Real-time encryption and authentication of medical video streams on FPGA

Jo Vliegen*†, Bob Koninckx‡, Dave Singelée*, Nele Mentens*†
*KU Leuven, iMinds, COSIC, Leuven, Belgium
†KU Leuven, ES&S, Technology Campus Diepenbeek, Belgium
‡eSATURNUS, Leuven, Belgium

***Abstract*—This work presents an FPGA-based solution for the encryption and authentication of video streams of surgeries. The most important is minimal latency. To achieve this, a block cipher with an authenticated mode of operation is used. We choose to use AES128 with Galois/Counter Mode (GCM), because the this mode of operation is patent-free and it allows for random read access. This solution minimizes the overhead on the existing critical path to a single XOR operation.**

Our solution supports the broadcasting of the video stream. When a new receiver announces itself, it should receive the active keys of the sender. Therefore, a key transport protocol is used to establish a key between the sender and the announcing receiver.

A proof-of-concept implementation of the proposed solution has been implemented and tested. While the complete video stream is encrypted and authenticated, the demonstrator confirms that the added latency, which is around 23 μs, could not be noticed by the human eye. Random read access and the key establishment protocol provide a flexible solution.

I. Introduction

The Internet is evermore present in our daily lives. This work contributes in the efforts of using the Internet for broadcasting surgeries. In an operating room more and more flat screens are used to provide the surgeon with vital information coming from an endoscope, a heart monitor, blood pressure levels and many more sources. We consider a setup that gathers all video input streams on the sender device, sends the data over Gigabit Ethernet to the receiver device, and makes this video stream available for the most frequently used video interfaces. The latency of the video stream in this setting is of the utmost importance because this video stream could be the only feedback an operating surgeon gets.

Given the network link between both devices, these devices can be placed anywhere in the world as long as both can be provided with a decent Internet connection. The proof-of-concept solution presented in this work handles the necessary encryption and authentication challenges. Moreover, in Europe, article 17 in directive 95/46/EU [1] states that adding encryption and authentication is legally required when data is transmitted over a network, .

By physically moving the receiving device to another location on earth additional features can be achieved. Firstly, it would allow other surgeons to virtually attend the surgery. This could be helpful when dealing with difficult operations. Next to other surgeons, also medical students could follow an operation for educational purposes. A second additional feature is to have a trusted party record the surgery. A verifiable authenticated video stream could also be used for legal purposes.

in

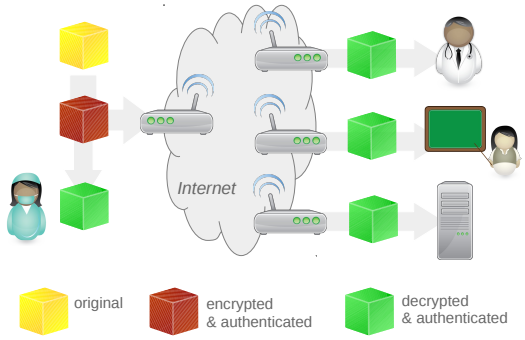

Fig. 1. The extended setup with physically separated sender and receiver devices

On the left hand side of Fig. 1 the operating surgeon is shown. The yellow box represents the video stream that is captured. Subsequently, the stream gets encrypted and authenticated which is represented by the red box. These data are then broadcast and fed back locally after which, on either side, the stream is decrypted and authenticated.

The remainder of this paper first describes the chosen protocol and the algorithms in Sect. II. Subsequently, the architecture and the network protocol are discussed in Sect. III, after which the implementation and the results of the proof-of-concept implementation are explained in detail in Sect. IV. Finally, conclusion are drawn in Sect. V.

II. Cryptographic protocol and Algorithms

A. Protocol

With two devices communicating with each other, a protocol has be installed. This protocol has to satisfy two requirements:

- a viewer of the video stream can start viewing at any given point in time (**REQ.1**);
- the maximum duration of a stream is set to 10h (**REQ.2**);
- the video stream needs to be confidential (**REQ.3**);
- the video stream needs to be authenticated (**REQ.4**).

Because the sending device is highly likely to broadcast to multiple receiving devices a key transport protocol (KTP) is

chosen in favour of a key establishment protocol. In the former a single key is determined by the sending device, whereas with the latter a different key with every receiving device needs to be established. A KTP is chosen to prevent the sender from encrypting and authenticating the same video stream multiple times.

The session keys which are used for a single surgery are generated by the sending device. When a new receiver device presents itself, the active keys need to be transported from the sending to the receiving device. The used protocol is the "Encrypting signed keys" protocol [2] and is shown in Table I.

