# Towards developing secure video surveillance systems over IP

Bogdan Groza, Ioan Silea
Politehnica University of Timisoara
Faculty of Automatics and Computers
Timisoara, Romania
bogdan.groza@aut.upt.ro, ioan.silea@aut.upt.ro

Dragos Pop
Continental Corporation
Infotainment and Interpersonal Department
Timisoara, Romania
dragos.pop@continental-corporation.com

Victor-Valeriu Patriciu
Military Technical Academy
Department of Computer Engineering
Bucharest, Romania
vip@mta.ro

***Abstract*— A framework of three attributes for video surveillance systems is underlined: availability, accessibility and authenticity. Under this framework, a scenario in which surveillance cameras can be accessed by remote devices, such as mobile phones, PDAs, over IP is addressed. Some security drawbacks of an off-the-shelf product are depicted and a new solution is proposed which uses cryptographic authentication for the broadcasted images. The proposed application is implemented in Java and can run on any device from standard computers to mobile phones. Also, some experimental results are presented for the case when a mobile phone is used as a receiver, this case being relevant as the device is the potential receiver with the most constrained computational resources.**

***Keywords - video surveillance; availlability; accessibility; authenticity***

## I. INTRODUCTION

Video surveillance has a number of benefits that are indispensable for today's society: it helps to prevent theft and vandalism of public and private places, prevent violence and crime, protect children and even improve customer image and trust on a particular vendor. A website dedicated to all aspects regarding video-surveillance is available at [16].

In the last decade, video surveillance has evolved from analog transmission over coaxial cables to digital transmission over the Internet. In this context, video surveillance raises new challenges and especially from a security perspective.

Nowadays, any end-user can buy almost inexpensive surveillance devices from the market in order to assure the security of his home and business. However, the question that remains is how secured are such off-the-shelf devices. Security can be characterized by a simple equation: the existence of vulnerabilities and adversaries implies the existence of security risks. So the first question that needs to be answered is what adversaries may have malicious interests on a system. It is no doubt that adversaries may vary from robbers to hackers. However, it is quite obvious that combinations between multiple kinds of adversaries, and not a single stand-alone adversary, are the most dangerous. So, an IP camera, such as the DCS-900 camera that we used as a practical example in one of the following sections, can certainly protect users up to some extent against burglars. But if burglars associate with hackers, does the camera offer protection as well? Since we have two adversaries, in the extent of the aforementioned security equation, if we also have vulnerabilities with respect to either of the adversaries then we have a security risk. So the question is what vulnerabilities do IP cameras, which transmit information over the Internet, hide.

Before answering to this question, let us first depict three characteristics that an efficient trustworthy video-surveillance system must have:

a) Accessibility – the system must be accessible by remote devices with different computational resources and communication bandwidth.

b) Availability – the system must respond positive when a particular service is requested.

c) Authenticity – the information that is sent by the system must be authentic (this implies both a secure timeline and a guarantee over the source of the information).

We can also add to these, two more objectives that are needed for the privacy of the video surveillance system: c) access control – the system must be accessible and configurable only by users that have the corresponding rights and d) confidentiality – the information from the system cannot be accessed by illegitimate users. Indeed, legal implications with respect to privacy may arise in practice and these two last objectives may be relevant in some contexts but, more important, they may be irrelevant as long as the first three objectives are not assured. Also, assuring privacy objectives implies the use of secret keys that cannot be shared since if two users know the same secret key they can impersonate each other to the system etc. Further, using distinct keys for each user implies computing different cryptographic operations as well, thus degrading

performance. Therefore, the discussion that follows and the system that is proposed are mainly oriented on the AAA framework. We note that the notion of AAA framework is also used in different contexts with different significance for the three A's. To the best of our knowledge the AAA framework with the significance from this paper is new and it is not our intention to overlap with well established terminology but this abbreviation follows straight forward from the desired security objectives of our system.

