Dronecrypt - An Ultra-Low Energy Cryptographic Framework for Small Aerial Drones

Muslum Ozgur Ozmen* Attila A. Yavuz*

Abstract

Aerial drones are becoming an integral part of application domains such as package delivery, construction, monitoring and search/rescue operations. It is critical to ensure the cyber-security of networked aerial drone systems in these applications. Standard cryptographic services can be deployed to provide basic security services; however, they have been shown to be highly energy costly for small aerial drones. Therefore, there is a significant need for a highly energy efficient cryptographic framework that can meet the requirements of small aerial drones.

In this paper, *we propose a new cryptographic framework for small aerial drones, which offers significant energy efficiency and speed advantages over standard cryptographic techniques*: (i) We create (to the best of our knowledge), the first highly optimized public key infrastructure (PKI) based framework for small aerial drones, which provides energy efficient digital signature and public key encryption techniques by harnessing special pre-computation methods and optimized elliptic curves. (ii) We also integrate some of the most recent lightweight symmetric primitives into our PKI techniques to provide a full-fledged cryptographic framework. (iii) We implemented standard counterparts and our proposed techniques on an actual small aerial drone (Crazyflie 2.0), and provided an in-depth energy consumption analysis. Our experiments showed that *our new cryptographic framework achieves up to 35× lower energy consumption than its standard counterpart*. To the best of our knowledge, this is the first realization of an ultra-light cryptographic framework targeting small aerial drones. We make our framework open-source for public testing and adaptation purposes.

1 Introduction

Aerial drones are emerging mobile cyber-physical systems with potential applications including but not limited to, package delivery, reconnaissance, environmental monitoring and disaster recovery/response [1, 2] missions. Due to the significant financial and strategic value involved, aerial drones are expected to be exclusively targeted by attackers. Therefore, it is critical to ensure the cyber security of aerial drone systems. In particular, the basic security services such as confidentiality, authentication, and integrity must be provided. These services are mainly guaranteed via fundamental cryptographic techniques (e.g., symmetric ciphers, PKI).

Although there are various standard cryptographic techniques, the vast majority of them may not meet the needs of aerial drones. Strict energy and bandwidth constraints pose critical limitations towards the deployment of standard cryptographic techniques on small aerial drones [3]. In the following, we outline the state-of-the-art cryptographic solutions that are considered for

*Oregon State University, {ozmenmu,attila.yavuz}@oregonstate.edu

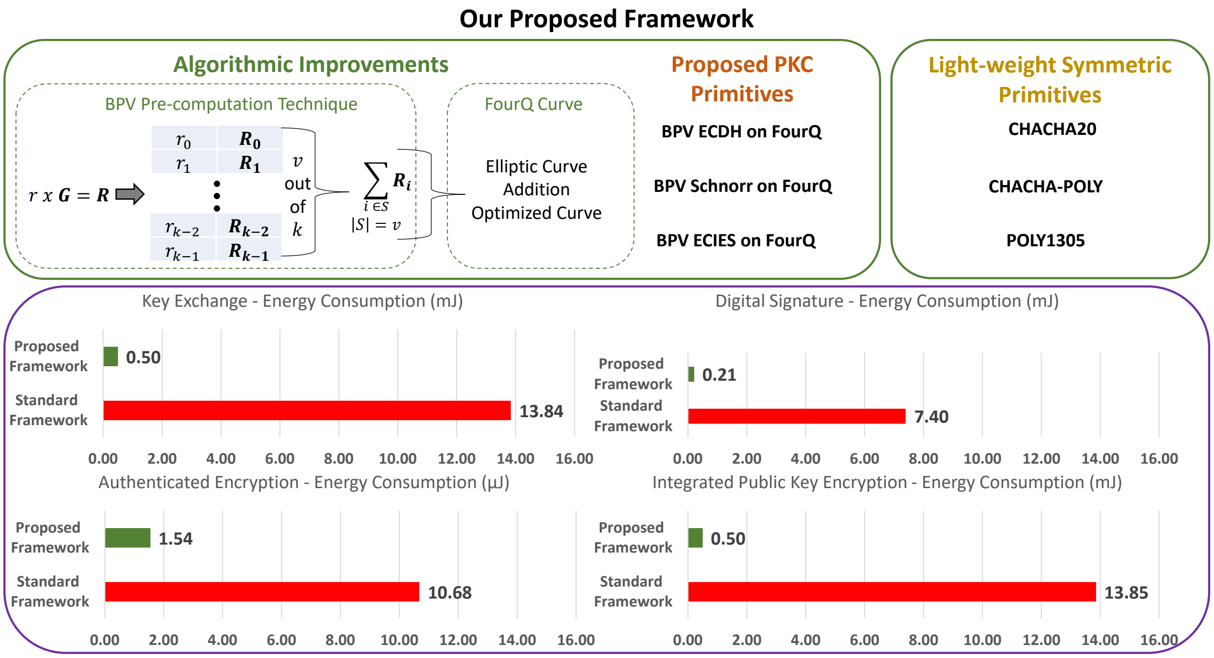

Figure 1: The performance comparison of the proposed and standard cryptographic frameworks.

aerial drones, and highlight a critical research gap to be addressed towards achieving practical deployments for resource-limited aerial drone systems.