TABLE I
THE "ENCRYPTING SIGNED KEYS" PROTOCOL AS DEFINED IN [2]

Sending device		**Receiving device**
key pair (a, A)		key pair (b, B)
		A
		m_1 =viewing request $\|\|$ B
	m_1	
	$\leftarrow$	
choose k		
$m_2 = k \,\|\| \, t^*$		
$m_3 = S_a(B \,\|\| \, m_2)$		
	$m_2 \,\|\| \, m_3$	
	$\rightarrow$	
		verify m_3 with A
$K_{session} = k = k_1 \,\|\| \, k_2$		$K_{session} = k = k_1 \,\|\| \, k_2$
$m_x = MAC_{k_2}(data)$		
$m_y = E_{k_1}(data \,\|\| \, m_x)$		
	m_y	
	$\rightarrow$	
		$D_{k_1}(m_y) = data' \,\|\| \, m'_x$
		verify m'_x

key pair (x, X) : x and X are the private and public key, respectively
$\|\|$: concatenation of messages
t: timestamp
*: indicates this is optional
$S_k(m)$: digital signature on m with key k
$MAC_k(m)$: MAC on m with key k
$E_k(m)$: encryption of m with key k
$D_k(m)$: decryption of m with key k

With this protocol in place **REQ.1** is partially met. Given that the network runs at Gigabit speed, broadcasting for 10h would end up with a total amount of data of $36 * 10^{12}$ bits. When a 128-bit block cipher is chosen, this total number of bits is smaller than $2^{128/2}$. Therefor no re-keying is necessary during a single broadcast session, meeting **REQ.2**,

Using the "Encrypting signed keys" protocol, both types of audiences (being colleague surgeons and class rooms) can attend a surgery-broadcast. For the recorder type of audience, the first phase would be identical. When a broadcast is finished, a recorder can use its own private key (from a different public key pair) to sign the complete received video stream. This signature cannot be generated by the sender because there is always the possibility that certain frames do not make it across the Internet, or that they are corrupted on the way. The signature on the video stream by a trusted recorder could be used in a legal setting to prove the authenticity of a recording. Through a public-key infrastructure the complexity is flexible with respect to size.

With the session key safely established the communication can occur encrypted (**REQ.3**) and authenticated (**REQ.4**).

B. Algorithms

With authenticated encryption being around for more than a decade, it needs no further argumentation that choosing such an algorithm fits the obtained goal. Bellare and Namprempre [3] have shown that Encrypt-then-MAC (EtM) provides the best security, assuming that the MAC is strongly unforgeable. The authenticated encryption algorithm used in [3] is GCM [4]. The underlying block cipher is AES [5] with a 128-bit key, and the authentication algorithm is GMAC [4]. GCM allows for random read access so a receiver can join the stream at any point in time. Moreover, when network frames get lost this will not break the decryption and authentication of the following network frames. This, in combination with the choice of the protocol in Sec. II-A, assures that **REQ.1** is completely met. An additional argument for choosing GCM is the fact that GCM is unencumbered by patents.

III. ARCHITECTURE AND NETWORK PROTOCOL

A. Architecture

As stated above, the existing product consists out of a sending and receiving device which communicate over a Gigabit network. During initialisation this connection is used by a microprocessor for the key establishment protocol and (for the receiver) to obtain the session key which is partially used for encryption and partially for authentication. These keys are then handed over to the hardware for operational use.

Fig. 2 shows a block diagram of the architecture of the GCM core. The actual implementations for sender and receiver slightly differ. The main difference is that the receiver has a comparator which verifies that the generated MAC matches the received MAC; and the receiver has additional glue logic to get the counter value, present as authenticated plain text, out of the network header frame.

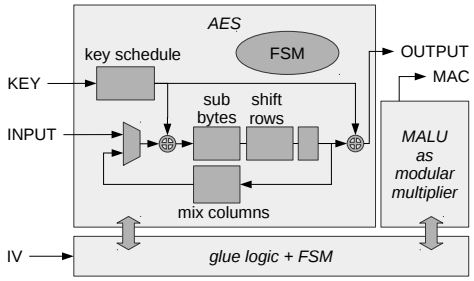

Fig. 2. Block diagram of the architecture of the GCM core

The block diagram of Fig. 2 is implemented for both the sender and the receiver. Due to reasons explained above, the size of the receiving device is larger than the size of the sending device.

B. Network protocol

The existing product uses the standardised TCP/IPv4 protocols. With the addition of encryption and authentication, every

network frame that is broadcast must include a plain text field containing the counter value of the GCM mode. This field needs to be in plain text, but it should also be authenticated so modifications to this counter value, both accidental and intentional, will be noticed.

With every new network frame, this counter field is processed as authenticated data. Hereafter the remainder of the frame is filled with raw video data which is encrypted and authenticated. In a real setting, encoding techniques need to be applied to increase efficiency, but this is out of scope for this proof-of-concept implementation.

