Detection of Images with Adult Content for Parental Control on Mobile Devices[*]

Giuseppe Amato
Istituto di Scienza e
Tecnologie dell'Informazione
"A. Faedo" - C.N.R.
Pisa, Italy
giuseppe.amato@isti.cnr.it

Paolo Bolettieri
Istituto di Scienza e
Tecnologie dell'Informazione
"A. Faedo" - C.N.R.
Pisa, Italy
paolo.bolettieri@isti.cnr.it

Gabriele Costa
Istituto di Informatica e
Telematica - C.N.R.
Pisa, Italy
gabriele.costa@iit.cnr.it

Francesco la Torre
Istituto di Informatica e
Telematica - C.N.R.
Pisa, Italy
francesco.latorre@iit.cnr.it

Fabio Martinelli
Istituto di Informatica e
Telematica - C.N.R.
Pisa, Italy
fabio.martinelli@iit.cnr.it

ABSTRACT

In this paper we present a prototype for parental control that detects images with adult content received on a mobile device. More specifically, the application that we developed is able to intercept images received through various communication channels (bluetooth, MMS) on mobile devices based on the Symbian^{TM} operating systems. Once intercepted, the images are analysed by the component of the system that automatically classify images with explicit sexual content. At the current stage the application that intercept images runs on the mobile device, the classifier runs on a remote server.

Categories and Subject Descriptors

I.5 [**Pattern Recognition**]: Miscellaneous

General Terms

Security

Keywords

Mobile, Security

1. INTRODUCTION

One of the most challenging security issues regarding mobile devices is parental control. Modern devices can receive large multimedia contents, e.g. high-definition pictures or video streaming, and process them. For instance, figure 1 shows the appearance of a modern smartphone MMS interface. Despite their increasing computational capabilities, mobile phones provide little or no protection against undesired contents exchange. This aspect becomes crucial when the user of the mobile phone is a minor. While the inspection of transferred contents is currently possible for messages moving through the infrastructured part of the network, the analysis of locally exchanged messages is still infeasible. Short range, point-to-point connections, for instance using bluetooth or IR links, create small, temporary networks providing no guarantees on the sent data.

Figure 1: A typical MMS interface

It seems clear that the devices must equip some self-protection mechanism in order to safeguard young users. Such mechanism must be effective, i.e. no undesired data can trespass or avoid it, and tamper-proof, i.e. no accidental or aimed attack can interfere with its execution. Moreover, it should not reduce the system usability, e.g. creating long, user-noticeable delays or interfering with safe functionalities.

Classifiers can recognize and discriminate between harmless and offensive multimedia contents. However the complexity of such

[*] This work was partially supported by the European project EU-FET-IP Emergent Connectors for Eternal Software Intensive Networked Systems (CONNECT).

systems discourages from implementing and running them on small mobile devices. On the other hand, every device has the connectivity capabilities necessary for sending and receiving relatively rich amount of information.

Starting from this consideration, we planned to implement the contents verification system as the composition of two, independent entities. From one side, on the small device, we have the enforcement architecture. Its goal is to capture the images before they can be accessed by the user and manage them depending on a classification score. On the other side, a server runs the classification engine that provides the classification score to the users that ask for it.

In this way, we aim to have a complete, effective content management system distributed in such a way that the heavier computation takes place on hardware with no severe limitations and, at the same time, the costumer's device holds a complete control on the enforcement mechanism.

2. RELATED WORK

Much work has been done for securing mobile devices. Their increasing potential and the critical information they have access to, attract more attention day by day. Almost every new mobile phone that comes on the market, offers some protection mechanism. Those mechanisms change from device to device depending on the manufacturer and the software/hardware configuration. For instance, is common practice to deliver to the user the responsibility to decide whether some suspicious action is permitted or not. This approach has one evident shortcoming: it assumes that the user can always distinguish between a valid operation and a dangerous one. Clearly, this requirement is in contrast with the parental control scenario. Indeed, if the actual user is a minor, we can not expect that he/she can decide autonomously on the correctness of what is happening, otherwise the parental control would be unnecessary.