Of course there are also other aspects of video surveillance such as pattern recognition, motion detection and tracking etc. [2]. Here we are not concerned with such objectives as we are mainly focused on the security of the transmission. Also, for a complete view on closed-circuit television (CCTV) surveillance a basic book on the subject can be consulted [4]. A survey on encryption techniques used for multimedia contents is available in [5].

The paper is organized as follows. Section 2 is a discussion along the AAA framework: accessibility, availability and authenticity. In section 3 we take a brief account on the DCS-900 IP camera, while in section 4 we describe the system that we propose as a valid response under the AAA framework. Section 5 holds the conclusions of our paper.

## II. DEALING WITH ACCESSIBILITY, AVAILABILITY AND AUTHENTICITY

### A. Accessibility

As accessibility tends to focus on people with disabilities and their right to access public services, we can increase our word power and use this term with focus on computational devices which can have a large spectrum of technical specifications. For example, remote devices can include laptop computers, PDAs, mobile phones etc. and certainly there are major differences between them from the hardware (e.g. memory, CPU) up to the software (e.g. operating system). An accessible service should be usable from any such device. Therefore, accessibility is not itself a security objective; however, more important, it is a desired characteristic for any service provided by some system.

The security implications of this objective are that as computational power tends to be very different on distinct devices, making a service accessible by different kinds of devices imposes certain security restrictions. In tables 1 and 2 we synthesize some experimental results that we get on the computational time required by some devices to compute several cryptographic primitives, time was taken as the mean of several hundreds to thousands runs of the corresponding cryptographic primitive. We used processors with different capabilities from Intel Core 2 to Intel PXA for mobile devices and ARM processors for mobile phones. The rows from table 1 correspond to the following: (1) Intel Core 2 Duo 6800 at 2.66 Ghz (Desktop) (2) Intel Core 2 T 2300 at 1.66 Ghz (Notebook) (3), (4) Fujitsu Siemens Pocket Loox and Asus MyPal A-696 both with Intel PXA270-312Mhz processors (PDA) (5) Nokia 6288 (Mobile Phone). Also we tested different implementations of cryptographic functions from .NET, Java and Bouncy Castle API [10], [13], and in tables 1 and 2 only the most relevant results are shown.

The results show that hash functions and MAC codes can be computed in the order of micro to milli-seconds, and therefore are affordable for our scenario. Still, 1024 bit modular exponentiation with 1024 bit exponents, which is the basic operation of an RSA digital signature, is in the order of seconds which may be too expensive. Also, there seems to be some implementation problem with the Bouncy BigInteger class in C# as modular multiplication is with one order of magnitude more computational intensive then modular multiplication with the BigInteger class in Java, as results from tables 1 and 2 show. However, the verification procedure for a digital signature is more efficient, and in the simplest case may be in the order of a few modular multiplications according to the size of the public exponent.

The advantage in developing a remote access video surveillance system is that while working with a client-server application different platforms can be used for implementation on the client and server side as long as the communication protocol matches. However, there is not a large set of choices for the implementation of the client side which will run on low power devices as well. For example, when we are working on mobile phones, Java may be the only alternative.

As a partial conclusion for accessibility considerations we can state that the computational requirements must be kept at the level of simple cryptographic operations such as hash functions and MAC codes while the implementation must be done in an environment that is independent on the computer architecture, such as Java.

TABLE I. COMPUTATIONAL TIME FOR SOME CRYPTOGRAPHIC PRIMITIVES IN .NET (C#).

| | **SHA256 (Bouncy)** | **SHA256 (.NET framework)** | **1024 Bit Modular Multiplication (Bouncy)** | **1024 Bit Modular Exponentiation (Bouncy)** |
|---|---|---|---|---|
| (1) | $3.1\times10^{-6}\,s$ | $1.87\times10^{-6}\,s$ | $371\times10^{-6}\,s$ | $33.43\times10^{-3}\,s$ |
| (2) | $7.8\times10^{-6}\,s$ | $3.28\times10^{-6}\,s$ | $671\times10^{-6}\,s$ | $60.93\times10^{-3}\,s$ |
| (3) | $0.2\times10^{-3}\,s$ | N/A | $40\times10^{-3}\,s$ | $4.30\,s$ |
| (4) | $0.1\times10^{-3}\,s$ | N/A | $30\times10^{-3}\,s$ | $3.30\,s$ |

