On the Power Consumption of Cryptographic Processors in Civil Microdrones

Abdulhadi Shoufan[1], Hassan Alnoon[2] and Joonsang Baek[1]
[1]*Electrical and Computer Engineering Department, Khalifa University, Abu Dhabi, U.A.E.*
[2]*Advanced Technology Consultancy L.L.C, Emirates Advanced Investment Group L.L.C, Abu Dhabi, U.A.E.*

Keywords: Civil Drone Security.

Abstract: In this paper we analyze the security requirements of civil microdrones and propose a hardware architecture to meet these requirements. While hardware solutions are usually used to accelerate cryptographic operations and reduce their power consumption, we show that the latter aspect needs to be reviewed in the context of civil drones. Specifcally, adding cryptgraphic hardware to a flying device increases its weight and, thus, the power needed to fly this device. Depending on the relative weight of the added cryptographic processor, the computational power advantage of the hardware solution may be undone by the additional hadrware weight. This aspect is analyzed for the proposed hardware solution.

1 INTRODUCTION

Unmanned aerial vehicles have long been used for surveillance in homeland security. Due to their high costs, however, the effectiveness of UAVs in civil applications has always been under discussion. This circumstance contributed to the development of lower-cost platforms in form of quadrotors or multirotors, which we refer to as civil microdrones (CMD). The absence of mechanical linkages to vary the rotor blade pitch angle, the usage of multiple rotors and the improvements in the algorithms that controls its advanced electronic systems and sensors has allowed CMDs to be of small-size and easy to fly, control, and maintain. Due to their ability to interact with the environment in close proximity safely, CMDs are gaining an increasing attention in different civil application areas (Quaritsch et al., 2008). In environment monitoring CMDs use sensors and cameras to collect information that is difficult to be obtained by helicopters because of space or cost reasons. An example for this application area is monitoring the woods, desert, animals, and agriculture. Disaster management is another application area where CMDs can give a quick overview of the size and the grade of a disaster caused by flooding, earthquake, or fire. Based on this information an efficient plan for rescue activities can be defined and implemented. Especially in the surveillance and law enforcement areas CMDs are receiving an increasing acceptance to observe restricted areas, state borderline, pipelines, private properties, or public events for detecting suspicious actions and preventing terrorism.

Commercial CMDs are penetrating and the importance of these devices was recognized by the governments of many countries. Several research projects were supported by governmental and non-governmental funding sources: USICO (Unmanned Aerial Vehicle Safety Issues For Civil Operations) is a European project that dealt with the operation and the safety of UAVs, e.g., for collision avoidance. CAPECON (Civil UAV Application & Economic Effectivity of Potential Configuration Solution) is another European project, which aimed at the specification of civilian applications for the UAVs. A major aspect of this project is the integration of large-scale UAVs into the air traffic control system.The design of a distributed control system for cooperative detection and monitoring using heterogeneous UAVs (Gancet et al., 2005) was the subject of the project COMETS funded by the European Union, too. μDrones is another European project which focuses on microdrones for autonomous navigation for environment sensing. The purpose is the development of hardware and software modules to ensure flight stability, navigation, localization, and robustness to unexpected events. Several topics related to networked UAVs were investigated at the Aerospace Controls Laboratory at MIT (How et al., 2008). These include sensor network design, adaptive control, and model predictive control.

Shoufan A., Alnoon H. and Baek J..
On the Power Consumption of Cryptographic Processors in Civil Microdrones.
DOI: 10.5220/0005273402830290
In *Proceedings of the 1st International Conference on Information Systems Security and Privacy* (ICISSP-2015), pages 283-290
ISBN: 978-989-758-081-9

Starmac is a project at Stanford University (Hoffmann et al., 2004), which works on autonomous collision and obstacle avoidance and task assignment formation flight using both centralized and decentralized techniques. Additionally to these projects, an enormous amount of individual research papers can be found in the literature that address different topics such as specific control systems (Chiu and Lo, 2011; Dierks and Jagannathan, 2010; Voos, 2006) and obstacle avoidance approaches (Minguez and Montano, 2004; Yuan et al., 2009; Zsedrovits et al., 2011).

