# Parental Control with Edge Computing and 5G Networks

Sara Ramezanian, Tommi Meskanen, Valtteri Niemi
University of Helsinki and
Helsinki Institute for Information Technology
Helsinki, Finland
sara.ramezanian, tommi.meskanen, valtteri.niemi@helsinki.fi

***Abstract*—Parental control methods are a popular tool to keep children safe in the digital world. Usually, digital parental control methods function by disclosing the entire log of child's on-line activities to their parents, and therefore, these methods do not consider the privacy of the child. Moreover, not all children are under protection, as not all parents provide this service for their children. In this paper, we propose a privacy preserving parental control protocol with edge computing that uses Artificial Intelligence techniques to automatically detect harmful content for minors in 5G networks. Moreover, our protocol provides protection for all children, regardless of whether they have parents who pay for this extra service. Artificial Intelligence makes it possible to classify digital content automatically and also in real time. In order to make our protocol privacy preserving, we use multi-party Private Set Operation protocols.**

## I. Introduction

In the world where the employment of Internet in everyday life is increasing, the usage of smart devices by children is inevitable. In average, children spend significant amount of time on-line [1].

Although there are lots of useful and kid-friendly material on the Internet, there is also harmful content which is not suitable for children. Several studies have sought short-term and long-term effects of digital world on children and adolescents [2], [3], [4]. However, a comprehensive parental control technique is missing in academia.

It has been shown that children have the tendency to overshare on social media, which creates more opportunities for abusers to find their victims [5]. On the other hand, a parent's knowledge of security and privacy in the digital world has a direct effect on the child's security and privacy on the Internet [6]. The importance of parental control use case has also been identified in the standardization [7].

The current parental control applications and methods are designed such that the children have no privacy while connected to the Internet [8], [9]. In other words, parents and Internet providers can have a direct access to the children's digital activities [10]. In this paper, we present an Artificial Intelligence (AI) assisted privacy preserving parental control protocol with edge computing in 5G networks. The contributions of this work are as follows:

- To the best of our knowledge, we propose the first privacy preserving parental control protocol that preserves the privacy of parents towards their children, the service providers and the network. Our protocol also preserves the privacy of the children towards their parents. When using our protocol, service providers and the network learn less information about the child's on-line activities, compared to the case where our protocol is not used.
- Our protocol provides automatic protection for all children, even in the situation where parents are not involved with the digital life of their children.
- Instead of blocking the child's access to certain materials, we suggest a method that we called a *Smart response*. A smart response is an automatic response generating technique that is used when the child is trying to access content that are harmful. Depending on the use case, a smart response can be, e.g., a URL or a pop up message that is designed to educate and entertain children.

Moreover, we study the existing AI methods that can be deployed in automatic detection of harmful digital content, for the purpose of parental control.

## II. Preliminaries

In design of our privacy preserving parental control protocol, we utilize several AI methods, computational techniques and privacy enhancing technologies. In this section, we present the necessary background on the concepts that are required to understand the rest of this paper.

### A. Edge Computing

With smart devices coming more widespread, increase in data generation is inevitable. *Edge Computing* is a method that handles the data at the edge of a network, where big part of the data is generated [11]. For example, a smart phone can act as the edge for all the Internet of Things (IoT) devices that are connected to that phone.

In our protocol, we use several classifiers. In order to reduce the possible latency and transmission costs that our method may cause, we assume that the classifiers are at the edge servers. Moreover, we use several applications and technologies that already exist and could be used at the edge of a network. AI-edge applications such as the ones presented in [12], AI accelerators such as Intel Neural Compute Stick 2 [13], and edge manager such as IBM Edge Application Manager [14], are few examples of such technologies that can be used in the edge server to perform parental control services.

ISSN 2305-7254

Moreover, there have been several studies on the methods to distribute the policy control at the edge of the network [15], [16]. These methods can be used as building blocks in our parental control system.

### B. 5G Networks

5G networks are the fifth generation of mobile networks. Compared to its predecessor, LTE/4G, 5G aims at significantly improving properties such as very high throughput (1-20 Gbps), ultra-low latency (<1ms), massive connectivity, and low energy consumption. Therefore, in addition to servicing cell-phones, 5G can cover a broad range of new use cases such as Virtual Reality, IoT, Augmented Reality, etc. Full list of all the components in 5G networks and their functionalities can be found in [17]. Next, we briefly introduce four functions that are used in our protocol.

*User Plane Function (UPF)* is the main function in charge of handling the data traffic (i.e. packet routing and inspection, etc.) from User Equipment (UE).

*Access and Mobility Management Function (AMF)* provides several functionalities, e.g., access authentication, access authorization, reachability management, etc.

The main responsibility of *Policy Control Function (PCF)* is to define policy rules. Other control plane functions use these policies to provide suitable service for each subscriber.

The external exposure of network functionalities is supported via the *Network Exposure Function (NEF)*. Few examples of such external exposures are policy capability, analytics reporting capability, and monitoring capability.

### C. Private Set Intersection

A Private Set Intersection (PSI) is a cryptographic protocol between two or more parties. Each party has a private set, and the protocol aims to obtain the intersection of these sets. After executing the protocol, no information about the elements of the sets that are not in the intersection will be revealed [18].

A special case for PSI protocol is when the protocol is between two parties, one of which has just one element and the other one has a set of elements [19]. These two parties want to check whether that one element is a member of the set, in a privacy-preserving manner. This protocol is called Private Membership Test (PMT) [20].

In our parental control protocol, we need more complex privacy preserving set operations than PSI. We also require that a third party is the only one that receives the outcome of the Private Set Operation (PSO), and that none of the other parties that are involved in the protocol should learn anything about the outcome of the protocol. In 2021, Ramezanian et al. presented a general solution to any PSO, where the outcome of the protocol is only revealed to one special party, called an external decider, who does not have an input set [21]. They presented two protocols for the general problem of PSO; one protocol is suitable for the case where the number of possible elements are limited, and the other protocol is suitable for the case where only the cardinality of the output set is required. The first protocol –with the limited universe– uses an additively non-deterministic homomorphic encryption, and the second protocol uses a keyed hash function.

### D. TLS Session

Transport Layer Security (TLS) [22] is a cryptographic protocol that provides privacy and data integrity between communication devices over Internet. TLS is used in many applications such as web browsing and instant messaging. In TLS there is an initial set up phase after which all communication between the two parties is secured.

## III. Problem statement

We formulate the problem of privacy preserving digital parental control as follows.