When considering a default network setting, the maximum packet size is 1'518 bytes of which 58 bytes are typically used for multiple protocol headers (14 bytes Ethernet header, 20 bytes IPv4 header, 20 bytes TCP header, and 4 bytes of CRC). This leaves 1'460 bytes for video data of which 16 bytes are assumed to be authenticated data (the counter value

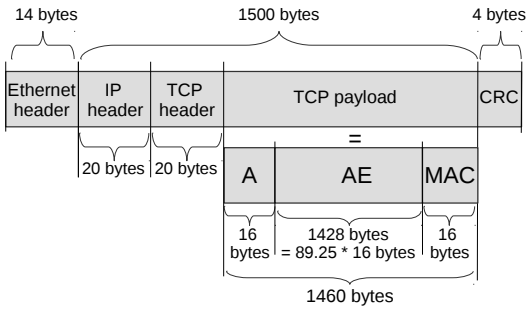

Fig. 3. Network encapsulation of the video stream

IV. Implementation and Results

To control the GCM core shown in Fig. 2, four commands are implemented. The first command ('init') is used to set the Internal Value and to initialise the control path. Subsequently, the GCM core can process one or more blocks of authenticated data, instructed with the command 'a'. As explained in Sec. III-B, 16 bytes are authenticated. The GCM core can then process one or more 16-byte blocks of data that needs to be authenticated and encrypted ('ae'). Finally, the GCM core processes the lengths fields (which indicate the length of the authenticated data and the encrypted-and-authenticated data, respectively) in order to calculate the MAC ('f').

The number of occupied slices for the sender is 1'442 and that of the receiver is 1'918 slices. Note that these numbers cannot be compared directly. The sender is implemented on a Xilinx ML507 development board which contains a Virtex 5 FPGA, while the receiver is implemented on a Xilinx ML605 development board which contains a Virtex 6 FPGA.

Table II summarises the results with respect to latency assuming the component is running with a 125 MHz clock which is used by Gigabit network devices.

command	init	a	ae	f	Σ
# clock cycles	13	5	16	14	
latency_unit [ns]	104	40	128	112	
latency_fnf [ns]	104	80	11'264	112	11'560

TABLE II
THE NETWORK LATENCY ON A DEFAULT NETWORK, WHERE FNF STANDS FOR FULL NETWORK FRAME

According to Miller [6] a latency of 100 ms (or less) is perceived as instantaneous. The solution provided in this paper introduces an overhead of 23.2 μs, which, which is negligible with respect to the 100 ms. Therefor the operating surgeon will not notice the additional overhead.

V. Conclusions

This work presents a solution for FPGA-based encryption and authentication for video streaming of surgeries. The choice of the cryptographic protocol ("Encrypting signed keys") together with the used algorithms (GCM with AES128) are explained and detailed discussions on the implementation and the network protocol are provided.

The additional latency due to encryption and authentication is kept to a minimum and is visually not detectable with an increase in latency of 23 μs.

VI. Acknowledgements

This work was supported in part by the Research Council KU Leuven: C16/15/058. In addition, this work is supported in part by the Flemish Government, FWO G.0550.12N, G.00130.13N and FWO G.0876.14N, by the Hercules Foundation AKUL/11/19, and by the European Commission through the Horizon 2020 research and innovation programme under contract No H2020-ICT-2014-644371 WITDOM and H2020-ICT-2014-644209 HEAT. Finally, this work was supported by IWT-O&O and the CORNET project "DynamIA: Dynamic Hardware Reconfiguration in Industrial Applications"; it is funded by IWT Flanders with reference number 140389.

References

[1] European Parliament and Council of the European Union, "Directive 95/46/EU," Brussels, pp. 31–50, November 1995.

[2] A. J. Menezes, S. A. Vanstone, and P. C. Van Oorschot, *Handbook of Applied Cryptography*, 1st ed. CRC Press, Inc., 1996.

[3] M. Bellare and C. Namprempre, "Authenticated encryption: Relations among notions and analysis of the generic composition paradigm," Cryptology ePrint Archive, Report 2000/025, 2000, http://eprint.iacr.org/.

[4] M. J. Dworkin, "Sp 800-38d. recommendation for block cipher modes of operation: Galois/counter mode (gcm) and gmac," Gaithersburg, MD, United States, Tech. Rep., 2007.

[5] J. Daemen and V. Rijmen, *The Design of Rijndael: AES - The Advanced Encryption Standard*. Springer-Verlag New York, Inc., 2002.

[6] R. B. Miller, "Response time in man-computer conversational transactions," in *Proceedings of the December 9-11, 1968, Fall Joint Computer Conference, Part I*, ser. AFIPS '68 (Fall, part I). New York, NY, USA: ACM, 1968, pp. 267–277. [Online]. Available: http://doi.acm.org/10.1145/1476589.1476628