With all this attention to the functional aspects of this new technology, the consideration of security aspects is still very limited. When it comes to security, the CMD should be considered from two points of view: as a threat and as a target. CMDs are considered as security threat because they can be used for mischievous or criminal activities. With the versatile technical possibilities, the drones can be abused to spy on unaware individuals. Privacy and civil liberty advocates have raised many doubts about the legitimacy of facial recognition cameras, thermal imaging cameras, open Wi-Fi sniffers,license plate scanners and other sensors. For example, the researchers at the London-based Sensepoint security firm showed how to equip a drone with a software program called Snoopy that allows the vehicle to steal data from surrounding mobile devices searching for a Wi-Fi network. The researchers successfully demonstrated the ability of the Snoopy application to steal Amazon, PayPal, and Yahoo credentials from random citizens while the drone was flying over their heads in the streets of London. Not only individual's privacy and security seem to be threatened by the CMD technology but also commercial, industrial, and governmental sectors. Recently, a drone was used to leak behind the scenes footage of the filming of the new Star Wars movie series without the authority to be able to do anything about it. The penetration of new commercial products such as DroneShield, that help detect spying drones based on acoustic noise or radio signals is a clear evidence for the seriousness of these concerns. CMDs, however, can themselves be an attack target. The principal risks are represented by the possibility that criminals and cyber terrorists can attack unmanned aerial vehicles for several purposes such as hacking or hijacking.

The paper deals with the CMD security from this latter point of view and provides an in-depth analysis of the CMD communication security. Based on this analysis a hardware-based solution is proposed to obtain the required performance for cryptographic functions. The solution is investigated with respect to power consumption and compared with a software solution. This investigation shows that the cryptoprocessors advantage of low power consumption is suppressed in this application because of the additional power that is needed to carry and fly the additional hardware.

The paper is structured as follows. Section 2 presents the security vulnerabilities and threats of civil microdrones. Section 3 describes the proposed hardware solution and Section 4 gives some implementation details. In Section 5 we analyze the performance and the power consumption of the developed prototype. Section 6 concludes the paper.

2 SECURITY ASPECTS IN CMD COMMUNICATION

This section describes the vulnerabilities of CMDs and the attacks they are exposed to. The security requirements are then defined and supporting technical solutions are discussed.

A CMD is a remote-controlled flying platform. As a communication system the CMD is exposed to different passive and active attacks with specific impacts as summarized in the following:

1. **Control Data Manipulation:** Control data manipulation is perhaps the most critical active attack on a CMD. Sending fake control commands by an attacker may be motivated by one of the following:

   (a) **Dislocating the CMD:** When CMDs are used for surveillance and law enforcement purposes, then people under surveillance, who may be criminals, may work against that by trying to fly the CMD away from the monitored area.

   (b) **Physical Theft (Hijacking):** Some commercial microdrones are highly expensive and can fly to far distances. Having it under control, an attacker may fly a CMD to a desired place where she or he can steal it physically. The physical theft may also be motivated by the intention to tamper with the device to disclose cryptographic keys or secret data.

   (c) **Damage to Persons and Property:** Due its light weight, a CMD can fly with high speeds. A malicious attacker may use this feature to steer a CMD and cause it to collide with high momentum with an arbitrary or certain target. In addition to the risk of person's injury and damage to property, such attacks may cause major legal liability problems to the CMD owner or to his or her insurance provider.