A parent has a list, which contains attributes of such content that their child should not be exposed to. This forbidden list of attributes should remain private from the child and the network. The child may want to access to Internet without disclosing all their on-line activities to their parent, and neither to the network. The network wants to provide a safe on-line environment for all children also when there is no input from parents.

Softwarization and virtual nature of 5G networks are designed in such a way that it is possible to add extra modules (such as parental control related functionality) to the system. A parental control feature in the architecture of 5G provides safety and privacy for all children. Compared to the traditional parental control applications that are installed on the child's device, the parental control that is provided by the network makes it harder for children to bypass this control.

In this paper we present a privacy preserving parental control protocol, that can be used in 5G networks. Although our protocol protects privacy of parents, it does not enable parents to put arbitrary restrictions on their children. Our approach would benefit from support in 5G standards but it could also be realized as a proprietary solution offered by a mobile network operator.

## IV. Related work

In this section, we first deliver the current state of the art on the existing parental control applications and methods. Then, we briefly describe the existing AI methods that can be utilized in our parental control method.

### A. Parental Control Methods and Applications

Many parental control technologies are in use and they operate locally. For instance, Wi-Fi routers with parental control functionality, built-in services such as parental control for installing applications on Android, parental control applications that can be purchased via IT companies [23], [24].

Next, we briefly give examples of some weaknesses of the above local parental control technologies. With Wi-Fi routers, child may be able to circumvent the control by connecting to the Internet via cable. Also, the child is only safe in the environments that are covered with a kid-safe router. The built in services have limited functionality, and can be bypassed by

a child. For example, a child can bypass the Android parental control for installing applications by guessing the parent's pass-code. The parental control applications that are provided by IT companies mostly operate in such way that the child has no privacy.

These weaknesses lead us to propose a network-based parental control technology.

The topic of parental control has been studied in many sociological and psychological papers, e.g., [25], [26], [27]. A comprehensive study of the current state of the art in digital parental control can be found in [28].

### B. AI-assisted Technologies for Parental Control

In this section, we suggest several AI methods that can be used to automatically detect content harmful to children in the on-line world, that are not suitable for children.

Classifying the web-pages and automatically detecting certain attributes in a web-page [29] have been studied intensively for various reasons, such as recognizing web-sites that are involved with human trafficking, encourage racism, provide access to drugs. An example of web-page classification in real-time can be found in [30]. In addition to classification, AI-assisted webpage filtering can also be used for the purpose of the parental control [31], [32].

Text classification methods are designed to automatically detect hate speech, fake news, etc. For the parental control reasons, we need text classifiers to protect children against harmful content such as cyberbullying [33]. In [34], authors presented a real-time text classifier. Moreover, we need to use AI-assisted sentiment analyses technologies that help to correctly classify potentially harmful textual content [35], [36].

For parental control purposes, it is important to automatically detect applications that are harmful for children. In [37], Luo et al. proposed a novel AI-based technique to detect kid-friendly Android application for different age groups.

## V. The protocol

In this section, we present our privacy preserving parental control protocol in 5G networks. We utilize AI methods and edge computing in our protocol. The goal of our protocol is to protect children from harmful digital content, while preserving their privacy towards the network providers and towards their parents. Moreover, we want to preserve the parents' privacy towards their children and towards the network providers.

Our protocol provides automatic protection for children in the on-line world, even in the situation where the child's parents are not involved with their digital life. Therefore, implementing our protocol in the real world would provide safety and privacy that is available for all children who have access to the Internet.

In our protocol we assume that every child has an application installed on their mobile device. We call this application *Kid-client*. Whenever the child wants to access a website, and send/receive a text message, the kid-client acts as the middle man between web/messaging applications, and the network. We also assume that the messaging applications and the browsing applications are aware of the kid-client, and can communicate with it.

Moreover, we assume that if a parent of a child wants to be involved in the parental control process, they also have installed an application on their mobile device which we call *Parent-client*. Parent-client is only needed in case where there is a parent who wants to have influence on the parental control process that is provided by the network.

As the number of legal guardians of a child might vary between different house-holds and different cultures, a child can have more than one person who has the parenting role in their life. In this case, we assume that there is one person who has the role of administrator and therefore uses parent-client application.

In the case where the parents are divorced and have shared custody for their child, there can be two (parent-client, kid-client) pairs for that child. Each pair has a validity time period. In this way, based on the time, the network knows which pair of (parent-client, kid-client) is currently in control of the child's UE. However, for simplicity and without loss of generality, hereafter, we assume there is only one person who is responsible as "the parent". Therefore, it is enough to consider one parent-client per kid-client.

We believe the network cannot use binary classification (good and bad) for digital content. This is due to the fact that a certain content that is not suitable for a 6 years old child, might be considered benign for a 10-year old user. For example, there may be a cartoon which contains scenes that are considered to be violent for a 6 years old, whereas the same cartoon might be listed as non-violent for a 10 years old. Therefore, when analyzing the content, we take the age of the child also into consideration.

The network stores several pieces of information for each subscriber that is marked as a child including: identity of the child's device (UE), the age of the child, information about the kid-client on the child's UE, type of the parental control service that has been subscribed and, when applicable, information about the parent-client that corresponds to that child.

We present parental control services for the situation where the child wants to access a website or send/receive text messages. We leave other situations, such as where the message to/from child contains audio, video and/or picture, for future work.

We also present the time and communication complexities of different parts of the protocol. The computation time is obtained by running the protocol on an x86-64 Intel Core i5 processor clocked at 2.7 GHz with a 4 MB L3 cache. As the cryptographic part of each protocol is the most time consuming part, in our implementation we only consider the cryptographic steps.

There are seven components involved in our protocol: A *Child* with a device that connects to Internet, the *Parent* of this child, the *Kid-client*, the *Parent-client*, the *Edge Server*, the *Service Provider* such as website host., and the *Network Provider*.

### A. Smart Response

As we mentioned in the Introduction, our protocol uses a technique that we call *Smart Response*. We use a smart response, whenever our parental control protocol decides that certain content is harmful for a child and therefore, the child's access to that content should be blocked.

A smart response is an AI based technique to influence the child's behavior on-line, in a positive way. For instance, let us assume that a child wants to access to a potentially harmful website that is about a movie that contains violent scenes. Instead of just blocking this child's access to the website that they were initially requesting, our protocol replaces the original requested URL with another URL that contains materials related to movie(s) that are interesting and educational for that specific child, within their age-group. This replacement can be done by first using classifiers to automatically categorize the original URL that the child was requesting, and then finding a suitable replacement which is similar in content (but not harmful) to that original URL. Examples of such smart responses are: news, video clips, memes, music. This type of strategy has been used in different contexts (e.g. suggesting a smart response to an email based on its content [38]), and we introduce it also in the context of digital parental control.

