in *Protecting Privacy in Video Surveillance*. Springer, 2009, pp. 65–89.

[4] M. Grambow, J. Hasenburg, and D. Bermbach, "Public video surveillance: Using the fog to increase privacy," in *Proceedings of the 5th Workshop on Middleware and Applications for the Internet of Things*, 2018, pp. 11–14.

[5] M. Boyle, C. Neustaedter, and S. Greenberg, "Privacy factors in video-based media spaces," in *Media Space 20+ Years of Mediated Life*. Springer, 2009, pp. 97–122.

[6] A. Senior, *Protecting privacy in video surveillance*. Springer, 2009, vol. 1.

[7] M. Upmanyu, A. M. Namboodiri, K. Srinathan, and C. Jawahar, "Efficient privacy preserving video surveillance," in *2009 IEEE 12th international conference on computer vision*. IEEE, 2009, pp. 1639–1646.

[8] D. Lee and N. Park, "Blockchain based privacy preserving multimedia intelligent video surveillance using secure merkle tree," *Multimedia Tools and Applications*, pp. 1–18, 2020.

[9] A. Chattopadhyay and T. E. Boult, "Privacycam: a privacy preserving camera using uclinux on the blackfin dsp," in *2007 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2007, pp. 1–8.

[10] Z. Erkin, M. Franz, J. Guajardo, S. Katzenbeisser, I. Lagendijk, and T. Toft, "Privacy-preserving face recognition," in *International symposium on privacy enhancing technologies symposium*. Springer, 2009, pp. 235–253.

[11] NVIDIA. (2020) Nvidia jetson tx2 module. [Online]. Available: https://www.nvidia.com/de-de/autonomous-machines/embedded-systems/jetson-tx2/

[12] V. Goyal, O. Pandey, A. Sahai, and B. Waters, "Attribute-based encryption for fine-grained access control of encrypted data," in *Proceedings of the 13th ACM conference on Computer and communications security*, 2006, pp. 89–98.

[13] C. Deb. (2020) Face mask detection. [Online]. Available: https://github.com/chandrikadeb7/Face-Mask-Detection

[14] S. Agrawal. (2020) Attribute-based encryption. [Online]. Available: https://github.com/sagrawal87/ABE

[15] C. Lachner and S. Dustdar, "A performance evaluation of data protection mechanisms for resource constrained iot devices," in *2019 IEEE International Conference on Fog Computing (ICFC)*, 2019, pp. 47–52.

[16] J. Redmon and A. Farhadi, "Yolo9000: better, faster, stronger," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 7263–7271.

[17] J. Huang, V. Rathod, C. Sun, M. Zhu, A. Korattikara, A. Fathi, I. Fischer, Z. Wojna, Y. Song, S. Guadarrama *et al.*, "Speed/accuracy trade-offs for modern convolutional object detectors," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 7310–7311.

[18] X. Ran, H. Chen, X. Zhu, Z. Liu, and J. Chen, "Deepdecision: A mobile deep learning framework for edge video analytics," in *IEEE INFOCOM 2018-IEEE Conference on Computer Communications*. IEEE, 2018, pp. 1421–1429.

[19] Z. Lu, S. Rallapalli, K. Chan, and T. La Porta, "Modeling the resource requirements of convolutional neural networks on mobile devices," in *Proceedings of the 25th ACM international conference on Multimedia*, 2017, pp. 1663–1671.