2. **Replay Attacks on Control Data:** Adversaries may record control data and resend it later to misuse the CMD. This kind of attack works even if control data are protected against manipulation.
3. **Denial-of-Service Attacks (DoS):** CMDs are power-critical microcontroller-based systems. Any computation overhead causes additional power consumption and shortens the flying time of the CMD. An attacker may use this vulnerability to flood the CMD with faked or replayed commands and, thus, impair the availability of the CMD. Even if the CMD is able to identify these commands as fake or replay, e.g., using a verification process, this approach demands considerable computation power and time. In extreme cases, such a DoS attack may cause the CMD to be completely engaged in the authentication process until the CMD battery is fully discharged. Note that flooding the CMD with faked or replayed commands, as treated in this paper, can be regarded as a DoS attack on the application layer. DoS attacks in wireless networks on lower layers such as the physical layer (jamming) and the MAC layer are well-studied in the literature (Pelechrinis et al., 2011; Bicakci and Tavli, 2009) and out of the scope of this paper.
4. **Information Theft:** The bird's-eye view provided by the CMD is attracting professional photographers to take high-quality pictures or record videos from the sky. As a rule, these pictures or videos are sent to the ground station not only to save the memory usage on the CMD but also for the photographer to control the picture taking or the video recording. Obviously, such data may be of high value and under copy right. Sending the data in plaintext opens the way for adversaries to record and distribute the data at low cost and to cause considerable financial loss to the professional photographer. Understandably, information theft may be exercised in many other CMDs' application areas with different motives.
5. **Information Manipulation:** Information data sent by the CMD may be used for critical purposes. For instance, pictures taken in surveillance and law enforcement applications may be used as means of evidence. Also, the success of a rescue plan for disaster management may rely on the picture taken by the CMD. Obviously, any manipulation of these data including replay attacks may cause severe problems.

The severity of the described attacks depends on the application area and its data criticality. For instance, CMD's used by traffic police to inspect a car accident area blocked by a traffic jam may pose a low risk because of the randomness of both the location and the time point of such a use case: The probability of the presence of an active attacker at the location of a random accident, at which a CMD is used, is fairly low. In contrast, CMDs used against organized crime, such as drugs' smuggling, are assumed to be at high security risk. This is because organized criminals are expected to be able to perform sophisticated attacks.

Based on the later analysis of CMD vulnerabilities and potential attacks, the following security requirements can be defined:

1. Confidentiality of information and control data communicated between the CMD and the ground station
2. Integrity and authenticity of information and control data
3. Resistance to replay-attacks
4. Resistance to denial-of-service attacks
5. Tamper resistant key management on the CMD

These security requirements can be met by different technical solutions as detailed in the following.

2.1 Data Confidentiality

Depending on the application, remote control data (Upstream), information data (Downstream), or both should be encrypted. For CMD communication the AES algorithm is selected with a key length of 128 bit (Daemen and Rijmen, 2002). AES in its native operation mode (Electronic Code Book (ECB)) results in the same ciphertext block if the same plaintext block is encrypted, which presents a security risk. To avoid this, the Cipher Block Chaining mode (CBC) can be used. CBC encryption provides confidentiality against Chosen Plaintext Attack (CPA), which is a strong but essential requirement for secure symmetric-key encryption (Katz and Lindell, 2008).

2.2 Data Integrity and Authenticity

To enforce Data Integrity and Authenticity we used a message authentication code scheme (MAC) that uses a symmetric key. A MAC scheme can be either proprietary, hash-based, or block-cipher based. We chose to use CBC-MAC which is block-cipher based MAC scheme so that is can share the same AES core used for encryption due to hardware resource constraints.

The high similarity between the MAC scheme and the CBC-mode of encryption is essential for the design of a compact cryptographic engine as will be detailed in 3.1. Note that message authentication code

uses AES in encryption mode only in both the sender and receiver sides of communication. MAC also implicitly provides data integrity, as any change in the message will result in a different MAC value.