Many authors, e.g. [9, 4], proposed monitoring architectures for enforcing security properties on mobile devices. These mechanisms ensures that the actual usage of the device always follows a given security policy. Such policy can be defined by the owner, e.g. parents, and all the users, e.g. children, are under its scope. In this way, some interesting properties can be defined and a real parental control can be implemented. However, these frameworks are specifically built for enforcing security properties on executing applications and they do not provide any particular control on data used or received. From this point of view, not only our system does not interfere with the currently available security mechanisms, but can also be integrated with them so to obtain a richer set of enforceable security properties.

In [6] the authors introduce WebGuard, an automatic filter for explicit web contents. Roughly, WebGuard exploits images and other information, e.g. textual contents and a shared knowledge of suspicious sites, to identify dangerous URL. This system aims to detect and publish on a list those sources that can provide adult contents to minors. Our prototype works under completely different conditions. Indeed, using connected devices, it is not always possible to obtain and share the identity of malicious contents providers. Moreover, our classification tool can work on images with no supplementary information.

Zeng et al. [11] present Image Guarder, a tool for analysing and recognizing pornographic images. Like in the previous case, this instrument is specifically designed for web contents. Moreover, it provides a binary output as a result of the classification. On the contrary, our classifier rates the image content allowing the user for applying more expressive policies.

The major attention received by parental control issues is also proved by the presence on the market of many commercial tools. However the main products for filtering sexual contents seem to be still far from a definitive solution. For instance, AdaptiveMobile [1] proposes a filter preventing mobile phones from receiving undesired MMS. However, this filter works on provider-side. This fact has at least two main drawbacks. First, it can not be tuned on the real necessity of every, single phone/user. Indeed, our approach allows for a wider class of definable properties that can depend on both the message and the current state of the device. Secondly, being not transmitted through the infrastructured part of the network, data sent using local connections, e.g. bluetooth and IRs, is completely out of control.

3. COMPLETE ARCHITECTURE

The three entities composing the system are the messages interceptor, the contents manager and the classifier. In the current implementation, the first and second component lay on the mobile device, while the classifier runs on a server waiting for clients requests.

The interceptor is implemented as a daemon running in background. Since no assumptions can be made on the time elapses between two consecutive received messages, this component must be kept free from any time-consuming computation. The interceptor carries on asynchronous communication with the contents manager. The first inserts requests in a shared queue while the second extracts and processes the waiting elements.

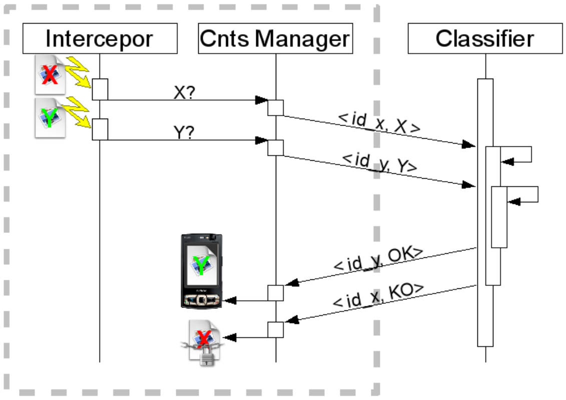

Figure 2: Interactions among the system components

The contents manager communicates with the remote classifier opening a socket connection. Then a triple $\langle id, s, v\rangle$, where id is a unique identifier and v is a vector of s bytes representing the image, is sent. The server answers with a pair $\langle id, n\rangle$ where id is the same as for the request and n is the computed score. As the classification may require some time, the messages exchange follows a sliding window protocol. Hence the client sends a sequence of messages without waiting passively for the corresponding answers. When a response packet arrives, the identifier is extracted and the received score is associated with the corresponding image.

4. IMAGE FILTER ON THE MOBILE DEVICE

During the execution, the interceptor keeps in an idle state listening for messaging events dispatched by the operating system. These events are fired whenever the device receives a new message. Since messages can be dispatched through different channels, e.g. bluetooth and MMS, the application must analyse the corresponding signals and access to the correct locations where the information has been stored. Generally, received messages are placed in a proper position in the file system that is called *inbox*. Replacing the default system handler, our interceptor accesses the inbox before the new data is made visible to the user and invokes the contents manager. Subsequently the listening state is restored so to catch any possible further message.