### B. Edge Computing for Parental Control

We utilize AI techniques to detect harmful digital content automatically. We use several classifiers to analyze the content that the child (which is under parental control) wants to access. For making it easier to preserve privacy, and also in order to reduce the latency that may appear when utilizing these classifiers, we use edge computing in the architecture of our protocol. Compared to cloud computing, utilizing edge computing provides better privacy because the data processing is done closer to the source that generated the data, than in cloud computing.

In our parental control protocol, we assume that the traffic related to the User Equipment (UE) that are in the functionality range of an Edge Server, is being handled by that Edge Server. Fig. 1 shows the edge computing paradigm that we use in our protocol. In this figure, devices UE 1 to UE 5 are in the functionality range of Edge Server 1, and devices UE 6 to UE 10 are in the functionality range of Edge Server 2.

We design the Edge Server such that it can perform the parental control functionality. Among other functionalities, the edge server has *Edge Application Manager*, *Parental Control* function, and *AI* function. The parental control function has *allow* and *deny lists*. The parental control function includes a *proxy server* [39]. The components of our edge server are presented in Fig. 1.

As soon as a device (e.g. UE 1 in Fig. 1) that belongs to a child goes on-line in the network, AMF finds the closest edge server to that device (e.g. Edge Server 1 in Fig. 1). Then, AMF notifies the edge application manager in that edge server about child device UE1 being in its functionality range. Moreover, AMF gives the edge server the cellular network identities of the paired parent-client and kid-client. The edge application manager in Edge Server 1 notifies the child's kid-client that it is the edge server that is responsible for handling the child's traffic. If UE 1 changes its location so much that it is no longer in the range of its assigned edge server, AMF assigns another edge server to this device.

As we mentioned before, the exposure of the core network to external 3rd party functions is supported by NEF. In other words, new applications such as parental control interact with PCF via NEF. In this paper, for simplicity, we do not explore the role of the NEF any further.

### C. General Check and Personalized Check

The network provides two types of checking for the purpose of protecting children: *General Check* and *Personalized Check*.

The general check is a service that the network provides for all its users who are children. This service automatically prevents children from being exposed to harmful content on the Internet.

The network provides the personalized check service for parents that wish to have more influence on their child's activity on the Internet. For instance, a parent decides that it is better to prevent their child, who is suffering from anxiety attacks, from being exposed to violent content. Therefore, this parent has zero tolerance for violent content, even though age-wise the content could be considered to be acceptable for this child. The personalized check can be a service that is provided only for parents who registered as premium subscribers, or it can be available for all parents.

As we explained before, AMF decides which edge server (e.g. Edge Server 1) is responsible for the UE that is under parental control. Moreover, AMF informs Edge Server 1 which kind of checking should be carried out for this UE. Finally, if the personalized check is performed for UE 1, AMF informs the edge application manager in Edge Server 1 about the paired parent-client and kid-client that correspond to UE 1.

### D. Accessing a Website via General Check

Now, we explain how the general check works in the case where the child is trying to access a website.

The off-line phase of our parental control protocol to access websites in the general check is as follows:

1) For each age-group, an AI-based module in the core network classifies the well-known kid-friendly websites in an allow list and the notorious websites with harmful content for children in a deny list.
2) Core network generates several smart responses for each age-group.
3) PCF defines proper policies for each age group. For instance, certain type of violence might be considered harmless for a child above 12 years old, while the same content should be blocked for children younger that 12 years old. Cultural factors may effect what is considered proper content for each age-group.
4) PCF notifies UPF about proper policies for each age group.

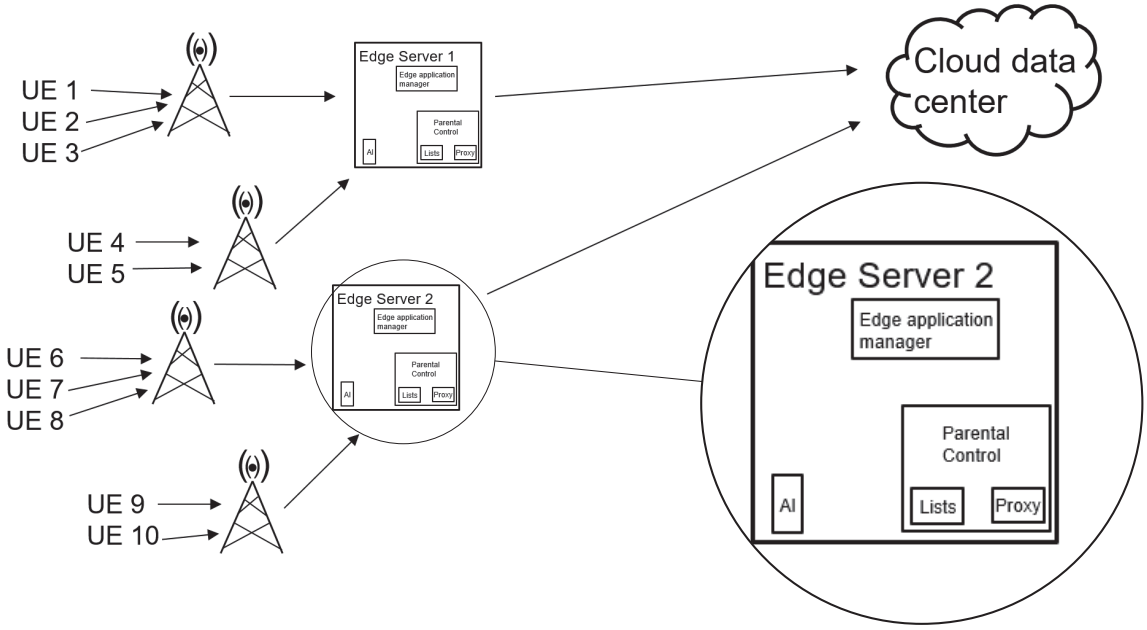


Fig. 1. Edge computing for parental control

5) AMF notifies UPF which users are children (e.g. UE 1 in Fig. 1).
6) UPF broadcasts the policies, the smart responses, and allow and deny lists to all edge servers.

The first four steps is initialized once, and afterwards, UPF will broadcast any possible updates for the allow and deny lists, list of smart responses, and policies.