2.3 Resistance against Replay-attacks

Even encrypted and authentic messages may be captured by an attacker and replayed at a later point of time. Replayed data passes the authentication process successfully on the receiver side. Therefore, to detect replay attacks additional information is added to the message, which enables the receiver to verify the freshness of the message. The additional information can be a nonce (number used only once), time stamp, or a sequential number. In CMD communication, we propose using sequential numbers against replay attacks, as these numbers can be deployed for additional purposes, specifically to detect packet loss and to restore the packet order.

2.4 Resistance to DoS Attacks

DoS attacks are versatile and can be carried out on different network layers. Countermeasures are usually specific to one or small number of attacks and operate on specific layers embedded in firewalls, network routers, switches, etc. As noted previously, we focus on a DoS on the application layer by assuming that the attacker is able to flood the CMD with fake or replayed commands to affect the availability of the vehicle. In CMD communication, command authentication and the replay attack countermeasure play a detective and a preventive role in the defense against DoS attacks. First, the CMD keeps track of received commands and periodically determines the ratio of non-authentic commands and replay commands to the total number of received commands. If this ratio becomes higher than a specific value a DoS attack is assumed. Second, command authentication and the replay attack countermeasure perform a primary filtering of DoS commands. This means that the system components that follow the authentication process are preserved from the DoS effects. Using authentication to mitigate (not eliminate) DoS attacks is deployed in the IPSec protocol suite, for instance.

2.5 Tamper-resistant Key Management

Key management is a challenging task that addresses key generation, key exchange, key storage, key update, and key revocation, in general. The following paragraphs describe how the proposed key management scheme works for CMD communication.

2.5.1 Key Generation

The encryption and authentication keys are generated by the ground station based on the key generator specified in ANSI X9.17 which is a key management standard for financial institutions published by the American National Standard Institute (NIST, 1985). This key generator relies on the block cipher 3DES but can use any other block cipher. In our case we use AES, as it is already available for other security functions.

2.5.2 Key Exchange and Storage

Key exchange and key storage are usually treated separately. In our case they are related due to the used hardware platform. As will be seen later, we use an FPGA to run the security functions on the CMD. The used FPGA is SRAM-based, which means that the configuration data must be loaded to the FPGA at each start. The configuration data is stored in a small external configuration memory mounted on the same board. Thus, a permanent key storage on the FPGA is impossible. Furthermore, storing the keys in the configuration memory as a part of the configuration data poses two risks. First, if the attacker gets physical access to the CMD she or he can temper with the memory chip to extract the key. Furthermore, the attacker may analyze the configuration data while transferred from the memory to the FPGA to obtain the key.

To avoid these risks we propose the following simple scheme. Before each flight, keys are generated by the ground station and written to the FPGA by wire. Wireless submission is improper as radio data can be intercepted by adversaries. The keys are stored in dedicated registers inside the FPGA. These registers can only be read by the cryptographic engine and has no read interface to outside. Writing the keys to the FPGA by wire may appear to cause an operational overhead. However, flying a CMD is always preceded by a setup phase. The proposed key exchange approach can be seen as a step in this setup phase.

2.5.3 Key Update and Revocation

Updating symmetric keys is recommended to enhance security. The idea is to reduce the amount of data encrypted with the same key to limit the damage, if the key is compromised. Usually, symmetric keys are updated for each new message or for each new communication session. For the CMD we propose updating the encryption and the authentication keys for each flight. The new keys are generated according to standardized key management algorithm (ANSI X9.17) and sent to the FPGA by wire as described in the previous paragraph.

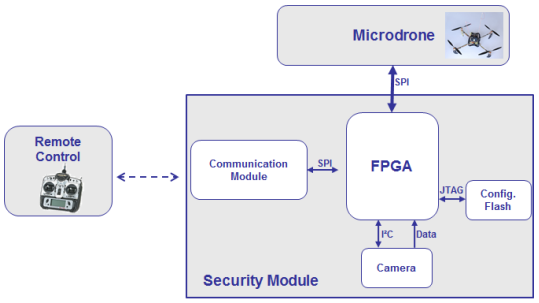

Figure 1: General Architecture of the Secure CMD System.