As the contents manager is called, it immediately hides the message to the user. In this way the user is prevented from accessing data that has not been classified yet. Indeed the asynchronous interaction between the interceptor and the manager can lead to a small delay during which users or applications could exploit for reading the uncategorised contents. Actually the massages waiting for being analysed are moved to a private, hidden directory, but other strategies are also viable, e.g. saving an encrypted copy. After this step, the application creates a direct connection with the classifier and sends the image to be analysed. When the classifier completes its computation, the resulting score is sent back to the mobile device that evaluates it depending on the current security policy, e.g. the score must be less than a given threshold. If the score complies with the active policy the content that was previously hidden is made visible to the user like if it is just arrived. Otherwise the content is kept definitively out of user's reach. The current prototype simply deletes it but we could decide that the offending message is stored for being available for the inspection of some authority, e.g. a parent.

The overall behaviour of the system results completely invisible to the user. Indeed the verification process prevents any signal informing the user that a new message has been received. In this way, our tool does not interfere with the normal exchange of legal contents but for a small delay noticeable from the recipient.

5. IMAGE CLASSIFICATION

In order to perform automatic classification of images having sexual content we defined a binary classifier, that is a classifier that basically distinguishes between images belonging to a class (images with sexual content in this case) and images that do not belong to the class. In order to define the classifier we used the Support Vector Machine (SVM) [3] technology applied to images described using MPEG-7 visual descriptors [7]. In the following we will discuss these aspects in more details. We decide to use the SVM technology due to its potential flexibility in being adapted to specific applications, given by the number of possible kernels to be used to characterize the feature space and the parameters that can be set for fine tuning. In addition, the decision function of an SVM is conceptually very simple and can be easily implemented on a mobile device.

6. DEFINITION OF THE CLASSIFIER

The SVM builds classifiers by learning from a training set that is composed of positive and negative examples.

The training set is denoted as $T = \{(\mathbf{x_1}, y_1), ..., (\mathbf{x_n}, y_n)\}$ where $\mathbf{x_i}$ is a convenient vector representation, which we will discuss later, of the image i in the training set. Each y_i is 1 or -1 according to the fact that the training image i is a positive or negative sample of the class to learn.

6.1 Support Vector Machine

Technically, a SVM finds a hyperplane that divides the space in two subspaces. The simplest case is that of the linear SVM, where the learning phase determines a hyper-plane that divides the space in two subspaces. One subspace corresponds to images belonging to the class, the other images that do not belong. In this case, omitting several theoretical details (see [3] for more information), the learning phase has to find a vector ω such that the decision function

$$f(\mathbf{x}) = <\omega, \mathbf{x}> +b$$

is able to optimally classify most of the training set examples.

In most cases it is not possible to linearly separate negative from positive examples, so a linear SVM is not effective. However, typically, linear separation is still possible by mapping the vectors in a higher dimensional space.

Suppose you have a mapping function that maps vectors in a space of large (even infinite) dimensions where a linear separation can be found between positive and negative examples. In this case the decision function can be written as

$$f(\mathbf{x}) = <\omega, \Phi(\mathbf{x})> +b$$

Actually, SVM methods do not define the mapping function explicitly, but use the properties of the kernel functions.

A kernel function K, defined as $K(\mathbf{x}_i, \mathbf{x}_j) = <\Phi(\mathbf{x}_i)^T, \Phi(\mathbf{x}_j)>$, computes the dot-product of vectors obtained from the mapping of $\mathbf{x}_i$ and $\mathbf{x}_j$. There are simple kernel functions that easily compute the dot-product of vectors mapped in very high or even infinite dimensional spaces without even knowing the actual mapping functions.

It can be proven that the kernel based decision function, defined above, can be also represented in the dual form as

$$f(\mathbf{x}) = \sum_{i=1}^{n} y_i \alpha_i K(\mathbf{x}, \mathbf{x}_i) + b$$

in terms of the training vectors. In this formulation, the learning phase consists in finding the parameters α_i (which basically determine the contribution of each example of the training set to the solution of the learning problem) rather than the vector ω.

We have used the kernel adatron to solve the classification problem. The adatron was first introduced in [2] as a perceptron-like procedure to classify data and then a kernel-based was proposed in [5]. The learning strategy of the adatron is to perform a gradient ascent to solve the margin-maximization problem between the positive and negative examples of the training set.