TABLE II. COMPUTATIONAL TIME FOR SOME CRYPTOGRAPHIC PRIMITIVES IN JAVA (BOUNCY USED ON THE MOBILE PHONE (5) ).

| | **SHA256** | **HMAC SHA256** | **1024 Bit Modular Multiplication** | **1024 Bit Modular Exponentiation** |
|---|---|---|---|---|
| (1) | $2.8\times10^{-6}\,s$ | $5.8\times10^{-6}\,s$ | $32.8\times10^{-6}\,s$ | $19.5\times10^{-3}\,s$ |
| (2) | $5.4\times10^{-6}\,s$ | $11.1\times10^{-6}\,s$ | $96.9\times10^{-6}\,s$ | $41.5\times10^{-3}\,s$ |
| (5) | $6.40\times10^{-3}\,s$ | $33.50\times10^{-3}\,s$ | $77.70\times10^{-3}\,s$ | $5.6\,s$ |

### B. Availability

For a surveillance system, availability is twofold as there are two distinct causes of unavailability. First it is equipment reliability: it may be expected that equipments fail, thus making the service unavailable. However, note that equipment availability is a necessary condition but not a sufficient one as equipments may be up and working while the communication may be halted due to some computational or communication overload. Therefore we distinguish between equipment availability and protocol availability.

Regarding equipment availability, which is closely related to equipment reliability, there were attempts to discus it in the context of video surveillance systems [9]. However, there are no particular details or experimental data in [9] that are specific for a surveillance system and only the general availability theory is stated.

Here we will briefly state in a few lines what we consider that is relevant to be taken into account in the context of a surveillance system. In general, the availability of a system can be characterized as the ratio between the up-time and the sum of the up-time and down-time, i.e. $A = \frac{T_{up}}{T_{up} + T_{down}}$. Further, unavailability is $U = 1 - A$. Obviously, if one wants to increase availability it can simply use parallel redundancy, i.e. more devices performing the same function. For parallel systems, unavailability is the product of the components unavailability, i.e. $U = \prod_{i=1}^{n} U_i$, and since $U_i < 1, \forall i = \overline{1,n}$, the unavailability of the parallel system drastically decreases.

However, what needs to be added is that in any system there may still be single elements that can cause the failure of the entire system. For example consider a power-down case, in this case, if there is no redundant source of power all cameras will fail. Such a failure, caused by a common element on which all other elements of the system depend, is called common mode failure. Further, this kind of failure can have a variety of more subtle causes from incorrect design of products by the same manufacturer to incorrect installation by the same individual.

Thus, for a given system one should compute the availability for the entire system, using well known rules and some experimental data, and add more redundancy if the availability level is not sufficient, any basic book on the subject can be consulted [1]. Four nines availability means that the system is available for 99.99% percents of the time, this still gives 52.6 minutes of unavailability per year. On a first view such an amount of time can be enough to exploit a system. Therefore 5 nines, which means 99.999% percents up-time or higher availability levels should be achieved. However this intuition is not necessarily correct, as availability is not a measure for the duration of a particular system failure.

We used as example, data records from May 2008 on the power failures at the *aut.upt.ro* server at our university. Due to several work-outs on the energy supply infrastructure, 19 power failures were recorded. However these power failures were due to energy fluctuations and they last in the order of seconds. Although their sum could give less than 5 nines availability, still they are not harmful as they are in separate time intervals of several seconds each which have almost no meaning for the video surveillance system.

Now we get to the second kind of cause for system unavailability which comes from communications failures. Here, our only concern is that the communication protocol itself can be subject to DoS attacks. Of course if the network resources are exhausted by legitimate users, then the network must be properly re-designed. In the context of protocol failures, DoS protection techniques, possibly the use of cryptographic puzzles, should be considered in the design of the protocol. In brief, precautions should be taken both on the sender and on the receiver side. On the sender side the biggest concern is computational power, therefore simple cryptographic primitives such as hash functions and MAC codes should be used, we will follow this in our application. On the receiver side, the concern is both on the computational power and communication bandwidth as it is expected to have receivers with different communication abilities. For this purpose it is desirable for the protocol to allow different frame rates for different receivers, our application will respond to this requirement as well.