The registers used to store the keys are provided by a reset input that can be activated by a special command from the ground station. The CMD operator can submit this command in emergency cases when she or he loses control over the CMD for any reason.

3 SYSTEM ARCHITECTURE

The general architecture of the CMD system given in Fig. 1 consists of three main components: The remote control, the CMD, and an extension module connected to the CMD that embeds the cryptographic engine.

The hardware architecture that secures the CMD communication is depicted in Fig. 2. The architecture comprises a cryptographic engine; the on-chip JPEG encoder , which on its part embeds a camera controller; a video streamer and the control data receiver to manage the data communication; A Digital-2-PPM converter, which receives the digital control data extracted from the upstream packet and converts it back into a PPM signal (Pulse Position Modulation). In the following sections the different components of this architecture will be explained with special focus on the Cryptographic Engine as it provides the core of the security solution.

3.1 Cryptographic Engine

The Cryptographic Engine is the central part of the hardware architecture. The authentication key and the encryption key are written to the corresponding registers during the CMD setup. The Cryptographic Engine serves the video streamer and the control data receiver. It expects from these modules an intermediate MAC value, a plaintext block, and a ciphertext block. The particular module indicates its need for data processing by sending a start signal, the cryptographic module also decides which module will be served next using a ”fairness scheme” that was developed to prevent the data-intensive video streamer

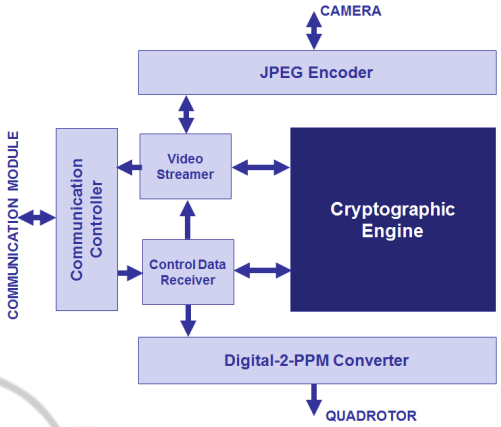

Figure 2: General Architecture of the FPGA Hardware System.

from occupying the cryptographic engine for a long time at the cost of the control data receiver. In the idle state the control data receiver are given priority over the video streamers. This is necessary because the control data include some information for video control.

3.1.1 Processing Video Streamer Data

Video data are prepared and sent by the CMD. Before sending the data, the Cryptographic Engine (CE) processes the video data in two stages for encryption and authentication. First, the Cryptographic Engine receives the video data block to be encrypted and either the ciphertext block of the previous block or the encryption initialization vector (IV). Secondly the Cryptographic Engine is fed with the encrypted video data block and either with the intermediate MAC value or with the authentication initialization vector.

3.1.2 Processing Control Data

Control data are sent to the CMD by remote control. The CMD processes the control data in two stages before sending it the Microdrone. First, in the authentication stage the Cryptographic Engine receives the data block and either the intermediate MAC value or the authentication initialization vector. Secondly, in a decryption stage the encrypted data block and either the previous encrypted control data block or the encryption initialization vector.

3.2 Remote Control Data Receiver

The Control Data Receiver receives control data from the communication controller in form of a 384-bit packet that is compound of three 128-bit blocks. The

first block represents the encryption initialization vector IV_{enc} which is not encrypted but authenticated by the ground station. The second block is the control block that includes the actual control data in encrypted form. The third block is the MAC value of the first two blocks. These data are sent to the Cryptographic Engine for authentication and to decrypt the control data block.

3.3 Video Streamers

Control data are sent from the ground station at a low rate according to the PPM scheme. In contrast, video data are sent continuously at high rate. One image occupies many packets. The size of the packet relies on the used communication module.

Each video packet includes an 128-bit encryption initialization vector IV_{enc} (or the last ciphertext block), a variable number of 128-bit video ciphertext blocks, a 128-bit encrypted status block, and the 128-bit MAC value of the entire packet.