As soon as the child's device (UE 1) is turned on, AMF finds the closest edge server to UE 1 (e.g. Edge Server 1 in Fig. 1).

Now, the child who possesses UE 1 wants to access to a website. Therefore, the on-line phase of the protocol starts as it is explained in the following.

1) The child enters the URL address in the address bar of the web browser.
2) The browser connects to the kid-client and sends the URL to the kid-client.
3) The kid-client starts a TLS session with the proxy in the edge that has been assigned to its device in the off-line phase of the protocol.
4) Now, the edge knows which URL the child is interested in. The edge checks this URL against its allow and deny lists.
5) If the URL is in the allow list, the edge starts another TLS session with the server which is hosting this website, gets the content of the website and sends it back to the kid-client.
6) If the URL is in the deny list, the edge picks a URL from the list of smart responses that is relevant to the original URL that the child requested. The URL is chosen based on the age of the child and content of the original requested URL.
7) The edge sends the content of that website to the kid-client.
8) If the URL is not in any of the lists, the edge sends the content of the website to the AI module of the edge that classifies this URL. If the page is classified as benign, the kid-client gets the content of this website, and otherwise, the edge picks a URL from the list of smart responses and sends the content of that website to the kid-client as in the previous step.

The time complexity of the on-line phase is acceptable because, even in the situation where the URL is not in any of the allow or deny lists, the classification can be done in real-time [30].

The parental control function in the edge server notifies the edge manager about possible updates in the allow/deny list. The edge manager sends the updates and the age-groups which these updates are suitable for, to the core network (e.g. every hour). Note that if a website is considered to be benign for a 6 years old, it is suitable for all children older than 6 as well. Also, if a website is classified as harmful for a 10 years old child, it is harmful for all children younger than 10 as well.

### *E. Accessing a Website with Personalized Check*

Now, let us consider a case where the parent wants to influence the website accessibility of their child. Let us assume that the parent has a set $W_p$ of attributes that the child should not be exposed to, while the child wants to access a website that contains certain attributes $W_c$. We want to preserve the privacy of the child and the parent towards each other and towards the network provider. Therefore, we use a PSI protocol. We need to use a special variant of the PSI protocol where:

- Two parties have input sets.
- The result is not the intersection itself but just one bit of information, namely whether the intersection is empty or not.
- The result goes to a third party.
- The protocol allows one party to complete their part of the protocol before the other party even knows what their set is going to be.
- One party ("first" party) does not need to communicate with the third party directly, only via the other party ("second" party).

Such protocol has been developed by Ramezanian et al [21]. The parent-client is the first party who has the set $W_p$ as input while the second party is edge server who has the set $W_c$. The kid-client is the third party. Note that neither the child nor the kid-client is able to see $W_c$. The parent-client creates the set $W_p$ in the set-up phase of our protocol, and therefore, the required time to create and perform PSI computation on $W_p$ can be pushed to the off-line phase of the protocol.

The off-line phase of the personalized check has all the steps of the off-line phase of the general check. Moreover, we add the following steps to the off-line phase.

1) Via a module that connects to UPF the network generates a set $U$, which contains the attributes that need to be checked in the packets, e.g. $\{$violence, bullying, drug, game, $\cdots\}$.
2) UPF shares the set $U$ with all parent-clients and all edge servers.
3) UPF sends several addresses of other edge servers to the Edge Server 1 which is assigned to handle UE 1.
4) Parent gives their list of forbidden attributes to the parent-client, thus creating the set $W_p$ (Off-line i in Fig. 2).
5) Parent-client performs the necessary computations for the PSI protocol and sends the result to Edge Server 1 (Off-line ii in Fig. 2).

On-line phase of the protocol starts when the child enters a URL in the browser (On-line I in Fig. 2). As explained before, the web browser sends the child's URL to the kid-client (On-line II in Fig. 2). The on-line phase of the personalized check to access a website is as explained in the following:

1) Kid-client starts a TLS session with Edge Server 1 (On-line III in Fig. 2).
2) Edge Server 1 checks the URL against its lists. If the URL is in the deny list, the Edge Server 1 sends a smart response to the kid-client and the protocol is done, otherwise:
3) Edge starts another TLS session with the server which is hosting the requested web-page.
4) The edge analyzes the web-page, with the help of the AI function.
5) The edge creates a set of attributes based on the results it gets from the AI function $W_c$.
6) The edge server executes the PSI protocol between $W_c$ and the parent's set of forbidden attributes $W_p$, and sends the encrypted result to the kid-client (On-line IV in Fig. 2).
7) The edge server chooses a URL for a potential smart response and sends it to the kid-client.
8) The kid-client decrypts the result of the PSI. If the intersection is non-empty then the web-site is not OK for the child, otherwise it is OK.
9) If the original URL is not OK, the kid-client uses the URL suggestion and replaces the original one with this suggested URL and sends the request to another edge server (Edge Server 2 in Fig. 1). If the original URL is OK, the kid-client sends the URL to another edge server (Edge Server 2). This step is shown in Fig. 2 as On-line V.
10) Edge Server 2 sends the content of the accepted web-page to the kid-client.
11) Finally, the kid-client sends the content of the web-page to the browser (On-line VI in Fig. 2).

In order to estimate the computation time of this part of the protocol, we use the PSI protocol by Ramezanian et al. with Paillier cryptosystem.

The network creates an ordered set $U$ which contains the attributes that are needed to be checked in the packets. Let us assume that there are 20 items in the set $U$ and the modulus $N^2$ is 4096 bits long. In the off-line phase of the protocol, the parent-client encrypts the parent's set $W_p$ in 2.8 seconds. Edge Server 1 also needs 2.8 seconds to encrypt $W_c$. In the on-line phase of the protocol, the edge server needs 0.14 seconds to perform computations between $W_p$ and $W_c$. The kid-client decrypts the result of PSI protocol in less than 3.4 seconds. Communication complexity of this protocol is presented in the Table I. Figure 2 shows an overview of our protocol, when the child wants to access a website with the personalized check.

Please note that we only report the computation time of the cryptography part of the protocol, because redirection of a URL and using proxy are common events in networking, and they are relatively faster than a PSI protocol.

### *F. Sending a Message via General Check*

In our parental control protocol, if a child wants to send a text message, the network automatically checks the message for any harmful content, such as bullying, violence, and drugs. To protect the privacy of the child, we use a PSI protocol as a key building block for our protocol. We apply a 2-party PSI protocol between the kid-client and the edge server that only reveals to the parties whether the intersection of the two input sets is empty. The off-line phase of the protocol is as follows.