As a partial conclusion, we consider that availability can be easily reached by following good engineering practices. Equipment reliability should not represent a problem as it is not difficult to achieve, but at most expensive, with parallel redundancy. Still, to the best of our efforts we were unable to find failure data for surveillance cameras, although such data can be easily found for many other components [14].

### C. Authenticity

In order to assure authenticity for the information, the use of cryptographic techniques is mandatory and is the only way to assure that images indeed originate from the video surveillance system and are not injected by some adversary. This is also acknowledged by research papers and general books on the subject [4].

For this purpose, the first cryptographic solution is the use of digital signatures; however these cryptographic primitives are expensive as shown in table 1. Also the use of secure point-to-point connections such as the SSL is not a good alternative as the same images needs to be sent to many users which will require different encryption keys and will require for the sender to encrypt the same message under each distinct key thus causing overhead on the sender's side.

Still, authenticity can be assured at a cheaper price by the use of MAC codes. However, MAC codes require a secret shared key which is not practical in a setting with many receivers. Fortunately there is a good solution to overcome this: by the use of time synchronization a single MAC key can be used for all users. The first proposal for this was the TESLA protocol which can be found in several variants [6], [7], [8]. In the context of the AAA framework we believe that TESLA like protocols are the best alternative.

## III. BRIEF ANALYSIS OF THE DCS-900 IP CAMERA

A brief analysis of the DCS-900 [11] security camera, a popular camera from D-Link, may be useful. DCS-900 has nice features; it is easily configurable via a web-interface and comes at a reasonable price. On-line captured images may be viewed from a browser that supports Java or ActiveX. However, with respect to some serious security levels it does not provide good enough characteristics for a surveillance system and is nothing more than a toy that will be easily breakable by an adversary. We enumerate some of its weaknesses:

- *Password length vulnerabilities*. For the Revision A variant, the camera manual specifies that user authentication to the camera is done by an 8 character password from the printable ASCII set, i.e. 94 characters. This is roughly the equivalent of a 52 bit key which is not a sufficient security level for today. More, as these passwords are chosen by users they can be subject to dictionary guessing attacks as well. This restriction is not stated in the Revision B manual, but the web interface for the Revision B camera that we have does not allow us to introduce passwords larger than 8 characters, so the restriction is the same. Also, if from the setup wizard for Revision B a passwords larger then 8 characters is set then the password is not recognized at the login via the web interface. More, if you enter just the first 8 characters of the password used in the setup wizard then it will work, which proves that the password is actually truncated by the setup wizard to the first 8 characters.
- *Password disclosure vulnerabilities*. In the default settings, with the user access control option disabled, we captured packets in which the admin id and password are sent in clear text. Although it is highly recommended in the user manual to change admin password to a stronger password (by default it is set to blank), by analyzing network traffic with Wireshark we seen that the password itself is sent in clear-text on the communication line and therefore can be extracted by any adversary that can analyze network traffic. It is not clear for us why this happens only when the access control option is disabled as the camera software is not open source; but an adversary that eavesdrops on the communication line in the initial setup stage when the user access control option is disabled can certainly capture the password. One such adversary may be an ISP provider, or any administrator from the network nodes over which your network traffic is routed. Capturing the admin password is a serious threat as one can subsequently do anything from making the camera inaccessible to increasing the delay at which the pictures are taken, even changing the IP of the camera or installing a dummy camera on the same IP. The scenario in which we tested is depicted in figure 1, Wireshark was running on the un-trusted server while the user was trying to access the camera from a different network. Also, when users are added through the web interface from a remote computer, new user names and passwords are sent in clear-text and can be easily captured. Therefore adding new users is insecure.
- *Confidentiality vulnerabilities*. In the same setting we were able to capture packets sent from DCS-900 with Wireshark and to reconstruct the sent images from these packets. It easy to spot the jpg files sent from the camera as jpg files start with 0xFFD8 and end with 0xFFD9. Therefore DCS-900 does not assure any confidentiality for the communication.
- *Authentication vulnerabilities*. As stated previously, user passwords can be extracted in certain situations, while password based authentication itself does not offer a high level of security. Also, no authenticity information for the transmitted pictures seems to be embedded; it is almost clear to us that packets can be replayed or replaced by an adversary and it remains as potential future work to try this.