The Video Streamer receives the byte-wise from the communication controller through a first-in first-out memory (FIFO). Using a shift register, 16 bytes of these data are concatenated to construct a 128-bit data block. In addition to the video data, the Video Streamer constructs and appends a status block to each downstream packet. The VS controller also exchanges control signals with the Control Data Receiver, the Cryptographic Engine, the communication Module, and the JPEG Encoder. Data is processed according to a developed abstract finite state machine which is responsible for creating the different block of encrypted data according to the stream.

3.4 Digital-2-PPM Converter

The used remote control generates a PPM frame with a length of 22ms. For each control channel a rectangular pulse is produced, whereas the pulse width is proportional to the control value. The minimum and the maximum width is 0.7ms or 1.5ms, respectively. The minimum distance between each two succeeding pulses is 0.4ms. The remaining time of the 22ms is used as a start signal.

The CMD receives all the channel data at once embedded in the control block inside the control data packet. The PPM signal is generated using different counters. The first counter has a fixed preset value that depends on the clock frequency to produce the start pulse. The second counter produces the control pulse. It has a variable preset value to produce time values between the minimum and maximum width of a PPM pulse. The preset value of this counter is the 8-bit digital value of the corresponding control channel. The third counter produces the pause periods which vary between 1.8 and 0.4 ms.

3.5 Other Components

The FPGA hardware includes a communication module and a JPEG encoder that includes a controller for the camera. The JPEG encoder was adopted from (Krepa,) and adapted to suit our design.

4 IMPLEMENTATION

To evaluate our solution a complete system prototype was implemented. The system prototype consists of a remote control, a personal computer, and a microdrone extended with a security module.

The microdrone was prototype as a quadrotor based on the Next-Generation Universal Aerial Video Platform (NG-UAVP). The NG-UAVP is “a community-driven open source project to build a modern autonomously flying multicopter” (ANG-UAVP,). The NG-UAVP has a special focus on expandability and configurability with a proprietary near real-time operating system and a hardware abstraction layer that enables the platform to support different hardware types and configurations. The NG-UAVP uses three microcontrollers for data acquisition and preparation on the sensor’s board and for flight control.

The quadrotor is extended with the security module and a camera module. The camera used is the TCM8240MD from Toshiba with an image resolution of 640x480 pixel, a frame rate of 30 FPS, and a color depth of 24 bit. One of the selection criteria of this camera was its small size of 6x6x4.5 mm. The camera has an Inter-Integrated Circuit (I^2C) interface for configuration and delivers pixel data over an 8-bit parallel interface along with two lines for vertical and horizontal synchronization.

The security module is compatible with the other NG-UAVP boards. It contains mainly a communication module, an FPGA, and a flash memory for programming the FPGA. The used communication module (from the company Avisaro AG) is connected to the FPGA through a serial peripheral interface (SPI), which allows a maximum data rate of 2.8 MBit/s.

As an FPGA we use Spartan E from Xilinx. Image processing demands large memory size. The used open-source JPEG encoder works on 8x8-pixel blocks. However, data is read from the camera row by row. Thus, for the JPEG encoder to start, at least seven rows and the first 8 pixels of the 8-th row must

Table 1: Image Parameters used in the Prototype Implementation.

Parameter	Value	Unit
Image resolution	640x480	pixel
Color depth	24	bit
Frame size before encoding	7372800	bit
Compression ratio	15	%
Frame size after encoding	491520	bit
No. AES blocks per frame	3840	block

be read first. With 640 pixel per row and 24-bit color depth a minimum memory space of $(7 * 640 + 8) * 24 = 108$ kbit is required. To increase throughput, the JPEG encoder uses pipelined processing of 16 rows which demands a total memory usage of 163.840 kbit. Additional memory space is required for the encoder output buffer, for the AES Sboxes, and for the input and output FIFOs of the video streamers.