1) The core network collects a deny list of harmful words $B = \{b_i\}$.
2) Core network broadcasts this list to all the edge servers.

The private input set of Edge Server 1 is $B$. Now, the on-line phase of the protocol starts.

1) The child has a message which contains a set of words $M = \{m_j\}$.
2) The kid-client and the Edge Server 1 together execute a PSI protocol between their respective sets $M$ and $B$.

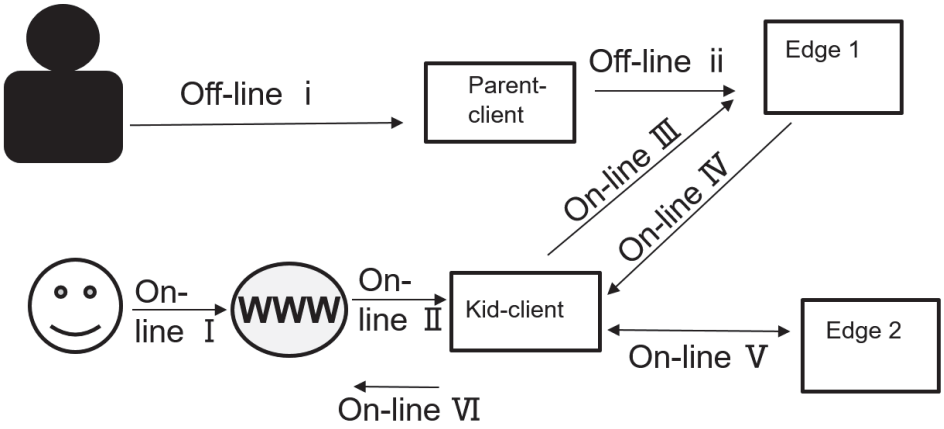


Fig. 2. An overview of the off-line and on-line phases of the personalized check to access a website in our protocol

3) If the intersection $M \cap B$ is empty, the Edge Server 1 will forward the message to recipients.
4) If the intersection is non-empty, the edge server asks the kid-client to send the message to the edge for further checking with the AI function.
5) If the result of AI checking shows that the message is appropriate, it will be forwarded to recipients.
6) Otherwise, based on the content of the message, the edge picks a smart response (e.g. a pop up message to encourage children to be kind, avoid bullying, etc.) that informs the kid-client that the message should be blocked and sends the smart response to the kid-client. Then, the kid-client shows the smart response to the child, and the protocol ends.

An AI classifier can detect potentially harmful content of a message in real-time, and therefore, we only report the computation time of the PSI protocol. In the following we present one example of a PSI protocol that can be used in step 2 of the on-line phase of the protocol. We assume that there is a way to interpret the words as integers smaller that $p$, for example, using a hash function.

1) Every edge server picks a random integer $b$, a prime modulus $p$, and computes $B' = \{b_i^b \mod p\}$.
2) The kid-client asks its assigned edge server (e.g. Edge Server 1) for its $B'$ list, and the modulus $p$.
3) The kid-client computes set $M'$, by picking a random integer $a$ and computing $M' = \{m_j^a \mod p\}$. Then, the kid-client sends $M'$ to the edge server.
4) The kid-client computes $B^{''} = \{b_i^{ba} \mod p\}$, and sends it to the edge.
5) The edge server computes $M^{''} = \{m_j^{ab} \mod p\}$.
6) Now, the edge server can compute the intersection between $B^{''}$ and $M^{''}$. If this intersection is empty then the intersection between $B$ and $M$ is empty.

If the modulus $p$ is 1024-bits long, computing each entry in $\{m_j^{ab} \mod p\}$ and $\{b_i^{ba} \mod p\}$ takes approximately 0.008 seconds. If there are 100 words in the deny list of the edge server, computing $\{b_j^{ab} \mod p\}$ takes 0.8 seconds. If there are 30 words in the child's message, computing $\{w_j^{ab} \mod p\}$ takes 0.24 seconds.

### G. Sending a Message with Personalized Check

In this section, we present a protocol for sending a message where the parent can also influence their child's on-line activities with the personalized check.

Similarly to the general check for sending a message, we want to check the child's message against a set of forbidden words. In the personalized check in addition to set $B$ of the Edge Server 1, we have another set of forbidden words, $L$, which is provided by the parent. We require a PSO protocol that computes whether $M \cap (L \cup B)$ is empty, and only a third party (e.g. Edge Server 2) should learn this outcome.

We also need two PMT protocols, between an item provided by the edge server and two sets that belong to the parent-client. The sets are an allow list of contacts (phone numbers) that are always trusted, and a deny list of contacts that should be always blocked. The recipient of the child's message is the item that should be checked against the allow and deny sets of the parent-client.

The off-line phase of the protocol is as follows:

1) The core network collects a deny list of harmful words $B = \{b_i\}$.
2) UPF broadcasts this list to all the edge servers.
3) The parent-client creates three lists: an allow list of contacts (phone numbers) that are always trusted, a deny list of contacts that should be always blocked, and a list of forbidden words $L = \{l_k\}$.

In the on-line phase of the protocol, first the recipient of the child's message $M$ is checked. In order to privately determine whether the recipient is in the allow (deny) list of contacts, we require to perform a PMT protocol. The protocol is between the edge, which has the receiver that child wants to connect to, and the parent-client, which has the lists of allow/deny contacts that the parent provided. The kid-client learns the outcome of the PMT protocol. If the receiver is in the allow list, the kid-client gives permission to the message to be sent

through another edge server (e.g. Edge Server 2 in Fig. 1). If the receiver is in the deny list, the kid-client blocks the message.

If the receiver is not in any of the lists, a PSO protocol will be executed to learn the cardinality of $M \cap (L \cup B)$. The result of the PSO should be only learned by another edge server (e.g. Edge Server 2). If $M$ is empty, Edge Server 2 informs the kid-client that the message is benign and can be forwarded via Edge Server 2. Otherwise, the message is blocked and kid-client shows a smart response to the child.

To implement our protocol, we use the protocols of Ramezanian et al. [21]. We use their PSO protocol with hash functions, to learn the cardinality of $M \cap (L \cup B)$.

For our PMT protocols, we use their PSI protocol with keyed hash function, and one of the sets is a singleton.

We assume that there are 20 contacts in allow (deny) list of contacts. Computing the hash values of the contacts in the allow and deny lists together takes 0.04 ms. If there are 100 words in $B$ ($L$), the Edge Server 1 (parent-client) needs 0.1 ms to compute the hash values of the words in $B$ ($L$). Finally, if the child's message contains 30 words the kid-client needs 0.03 ms to compute the hash values of the words in $M$.