As a partial conclusion on the DCS-900 we can state that it behaves well with respect to accessibility as it allows remote access via a web-interface by any remote device that supports Java or ActiveX. However the system is vulnerable with respect to the objective of authenticity as there is no authenticity on the broadcasted images. Even with respect to availability the system may be vulnerable as an adversary may be able to eavesdrop and capture the admin password, if the camera is not properly configured as stated previously, and simply stop the camera from broadcasting, thus causing a DoS.

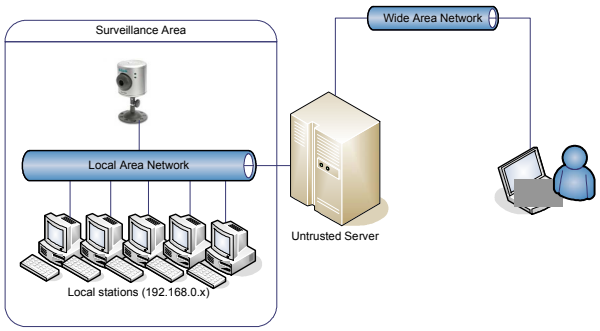


Figure 1. Application setting for the DCS-900 camera

## IV. THE PROPOSED SYSTEM

### A. Outline of the system

The scenario that we address is depicted in figure 2. It consists in a camera that broadcasts images to a number of receivers. A video cryptographic module (VCM) is used to communicate with the camera. Only the VCM is forwarded over the WAN. For the moment the VCM was implemented as a server application in Java and runs on a standard computer. However, it remains for us as future work to implement this on a single board computer (SBC) or on a FPGA board that can be rack-mountable in order to obtain a

handier device. The VCM is mainly responsible for assuring authenticity. The VCM is also flexible and allows connection of different kinds of cameras that are connected over RJ45 or USB connectors as well.

The cryptographic protocol for assuring authenticity is based on MAC codes and time synchronization similar with the TESLA protocol [7]. The application from the VCM provides time-synchronization as well, however it is better if time-synchronization is not provided by the VCM as synchronization requires one digital-signature which is more computational intensive and can be exhausted to cause DoS. For this purpose, external time-servers may be used as well.

Now we outline the characteristics of the proposed system in the introduced AAA framework:

- *Accessibility*. Both the client and server parts are implemented in Java which is architecture independent. Because of this we were able to run the client parts of the application on standard computers, PDAs and mobile phones. The client is a stand-alone application; however it is also possible to develop an applet that runs in a browser (as the DCS-900 camera does).
- *Availability*. Threats against protocol availability are partially removed by the fact that it is a broadcast protocol and the sender does not get additional requests from the receivers that consume resources, except from sending the images which are embedded in a packet that is the same for all receivers. This also removes the drawback that each new receiver consumes new sender's resources as would happen if the images are encrypted (or authenticated) with a different key for each receiver (which will linearly increase the computational overhead). However, more serious DoS attacks can take place on the sender in the synchronization stage, as the sender is required to compute a digital signature that is more computationally intensive. To see how much computational time it will take to make time synchronization, a comparison between the computational time for hash functions and for digital signature can be done. In brief, synchronization requires the same amount of time as authenticating hundreds of messages. The cleanest solution that can be used is to use a distinct time server than the VCM as stated previously. Another solution will be to provide time-synchronization only to legitimate users that authenticate to the VCM and only after certain periods of time as it is unlikely for users to lose synchronization too quickly. A detailed analysis for DoS attacks on the TESLA protocol is available in [7], we consider as potential future work to use several suggested improvements.
- *Authenticity*. The protocol guarantees the authenticity of the information by the use of MAC codes and time synchronization. Two constructive techniques can be employed for the key chain: the use of hash functions and the use of modular multiplication whose computational time is close to the range of a hash functions. In the later case the security of the protocol is provable to be equivalent to the integer factorization problem which is infeasible to solve.