Instead of using external memory for data storage we decided to use the largest member of the Spartan E family. By this means, the design is kept simpler and data are transferred synchronously between the different FPGA components, which is essential for the performance. The FPGA works at 50 MHz clock frequency that is supplied by an external oscillator. The FPGA is configured using a SPI serial flash PROM (Programmable read-only memory) from Atmel.

5 RESULTS AND ANALYSIS

The hardware prototype was tested for the data given in Table 1. The raw frame size results from multiplying the image resolution by the color depth. Through encoding, the frame size is reduced 15 times. The encoded frame is divided by 128 to get the number of AES blocks that need to be processed for one frame.

Table 2 shows a comparison between the hardware solution and a comparable software solution running on the drone microcontroller ARM7 LPC2148 clocked at 60 MHz. The comparison metrics are the timing, the power consumption, and flight time.

With respect to timing, the developed FPGA solution provides an approximately 20-time higher throughput than the software solution. The software solution only achieves a frame rate of 3.2 fps, which is clearly below acceptable rates for live feed, and therefore fails to deliver a moving picture.

With respect to power, the consumption of the hardware solution exceeds the consumption of the software solution by 3.356 watts. This amount comprises three components:

1. The power consumption of the FPGA chip as provided by the Xilinx Xpower Analyzer (0.22 watts).
2. The measured electrical power consumed by the FPGA board exclusive of the FPGA chip (0.64 watts)
3. The mechanical power needed to raise the additional weight caused by the FPGA card (2.5 watt). This value was determined using a commercial calculator as detailed below.

By comparing the pure computational power consumption, the FPGA with 0.22 watt is superior to the software solution that consumes 1.8 watt on ARM7 LPC2148. This is in line with the general understanding of FPGAs as power-efficient alternatives to software. This is valid even when we compare with the power consumed by the entire FPGA board ($0.64 + 0.22 = 0.88$ watt).

In this application, however, we have to consider the power needed to carry the weight of the FPGA board. Apparently, this part is significant and it contributes to shortening the flight time by almost 1.5%. The latter figure is based on the voltage and the capacity of the battery used in our model. These are 14 volts and 2300 mAh, respectively.

Note, however, that the actual impact on the flight time should be considered in the light of the required throughput. The advantage of the software solution is only valid when 3.2 FPS or less are required. As stated before, this rate is not sufficient for motion pictures.

To determine the power needed to carry the FPGA we used eCalc which is a commercially available online tool that provides calculations for electrically driven remote controlled devices (Mller,). The vendor states that the calculation accuracy is 10%. eCalc provides four different tools for RC model calculation: propeller airplanes, multicopters, ducted fan airplanes and helicopters. eCalc can be used for free however with limited capability in terms of supported models and components as well as provided output data. For our calculations we used the commercial edition of eCalc to be as specific as possible in the data entry.

6 CONCLUSIONS

The paper presented a security analysis for civil microdrone communication. Based on this analysis technical solutions were proposed, implemented using an FPGA platform, and embedded into a self-constructed quadrotor. The hardware solution shows

Table 2: A HW-SW Comparison of the Image Coding (Encryption and Authentication).

Compared Value	Software	Hardware	Unit
Time needed to encode one JPEG frame	100	14.4	msec
Time needed to encrypte and authenticate one JPEG frame	209.6	1.54	msec
Time needed to encode, encrypt and authenticate one JPEG frame	309.6	15.9	msec
Throughput	3.2	62.8	fps
Measured prone power consumption while hovering	220	223.36	watt
Estimated flight time under maximum possible throughput	9.03	8.89	minute

as expected a clear throughput advantage when compared to a similar software implementation. In contrast, the power consumption of the hardware solution exceeds that of the software alternative due to the extra weight of the FPGA board. It can be assumed that this aspect gets more relevant for low-weight microdrones that need to meet security requirements. Specifically, such vehicles have very limited computational power, as a rule, and are expected to fail to support cryptographic functions. While enhancing these vehicles with cryptoprocessors can solve the performance issue, the relative weight of the additional hardware can significantly worsen the power consumption of the device and thus its flight time. The specification and the modeling of this aspect is an interesting research question that will be addressed in future work.