Table II show the communication complexity of sending messages via our parental control protocol.

### H. Receiving a Message

Our parental control protocol functions as follows for the case where a child receives a text message.

As we explained earlier, all the child's messaging activities are done via the kid-client. Therefore, it is the kid-client that receives the message. The kid-client asks the network whether the sender is a child or not. If the sender of the message is a child in a network that has parental control functionality, the message has been checked before it got permission to be forwarded. Therefore, this message is benign and the kid-client lets the message to be shown on the child's UE.

If the sender is an adult, or the sender is a child in a network without parental control, the kid-client acts as if this is a message that the child wants to send. In other words, the kid-client starts the protocol of Subsection $F$, and if the child is under personalized check, then the kid-client starts the protocol of Subsection $G$. The allow/deny lists of contacts that have been presented in Subsection $G$ can be used here as well to check whether the sender's phone number is in any of these lists.

If after checking the message, it turns out that it is benign, the kid-client lets the message to be shown to the child. Otherwise, the kid-client blocks the message. There is no need to generate a smart response, instead, the parent or the authorities will be notified about the harmful messages and their senders.

### I. Impact on Standardization

It is clear that standardized 5G network elements have a crucial role in our protocol. On the other hand, importance of parental control has been recognized by 3GPP [7]. Because of these reasons, including support for our protocol or its derivative would be beneficial in future releases of 3GPP specifications. From operational point of view, one of the smoothest deployment option is where both kid-client and parent-client are provided by the mobile network operator to their subscribers. In this case all necessary interfaces are fully controlled by the operator and the whole protocol could be implemented in proprietary way. However, it would be useful to allow other deployment scenarios, e.g., the client side could be developed by third parties. In that case, at least some amount of standards support would be necessary to guarantee interoperability.

## VI. Privacy analysis

In this section, we present the privacy analysis of our parental control protocol. We assume that all parties follow the protocol as they are supposed to, i.e., we apply the honest-but-curious adversary model. We also require that different edge servers do not communicate with each other in regards of content that are obtained during parental control process. In other words, the edge servers do not share information about the data they obtained from children.

As a prerequisite for our protocol, the operator and parent sign a legally binding contract, where they both promise to follow the protocol honestly. The operator is responsible to identify the birthday of the subscribers that are labelled as children. Therefore, when a subscriber reaches a certain age, they will automatically be removed from digital parental control.

In our protocol, the network is responsible to define proper parental control policies for its subscribers. This prevents malicious users from utilizing our protocol for something else, rather than parental control. For example, our protocol cannot be manipulated such that a malicious subscriber uses it to control the employees, spouse, elderlies, etc. However, if the network is malicious it can use our protocol for sinister use-cases such as content censorships. Please note that if the network is malicious, it probably can cause even more harm to its subscribers than what is achievable with using our protocol in a malicious way. For example, the network can only allow the usage of a certain messaging application that does not use end-to-end encryption. Then, the malicious network is able to monitor, modify and delete, all their subscribers' messages and calls.

One of the goals of our protocol is to preserve the privacy of the child towards their parent. In our protocol, neither the parent nor the parent-client receive any information about the child's (kid-client's) activities. In other words, the website that the child requested, or the message and its receiver remain private towards the parent. Therefore, the privacy of the child is preserved towards their parent.

As the network is responsible to move the data traffic of child's UE, it will anyway learn the recipient of the message and the requested URL. However, for the personalized check, by utilizing more than one independently operating edge server to perform the parental control service, we ensure that the

TABLE I. COMMUNICATION COMPLEXITY FOR ACCESSING WEBSITES WHEN UTILIZING THE PSI PROTOCOL OF RAMEZANIAN ET AL., WHEN THE SET OF ATTRIBUTES ($U$) HAS 20 ENTRIES, AND $N^2$ IS 4096 BITS LONG.

| Type of the service | Parent-client to edge | Kid-client to edge and parent-client | Edge to kid-client | Kid-client to edge |
|---|---|---|---|---|
| General check | - | - | Size of the content of a website | One URL |
| Personalized check | 10 KB | 0.25 KB | Size of the content of a website + 10 KB | One URL or two URLs |

TABLE II. COMMUNICATION COMPLEXITY FOR SENDING A MESSAGE WHEN UTILIZING THE PSO PROTOCOL BY RAMEZANIAN ET AL. WE ASSUME THAT SET $B$ HAS 100 ITEMS AND $M$ HAS 30 ITEMS. WE ALSO ASSUME THAT THE DENY/ALLOW LIST OF CONTACTS EACH HAVE 20 ITEMS. THE PARENT'S LIST OF FORBIDDEN WORDS HAS 100 ITEMS AS WELL.

| Type of the service | Parent-client to edge | Edge to kid-client | Kid-client to edge |
|---|---|---|---|
| General check | - | 25 KB | 32.5 KB |
| Personalized check | 140 hash values | Results of the PSO protocols (empty or not) | 31 hash values + actual message |

network will not learn whether the message/the requested web-site was benign or not.

When child sends the message via general check, the kid-client and the edge server execute a PSI where the result, whether the intersection is empty or not, is learned by the edge server. Because of the properties of PSI, the kid-client will not learn the forbidden words. If the PSI raises a red flag, then the edge requests the message for further checking with AI. Therefore, the edge server does not learn the words in the message, unless the result of the PSI protocol is not empty, which indicates that the message contains potentially harmful words.

We also aim to preserve the privacy of the parent towards their child and the network. Next, we explain how the parent's activity remains private. Let us first consider the use-case of accessing a website with personalized check: A PSI is executed between the edge server, the parent-client and the kid-client. Because of the properties of the PSI, the edge server and the kid-client do not learn the set of forbidden attributes of the parent-client ($W_p$). Neither the parent-client nor the the kid-client learn the attributes that the edge server has found ($W_c$). The kid-client learns only whether the intersection $W_p \cap W_c$ is empty or not. The parent-client and the edge server do not learn even this. Therefore, the parent's set of forbidden attributes remains private. The child cannot learn anything else except that certain website is blocked. However, if two or more kids try to access a website from their devices, they might learn some information about their parents' restrictions. For example, let us assume that two kids of the same age together try to access a website with their devices. One kid may observe that he can access the site, whereas the other kid sees that she is redirected to another web-site. In this scenario, the kids learn that this difference in the result of their requests was caused by their parents, and not by the policies defined by the network.