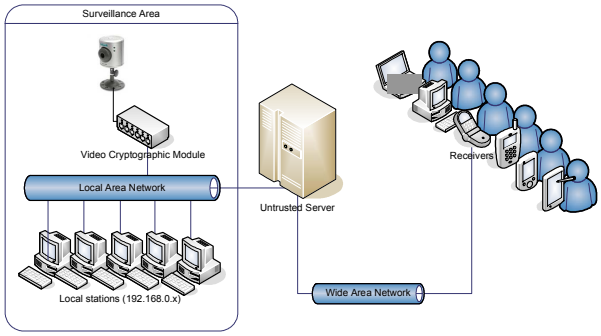


Figure 2. Application setting for the proposed system

## B. *Protocol description*

In our application we have also used the function $f(x) = x^2 \bmod n$ instead of a hash function for the construction of the chain, this is a significant difference from the TESLA protocol. We previously used a related solution on mobile phones [3] which proved to be efficient. This makes the i-th key from the key chain equal to $k_i = x^{2^i \bmod \phi(n)} \bmod n$ and by using a time-memory trade-off each key can be computed at the reduced cost of one modular multiplication. The advantage in this construction of the key chain is that it allows us to broadcast for an unbounded period of time, in contrast to hash chains which exhaust and must be re-initialized.

The protocol has the following send stage used for all receivers:

**Send Stage:**

$VCM \rightarrow Receivers$ :

$$P_i = \left\{ image_i, MAC_{KD(k_{i+1})}\left(image_i\right), k_i \right\}, i = \overline{1,\eta}$$

In order to allow receivers to receive images at a desired frame rate, the protocol is modified in the following way: in the synchronization stage, each receiver will also send a constant which establishes the frame-rate as a multiple of the broadcast rate of the server. Thus, each receiver can choose its own frame-rate and this choice depends mostly on its computational power and bandwidth. This is resumed in the following:

**Synchronization Stage:**

$Receiver \rightarrow VCM$ : $Nonce_R, r$

$$VCM \rightarrow Receiver : \left( Nonce_R, r, t^{VCM}, t^{VCM}_{broadcast}, k_{current}, \delta \right)_{SigVCM}$$

The actual frame-rate for the receiver will be $frame\_rate = \frac{1}{r\delta}$. Here $t^{VCM}$ denotes the current time on the VCM and $t^{VCM}_{broadcast}$ denotes the time at which broadcast was started, $k_{current}$ is the key disclosed in the current time interval. All these values are digitally signed by the VCM.

### C. Experimental results

In tables 3 and 4 some communication statistics in the context of a mobile phone used as a receiver are presented, the squaring function was used for the construction of the chain. The mobile phone is the device with the lowest computational power that we have and most of the limitations on the number of packets that can be sent per second come from the speed of the Internet connection offered on mobile phones by Orange Mobile Operator [15]. Experimental results show that 2 to 4 frames per seconds are feasible to receive on the Nokia 6288 mobile phone; implementation was done on the Nokia Series 40 SDK [12]. Even if this is not a real-time transmission, which requires in the order of tens of frames per second, we believe that this is a sufficient frame-rate for a home video-surveillance task carried on a mobile-phone. More, this frame-rate is available on a mobile phone that has low computational power, and the main limitation is from the network speed. Of course, the frame rate will get to real-time if a standard computer is used as a receiver.

We also note that in the applications it is better for the keys and MAC to be received on different threads than the images since the images may be checked for authenticity even if they arrive lately as long as the key and MAC is in the correct time interval. Also, to save up computational time on the remote device in the registration stage the last disclosed key signed by the sender is given. Otherwise, if a receiver has a key which was disclosed he will need to compute $\Delta = t_{key} \cdot t^{R}_{start} \big/ \left(\delta - t_{key}\right)$ until he will synchronize with the current key of the sender.