REFERENCES

ANG-UAVP. Next generation universal aerial video platform. *http://ng.uavp.ch/moin/FrontPage*.

Bicakci, K. and Tavli, B. (2009). Denial-of-service attacks and countermeasures in ieee 802.11 wireless networks. *Computer Standards & Interfaces*, 31(5):931–941.

Chiu, C. and Lo, C. (2011). Vision-only automatic flight control for small uavs. *Vehicular Technology, IEEE Transactions on*, 60(6):2425–2437.

Daemen, J. and Rijmen, V. (2002). *The design of Rijndael: AES–the advanced encryption standard*. Springer-Verlag New York Inc.

Dierks, T. and Jagannathan, S. (2010). Output feedback control of a quadrotor uav using neural networks. *Neural Networks, IEEE Transactions on*, 21(1):50–66.

Gancet, J., Hattenberger, G., Alami, R., and Lacroix, S. (2005). Task planning and control for a multi-uav system: architecture and algorithms. In *Intelligent Robots and Systems, 2005.(IROS 2005). 2005 IEEE/RSJ International Conference on*, pages 1017–1022. Ieee.

Hoffmann, G., Rajnarayan, D., Waslander, S., Dostal, D., Jang, J., and Tomlin, C. (2004). The stanford testbed of autonomous rotorcraft for multi agent control (starmac). In *Digital Avionics Systems Conference, 2004. DASC 04. The 23rd*, volume 2, pages 12–E. Ieee.

How, J., Bethke, B., Frank, A., Dale, D., and Vian, J. (2008). Real-time indoor autonomous vehicle test environment. *Control Systems Magazine, IEEE*, 28(2):51–64.

Katz, J. and Lindell, Y. (2008). *Introduction to modern cryptography*. Chapman & Hall.

Krepa, M. Jpeg encoder, project, mkjpeg. *http://opencores.org/*.

Minguez, J. and Montano, L. (2004). Nearness diagram (nd) navigation: collision avoidance in troublesome scenarios. *Robotics and Automation, IEEE Transactions on*, 20(1):45–59.

Mller, M. ecalc, on-line calculator for electric driven rc models. *http://www.ecalc.ch/*.

NIST (1985). American national standard for financial institution key management (wholesale). Available at *http://csrc.nist.gov/publications/fips/*.

Pelechrinis, K., Iliofotou, M., and Krishnamurthy, V. (2011). Denial of service attacks in wireless networks: The case of jammers. *Communications Surveys & Tutorials, IEEE*, (13):245–257.

Quaritsch, M., Stojanovski, E., Bettstetter, C., Friedrich, G., Hellwagner, H., Rinner, B., Hofbaur, M., and Shah, M. (2008). Collaborative microdrones: applications and research challenges. In *Proceedings of the 2nd International Conference on Autonomic Computing and Communication Systems*, page 38. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering).

Voos, H. (2006). Nonlinear state-dependent riccati equation control of a quadrotor uav. In *Computer Aided Control System Design, 2006 IEEE International Conference on Control Applications, 2006 IEEE International Symposium on Intelligent Control, 2006 IEEE*, pages 2547–2552. IEEE.

Yuan, C., Recktenwald, F., and Mallot, H. (2009). Visual steering of uav in unknown environments. In *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*, pages 3906–3911. IEEE.

Zsedrovits, T., Zarandy, A., Vanek, B., Peni, T., Bokor, J., and Roska, T. (2011). Collision avoidance for uav using visual detection. In *Circuits and Systems (ISCAS), 2011 IEEE International Symposium on*, pages 2173–2176. IEEE.