In our protocol when a child sends a message with personalized check, a more complicated PSO is used, together with a PMT. The PSO calculates whether the intersection of the words in the message and the words in the union of forbidden words of both the parent-client and the edge server is empty or not. The Edge Server 2 learns the result of this, but nothing else about the set of the kid-client and the parent-client, because of the PSO. The parent-client and the Edge Server 1 learn nothing about the other parties sets. The kid-client and the Edge Server 2 learn whether the intersection is empty or not. The privacy of the allow and deny lists of the parent client is guaranteed by the properties of PMT, and only the kid-client (not the child) learns whether the intersection is empty or not. The privacy of the words in the message is guaranteed by the PSO, and only the Edge Server 2 learns whether the intersection is empty or not.

In the personalized check to access a website, there is a set of possible attributes that can be found in a website and the parent can only choose restrictions from that set. Therefore, it is not possible for a parent to enforce random restrictions on their child.

The child cannot bypass the personalized check because, they do not have access to the parent-client nor the edge server. Moreover, the child cannot bypass the generalized check because the edge server is not accessible for the child.

Our protocol provides similar protection for all children via general check.

In the use-case where a general/personalized check is done for accessing a web-page, the kid-client connects to the proxy server in the edge. If the web-page is already stored in the cached web page database of the proxy, the edge server does not need to connect to the server to get the content of the web page. Otherwise, the proxy connects to the server which is hosting the web-site. After executing this process, the server has no clue about the user whom the edge server requested the web-site for. Therefore, using the proxy hides the identity of the child. Moreover, the child's interests, on-line behaviour and exact location remain private to the service providers.

We assumed that our parties are honest-but-curious. How-

ever, our protocol provides some protection when one edge server is malicious. For instance, if the Edge Server 1 is malicious and wants to know whether something in the parent's list of forbidden attributes blocked the child from accessing a web-site, it should collude with at least one other edge server in the network. This is due to the fact that only the kid-client gets the results of the PSI protocol, and sends the original URL or the replaced URL to another edge server than Edge Server 1.

Now, let us assume a common attack to a two-party PSI protocol between parties $A$ and $B$. In this attack, one party (party $A$) uses a set which only has one element. After executing the PSI protocol, party $A$ will learn whether that element is in the set of the other party (party $B$). After repeating this process for many times, party $A$ can learn the set of party $B$. However, this attack is not effective in our setting, because it is always the kid-client that gets the result of the PSI protocol, and the edge server and the parent-client remain oblivious about this result. However, if two curious kids put only one word in their messages, and observe that one of them gets a smart response, then that kid learns that the word in the message is in the forbidden list of their parent.

In our personalized check, the network uses a set $U$, which contains the attributes that need to be checked in the packets. Utilizing a unified set of attributes prevents the parents from putting arbitrary restrictions on their child. For instance, let us assume that a parent wants to forbid their child from watching a certain benign cartoon, for whatever reason. As this cartoon is kid-friendly program, it does not have any attribute that can be found in $U$. Therefore, the parent cannot prevent their child from watching this certain cartoon by utilizing our parental control protocol.

## VII. Conclusion

Nowadays, children are being exposed to the on-line world at the very young age. With the emerge of 5G networks, it is probable that children would spend even more time in the digital world. On the other hand, at the time of writing, most of the parental control services are available only for children whose parents are willing to pay the extra fee for this service. In this paper, we propose a privacy preserving parental control protocol for children. To the best of our knowledge, our proposal is the first privacy preserving parental control protocol.

Our protocol uses edge computing and AI methods to analyze the data as fast as possible. Moreover, we use privacy preserving set operation protocols to insure privacy for parents and their children. We designed our protocol in such a way that the parent's privacy is preserved towards the network and the child. The child's privacy is also preserved towards the network, and the parent. Moreover, the results of our implementations show that our protocol is feasible in practice.

In our protocol, we provide two types of checking: the general check which is done for all children who have access to Internet, and the personalized check which is done for the children whose parents want to be involved with the parental control process. The general check brings equality for the on-line experience that children have. The general check guarantees safe surfing over the Internet, even in the case where the child's guardian might be neglectful (or not present) in supervising the child's on-line activity. With the personalized check, parents can influence their children's on-line activities in a privacy preserving way.

Our protocol provides parental control over accessing a web-site and sending/receiving text messages. One direction for future work is to extend our protocol such that it provides parental control for multimedia messages.

## Acknowledgment

We thank the anonymous reviewers of the 29th Conference of Open Innovations Association FRUCT, for their insightful comments and suggestions on this paper. This paper is supported by 5GFORCE project funded by Business Finland.

## References

[1] Jonathan Y Bernard, Natarajan Padmapriya, Bozhi Chen, Shirong Cai, Kok Hian Tan, Fabian Yap, Lynette Shek, Yap-Seng Chong, Peter D Gluckman, Keith M Godfrey, et al. Predictors of screen viewing time in young singaporean children: the gusto cohort. *International Journal of Behavioral Nutrition and Physical Activity*, 14(1):112, 2017.

[2] Neza Stiglic and Russell M Viner. Effects of screentime on the health and well-being of children and adolescents: a systematic review of reviews. *BMJ open*, 9(1), 2019.

[3] Pamela Wisniewski, Haiyan Jia, Na Wang, Saijing Zheng, Heng Xu, Mary Beth Rosson, and John M Carroll. Resilience mitigates the negative effects of adolescent internet addiction and online risk exposure. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, pages 4029–4038, 2015.

[4] Sonia Livingstone, Mariya Stoilova, and Rishita Nandagiri. Children's data and privacy online: growing up in a digital age: an evidence review. 2019.

[5] Pamela Wisniewski. The privacy paradox of adolescent online safety: A matter of risk prevention or risk resilience? *IEEE Security & Privacy*, 16(2):86–90, 2018.

[6] Loredana Benedetto and Massimo Ingrassia. Digital parenting: Raising and protecting children in media world. In *Parenting*. IntechOpen, 2020.

[7] 3GPP. Service requirements for the evolved packet system (eps). https://www.3gpp.org/ftp/Specs/archive/22_series/22.278/, 2019. Accessed: 2020-08-17.

[8] Arup Kumar Ghosh, Karla Badillo-Urquiola, Shion Guha, Joseph J LaViola Jr, and Pamela J Wisniewski. Safety vs. surveillance: what children have to say about mobile apps for parental control. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, pages 1–14, 2018.