It is important to stress that the learning phase has to be performed just once for all and can be executed on powerful computers. The result of the learning phase is just a vector of weights (most of which are 0) that is used by the decision function to decide the nature of any analysed image.

6.2 Tuning consideration

The performance of the SVM classification is strongly related to the choice of the kernel function, the kernel parameters, and some parameters related to the adatron algorithm, such as the penalty parameter C, the maximal tolerance on the margin, and maximum number of iterations.

There is a large number of kernel functions available. We have chosen the Gaussian Radial Basis Function (RBF) given its capability to recognize separate areas of the vector space where positively classified elements can be found. The RBF takes as parameter the variance σ of the underlying Gaussian.

In general, it is not not possible to know in advance the parameters that offer the best performance for a specific problem. A widely used procedure, to chose optimal parameters, is the v cross-validation. This procedure divides the training set into v different subsets (folders) of equal size: one folder is used as test set for the classifier and the other $v - 1$ folders are used as training sets. The cross-validation process is then repeated v times using every time a new folder of the v subsets as test set. Finally the v results obtained can be averaged to produce an overall evaluation of the classification system.

In order to automatically find the optimum values for the penalty parameter C and the variance σ of the RBF, one parameter is kept fixed and the other parameter grows exponentially and vice versa. After identifying the best pairs, using cross validation, we performed a finer search on their neighbourhood. Once we have determined the best choice for (C, σ), we have also tested some possible values to find the best maximal tolerance on the margin and maximum number of iterations.

7. IMAGE INDEXING AND IMAGE COMPARISON

As previously stated, the classifier works on a convenient representation of images, in terms of mathematical descriptions of the visual feature of the images, rather than the original raw images themselves. The basic characteristics of an image representation is the possibility be used to mathematically measure the similarity between images. Typical representations of images includes, color histograms, textures, shapes, etc. The process of building representations of images is called image indexing.

In our application we have used the MPEG-7 standard to represent the visual features of images. More specifically we have used the MPEG-7 image Visual Descriptors defined in [7] using a feature extractor built upon the MPEG-7 experimentation model (XM, [8]) of MPEG-7 Part 6: Reference Software. We extract 5 MPEG-7 descriptors: ScalableColor (a color Histogram in the HSV Color Space), ColorStructure (captures both color content and information about the spatial arrangement of the colors), ColorLayout (represents the spatial layout of color images), EdgeHistogram (spatial distribution of five types of edges), and HomogeneousTexture (characterizes the properties of texture in an image).

Image size	Query session	Image processing	%
66 Kb	9047 ms	8094 ms	89
118 Kb	8791 ms	7859 ms	89
169 Kb	8360 ms	7177 ms	86
593 Kb	22081 ms	19062 ms	86

Table 1: Execution times

In order to measure the similarity between two images according to a specific visual descriptor, [10] suggests some distance measures: the shortest the distance the higher the similarity. To have a single similarity measure we perform a linear combination of the individual distance functions. The obtained distance function is used in place of the Euclidean distance in the RBF kernel.

We apply the image indexing process in two different phases: 1) images of the training set have to be indexed before starting with the learning process; 2) images received by the smart phones have to be indexed before being analysed by the classifier.

8. VALIDATION AND PERFORMANCES

In order to validate the prototype we have created a test set consisting of about 400 images with sexual content and 10000 images with no sexual content. These images were clearly not part of the training set. Our classifier was able to to correctly classify more than 90% of the images with sexual content and also above 90% of the images with no sexual content. In other words, 10% of the images with sexual content where erroneously classified as non offensive images, and also 10% of the non offensive images were erroneously classified as offensive.

Consider also that the threshold to decide between offensive and non offensive images can be personalized. For instance, by lowering the threshold we can increase the percentage of recognized offensive images. However, this also increase the percentage of non offensive images that are erroneously detected as offensive. As an example, if we use a threshold that recognize 97% of offensive images, we erroneously classify as offensive 21% of non offensive images.

Table 1 summarises the results of our experiments. The second column shows the time interval taken by the whole process (from the message receiving to the message (dis)approval) while the third column contains the classification times. The last value of each row is the percentage of the total time consumed by the server for processing the image.