TABLE III. COMMUNICATION STATISTICS AT 320X240 RESOLUTION.

| Key disclosure period (seconds) | Packets sent | Authentic packets received | Loss rate (%) |
|---|---|---|---|
| 1 | 169 | 169 | 0 |
| 0.5 | 265 | 265 | 0 |
| 0.25 | 77 | 28 | 63.63 |
| 0.1 | 30 | 4 | 86.66 |

TABLE IV. COMMUNICATION STATISTICS AT 160X120 RESOLUTION.

| Key disclosure period (seconds) | Packets sent | Authentic packets received | Loss rate (%) |
|---|---|---|---|
| 1 | 175 | 175 | 0 |
| 0.5 | 161 | 161 | 0 |
| 0.25 | 179 | 108 | 39.66 |
| 0.1 | 55 | 12 | 78.18 |

### D. Performance analysis

An analysis of the performance of the protocol with respect to the frame-rate is useful. In order to correctly display a frame, one will need to receive a packet and verify the key and the MAC code, this gives:

$$t_{frame} = t_{receive} + t_{verKey} + t_{verMAC} \quad (1)$$

Now, because of the time synchronization issues, in the case when each frame is sent in a distinct packet we must have:

$$\delta \gg t_{frame} \quad (2)$$

If (2) does not hold, then the received frame will fail the authenticity test as it will be received to late, outside the correct time interval for the corresponding key. This relation actually gives the limitation on the frame-rate for the case when each frame is authenticated with a distinct key. Therefore, the maximum achievable frame rate in this case will be:

$$\max_{frame-rate} = \frac{1}{t_{frame}} \quad (3)$$

However this will impose the use of a transfer rate that is established by the receiver with the worst bandwidth and this may not be fair with respect to other receivers. For this purpose we allow each receiver to choose its own frame-rate.

We underline that this can be further improved if more than one frame are authenticated with the same key, thus reducing the time for the verification of a key and MAC in relation (1). This will lead to the following:

$$t_{frame} = t_{receive} + \frac{t_{verKey}}{n} + \frac{t_{verMAC}}{n} \quad (4)$$

However, in this situation each receiver will need a buffer to store $n$ frames until the authentication key will arrive. Also the delay until the received frame can be actually proved to be authentic is larger.

Thus, now we can establish the frame rate based on the network speed and the time to compute the cryptographic primitives. In the plot from figure 3 we depict the theoretical frame rate for both hash chains and quadratic residues chains. As can be easily seen in figure 3 the frame rate cannot get to higher than 10 fps mainly because of communication bandwidth, in this plot we have assumed 128 kbps which is a suggested value for 3G networks in moving vehicles, however on the Internet connection used for the results in tables 2 and 3 the bandwidth was even worst than this. In figure 3 we also indicate the frame rate for the case when more then 1 frame is inside a packet, 5 fpk denotes that there are 5 frames in a packet, i.e. 5 frames per key.

We note that on standard computers as modular multiplications and hash functions can be computed in the

order of micro-seconds the computational time for the cryptographic primitives will not affect the frame-rate. Also the size of the key is not significant compared to the size of the packet which is determined mostly by the size of the image. Usually the image has several kilo-bytes while the size of the key will be at most several thousand bits if the squaring function is used.