[9] Lindsay Blackwell, Emma Gardiner, and Sarita Schoenebeck. Managing expectations: Technology tensions among parents and teens. In *Proceedings of the 19th ACM Conference on Computer-Supported Cooperative Work & Social Computing*, pages 1390–1401, 2016.

[10] Jelena Gligorijević. Childrens privacy: The role of parental control and consent. *Human Rights Law Review*, 19(2):201–229, 2019.

[11] Weisong Shi, Jie Cao, Quan Zhang, Youhuizi Li, and Lanyu Xu. Edge computing: Vision and challenges. *IEEE internet of things journal*, 3(5):637–646, 2016.

[12] Yen-Lin Lee, Pei-Kuei Tsung, and Max Wu. Techology trend of edge ai. In *2018 International Symposium on VLSI Design, Automation and Test (VLSI-DAT)*, pages 1–2. IEEE, 2018.

[13] Intel neural compute stick 2. https://ark.intel.com/content/www/us/en/ark/products/140109/intel-neural-compute-stick-2.html, 2018. Accessed: 2020-08-17.

[14] IBM. Ibm edge application manager. https://www.ibm.com/fi-en/cloud/edge-application-managerl, 2020. Accessed: 2020-08-17.

[15] Evelina Pencheva, Ivaylo Asenov, Ivaylo Atanasov, and D Ventsislav Trifonov. Programmability of policy control at the edge of the mobile network. In *2020 International Symposium on Networks, Computers and Communications (ISNCC)*, pages 1–6. IEEE, 2020.

[16] Ivaylo Atanasov, Evelina Pencheva, Aleksandar Nametkov, and Ventsislav Trifonov. On functionality of policy control at the network edge. *International Journal on Information Technologies and Security*, 3(11):3–24, 2019.

[17] 3GPP. 3gpp ts 23.501 system architecture for the 5g system. https://www.3gpp.org/ftp/Specs/archive/23_series/23.501, 2020. Accessed: 2020-08-17.

[18] Michael J Freedman, Kobbi Nissim, and Benny Pinkas. Efficient private matching and set intersection. In *International conference on the theory and applications of cryptographic techniques*, pages 1–19. Springer, 2004.

[19] Sara Ramezanian, Tommi Meskanen, Masoud Naderpour, Ville Junnila, and Valtteri Niemi. Private membership test protocol with low communication complexity. *Digital Communications and Networks*, 2019.

[20] Tommi Meskanen, Jian Liu, Sara Ramezanian, and Valtteri Niemi. Private membership test for bloom filters. In *2015 IEEE Trustcom/BigDataSE/ISPA*, volume 1, pages 515–522. IEEE, 2015.

[21] Sara Ramezanian, Tommi Meskanen, and Valtteri Niemi. Multi-party Private Set Operations with an External Decider. *arXiv e-prints*, page arXiv:2103.08514, March 2021.

[22] Eric Rescorla and Tim Dierks. The transport layer security (tls) protocol version 1.3. 2018.

[23] eSafety. How to use parental controls and other tools to maximise online safety in your home. https://www.esafety.gov.au/parents/skills-advice/taming-technology. Accessed: 2020-08-17.

[24] Walter Fuertes, Karina Quimbiulco, Fernando Galárraga, and José Luis García-Dorado. On the development of advanced parental control tools. In *2015 1st International Conference on Software Security and Assurance (ICSSA)*, pages 1–6. IEEE, 2015.

[25] Brian K Barber, Heidi E Stolz, Joseph A Olsen, W Andrew Collins, and Margaret Burchinal. Parental support, psychological control, and behavioral control: Assessing relevance across time, culture, and method. *Monographs of the society for research in child development*, pages i–147, 2005.

[26] Rita Brito, Rita Francisco, Patrícia Dias, and Stephane Chaudron. Family dynamics in digital homes: the role played by parental mediation in young childrens digital practices around 14 european countries. *Contemporary Family Therapy*, 39(4):271–280, 2017.

[27] Lynn Schofield Clark. Parental mediation theory for the digital age. *Communication theory*, 21(4):323–343, 2011.

[28] Hamza HM Altarturi, Muntadher Saadoon, and Nor Badrul Anuar. Cyber parental control: A bibliometric study. *Children and Youth Services Review*, page 105134, 2020.

[29] Mahdi Hashemi. Web page classification: a survey of perspectives, gaps, and future directions. *Multimedia Tools and Applications*, pages 1–25, 2020.

[30] Kejing He and Chenyang Li. Structure-based classification of web documents using support vector machine. In *2016 4th International Conference on Cloud Computing and Intelligence Systems (CCIS)*, pages 215–219. IEEE, 2016.

[31] Abrar Noor Akramin Kamarudin and Bali Ranaivo-Malançon. Simple internet filtering access for kids using naïve bayes and blacklisted urls. In *International Knowledge Conference*, 2015.

[32] Sungjin Kim, Jinkook Kim, Seokwoo Nam, and Dohoon Kim. Webmon: Ml-and yara-based malicious webpage detection. *Computer Networks*, 137:119–131, 2018.

[33] Sara Ramezanian and Valtteri Niemi. Privacy preserving cyberbullying prevention with ai methods in 5g networks. In *2019 25th Conference of Open Innovations Association (FRUCT)*, pages 265–271. IEEE, 2019.

[34] Zichuan Liu, Yixing Li, Fengbo Ren, Wang Ling Goh, and Hao Yu. Squeezedtext: A real-time scene text recognition by binary convolutional encoder-decoder network. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[35] G Vinodhini and RM Chandrasekaran. Sentiment analysis and opinion mining: a survey. *International Journal*, 2(6):282–292, 2012.

[36] Anastasia Giachanou and Fabio Crestani. Like it or not: A survey of twitter sentiment analysis methods. *ACM Computing Surveys (CSUR)*, 49(2):1–41, 2016.

[37] Qian Luo, Jiajia Liu, Jiadai Wang, Yawen Tan, Yurui Cao, and Nei Kato. Automatic content inspection and forensics for children android apps. *IEEE Internet of Things Journal*, 2020.

[38] Anjuli Kannan, Karol Kurach, Sujith Ravi, Tobias Kaufmann, Andrew Tomkins, Balint Miklos, Greg Corrado, Laszlo Lukacs, Marina Ganea, Peter Young, et al. Smart reply: Automated response suggestion for email. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 955–964, 2016.

[39] Ari Luotonen and Kevin Altis. World-wide web proxies. *Computer Networks and ISDN Systems*, 27(2):147–154, 1994.