We tested our framework on a Nokia E-61, equipped with the Symbian OS v9.1, and with the classifying service running on a 2 GHz desktop machine. Clearly, this configurations in not appropriate for a server that, in principle, should be able to process many concurrent requests. However, this experiment proved that our approach is feasible without compromising the overall usability of the mobile device. Indeed, being in the order of few seconds, the produced delays are consistent with the standard messages delivery services since, in general, they provide no time guarantee.

9. CONCLUSION AND FUTURE WORK

We presented a framework for sexual contents filtering on mobile devices. Our prototype intercepts received messages before the user can access them. Images are processed and sent to a remote classifier that rates them. Then, the obtained value is used to decide whether the message can be delivered or not. Our system is designed to work with both data received from infrastructured networks and from temporary, short range connections. Hence, the protection provided is effective on every kind of incoming message. Moreover, it does not reduce the usability but for generating a small delay in the reception of multimedia messages. As the application does not need any manual configuration, it is appropriate for users with no technical skills, for instance a parent.

However, further improvements can be done. Indeed, in order to reduce the computational load of the mobile devices, the classification process takes place on a remote server. This approach offers several advantages deriving from the computational capabilities of the server. On the other hand, we can imagine that the direct access to a remote classifier is not always available or convenient. This problem can be solved applying some pragmatic solution like storing in some protected location the messages not yet inspected and processing them when a suitable connection to the server is available. However, we must observe that the actual computational profiles of the most common mobile devices allows for executing relatively complex tasks. This consideration suggests that a classification algorithm can be implemented and run directly on device. In this sense we are already working on a lightweight classifier. Clearly this approach need to be further studied since, at the best of our knowledge, a similar system is still missing on mobile devices.

Exploiting this classification framework and our security monitoring techniques, see [9], we also plan to create an integrated environment for defining fine-grained, high-level security policies. Using history-based security mechanisms we can define sophisticate, quantitative properties that also cope with parental control issues. For instance imagine properties like “do not play games or music at school” or “do not send more than X SMS”. This new approach can be used to extend our model with qualitative properties like “do not send/receive SMS to/from phones that attempted to share pornographic images”. The composition of these two techniques generates a completely new set of security properties that become available for defining and enforcing top level parental control policies on small devices.

10. REFERENCES

[1] Adaptivemobile home page. `http://www.adaptivemobile.com/`, 2009.

[2] J.K. Anlauf and M. Biehl. The adatron-an adaptive perceptron algorithm. *Europhysics Letters*, vol.10, 1989.

[3] Nello Cristianini and John Shawe-Taylor. *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods*. Cambridge University Press, March 2000.

[4] L. Desmet, W. Joosen, F. Massacci, P. Philippaerts, F. Piessens, I. Siahaan, and D. Vanoverberghe. Security-by-Contract on the .NET platform. *Inf. Secur. Tech. Rep.*, 13(1):25–32, 2008.

[5] Thilo-Thomas Frieß, Nello Cristianini, and Colin Campbell. The Kernel-Adatron algorithm: a fast and simple learning procedure for Support Vector machines. In *Proc. 15th International Conf. on Machine Learning*, pages 188–196. Morgan Kaufmann, San Francisco, CA, 1998.

[6] Mohamed Hammami, Youssef Chahir, and Liming Chen. Webguard: Web based adult content detection and filtering system. *Web Intelligence, IEEE / WIC / ACM International Conference on*, 0:574, 2003.

[7] ISO/IEC. Information technology - Multimedia content description interfaces. Part 3: Visual. 15938-3:2002.

[8] ISO/IEC. Information technology - Multimedia content description interfaces. Part 6: Reference Software. 15938-6:2003.

[9] F. Martinelli, P. Mori, T. Quillinan, and C. Schaefer. A runtime monitoring environment for mobile Java. In *1st International ICST workshop on Security Testing (SecTest08)*, 2008.

[10] Phillipe Salembier and Thomas Sikora. *Introduction to MPEG-7: Multimedia Content Description Interface*. John Wiley & Sons, Inc., New York, NY, USA, 2002.

[11] Wei Zeng, Wen Gao, Tao Zhang, and Yang Liu. Image guarder: An intelligent detector for adult images. In *Asian Conference on Computer Vision*, pages 198–203, Jeju Island, Korea, January 2004.