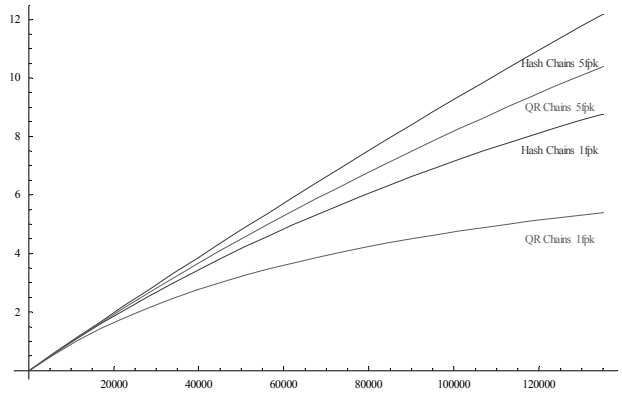


Figure 3. Frame rate variations with network-speed of 128kbps for an average 10kb frame

## V. CONCLUSIONS

A framework of three properties of a surveillance system was depicted: availability, accessibility, authenticity (AAA). The DCS—900 camera, an off-the-shelf device was analyzed and its weaknesses shown. Also an improved solution was proposed which responds to the necessities of the AAA framework. The experimental results show that frames can be received even by a mobile phone with low computational power. The proposed solution is still under development, and its complete specifications remain as future work for us. We used the DCS-900 camera just as an intuitive example; however, a comparison between several surveillance cameras in the context of the introduced framework is a potential interesting future work.

**Acknowledgements:** This work was partially supported by national research grant PNCDI PN II 940/2009.

## REFERENCES

[1] J. P. Bentley, “An Introduction to Reliability and Quality Engineering”, Addison Wesley, ISBN 0201331322, 216 pages, 1998.

[2] R. Collins, A. Lipton, T. Kanade, H. Fujiyoshi, D. Duggins, Y. Tsin, D. Tolliver, N. Enomoto, O. Hasegawa, “A System for Video Surveillance and Monitoring”, tech. report CMU-RI-TR-00-12, Carnegie Mellon University, May, 2000

[3] B. Groza, D. Pop, I. Silea, “Java Implementation of an Authentication Protocol with Application on Mobile Phones”, IEEE-TTTC International Conference on Automation, Quality & Testing, Robotics, AQTR 2008 (THETA 16).

[4] H. Kruegle, CCTV Surveillance: Analog and Digital Video Practices And Technology, Elsevier, ISBN-13: 978-0-7506-7768-4, ISBN-10: 0-7506-7768-6, 656 pages, 2006.

[5] X. Liu, A. Eskicioglu, “Selective Encryption of Multimedia Content in Distributed Networks: Challenges and New Directions,” IASTED Communications, Internet & Information Technology (CIIT), November 2003.

[6] A. Perrig, R. Canetti, J. D. Tygar, D. Song, “The TESLA Broadcast Au-thentication Protocol”, In CryptoBytes, 5:2, Summer/Fall, pp. 2-13, 2002.

[7] A. Perrig, R. Canetti, D. Song, D. Tygar, “Efficient and Secure Source Authentication for Multicast”, Proceedings of Network and Distributed System Security Symposium, 2001.

[8] A. Perrig, R. Canetti, J. D. Tygar, D. Song, “Efficient Authentication and Signing of Multicast Streams Over Lossy Channels”, IEEE Symposium on Security and Privacy, 2000.

[9] S. Sutor, F. Matusek, F. Kruse, K. Kraus, R. Reda, “Large-Scale Video Surveillance Systems: New Performance Parameters and Metrics”, The Third International Conference on Internet Monitoring and Protection, 2008.

[10] Bouncycastle.org, Bouncy Castle Crypto API, http://www.bouncycastle.org/latest_releases.html.

[11] D-Link, Building Networks for People, http://www.dlink.com/

[12] Forum Nokia, Series 40 Platform 3rd Edition SDK, http://www.forum.nokia.com/info/sw.nokia.com/id/cc48f9a1-f5cf-447b-bdba-c4d41b3d05ce/Series_40_Platform_SDKs .html.

[13] Java Sun, Java 2 Micro Edition, http://java.sun.com/javame/index.jsp.

[14] MIL-HDBK-217F, Reliability Prediction of Electronic Equipment, 1995, availlable at http://assist.daps.dla.mil/quicksearch/basic_profile.cfm?ident_number=53939.

[15] Orange Mobile Operator, Mobile Internet Access http://www.orange.ro/abonamente-date/mobile-internet.html.

[16] Video surveillance security for your World, http://www.videosurveillance.com/.