1.1 Related Work and Research Gap

The development of cryptographic techniques specifically tailored for aerial drones only recently received slight attention from the academic community. Currently, there are two main lines of works:

Standard Cryptographic Techniques: Initial approaches to secure the aerial drones include implementing the well-known protocols such as RSA and AES on FPGAs [4, 5]. However, they showed that standard techniques consume high amount of energy on small aerial drones. Our experiments further confirmed this (see Section 4).

Certificataless Cryptography and White-Box Approaches: Seo et al. in [6] proposed a security framework that implemented some symmetric ciphers with white-box cryptography to mitigate the impacts of drone capture attacks [6]. However, the insecurity of this line of research has been later shown in [7]. Won et. al. proposed Certificateless Signcryption based protocols for mid-size aerial drones (e.g., AR.Drone 2.0) in [8]. This protocol reduces the communication overhead by eliminating certificates, but requires several exponentiations and therefore introduces heavy overhead, which might not be practical for resource-limited small aerial drones. Moreover, aerial drones are expected to be an integral part of Internet of Things (IoT), which vastly relies on PKI technology. Certificateless protocols cannot be seamlessly integrated into existing PKI-based IoT systems without significant alterations.

Research Gap: One may observe the following research gaps that must be addressed towards a practical deployment of cryptographic techniques on small aerial drones: (i) The existing standard primitives and cryptographic protocols are extremely energy costly for battery-limited small aerial drones. (ii) The existing cryptographic protocols for drones only focus on a few specific primitives and do not offer a comprehensive framework that harbors a vast variety of light-weight symmetric and asymmetric primitives. (iii) There is no open-source framework, even for some standard cryptographic primitives, which offers a detailed energy assessment of the cryptographic primitives on small aerial drones.

1.2 Our Contribution

Towards filling the aforementioned research gap, we propose a novel cryptographic framework by harnessing various cryptographic primitives and optimizations, which provide significantly lower energy consumption for small aerial drones compared to the deployment of standard cryptographic techniques. We further outline our contributions as follows:

• *A New Ultra-Low Energy Cryptographic Framework with Algorithmic Improvements*: Our objective is to reduce the computation and communication overhead of cryptographic primitives to minimize their energy consumption for small aerial drones. Therefore, we exploit synergies among special pre-computation techniques and elliptic curves (EC), which not only offer the most compact key/signature sizes among available alternatives, but also significantly reduce the computational cost of EC scalar multiplication.

We develop (to the best of our knowledge) the first realization of Boyko-Peinado-Venkatesan (BPV) pre-computation technique [9] on FourQ curve [10]. BPV technique reduces the cost of an Elliptic EC scalar multiplication to only a few EC additions with only a small constant-size storage overhead. Remark that FourQ curve is one of the most addition friendly curves (even with a better efficiency than that of Curve25519 [11]), and therefore our integration further enhances the efficiency of BPV. We then instantiate *BPV-FourQ-Schnorr* and *BPV-FourQ-ECIES* as our improved digital signature and integrated encryption schemes, which significantly outperform their standard counterparts. Moreover, we also integrate some of the most recent light-weight symmetric primitives into our new PKI suite to create a full-fledged cryptographic framework.

• *In-depth Energy Analysis*: The energy consumption of standard Elliptic Curve Cryptography (ECC) based primitives have not been investigated thoroughly for recently emerging small aerial drones. In this paper, we implemented both standard ECC techniques and our proposed cryptographic framework with algorithmic improvements and presented a detailed energy consumption analysis. As demonstrated in Figure 1 for different cryptographic primitives, our experiments showed that proposed improvements enable approximately 35× less energy consumption compared to the standard techniques for small aerial drones (see Section 4). Similar performance gains were observed for the light-weight ciphers over the standard symmetric primitives.

• *Open-Source Framework*: While isolated implementation results were reported for particular cryptographic primitives such as RSA and AES [4, 5], to the best of our knowledge, no comprehensive open-source cryptographic framework is available for small aerial drones. Towards meeting this need, we fully implemented our standard and improved cryptographic frameworks. Moreover, we open-source both to enable a broad test and potential adaptation[1].

2 Preliminaries

We first outline the notation in Table 1 and then describe our building blocks.

[1]https://github.com/ozgurozmen/Dronecrypt

Table 1: Notation followed to describe schemes.

F_q	Finite Field
$\mathbf{G}$	Generator Group Point
n	Order of Group
$\langle y, \mathbf{Y} \rangle$	Private/Public key pair
Γ	BPV Precomputation Table
m	Message
$\mathcal{E}_k$	IND-CPA Encryption via key k
$\mathcal{D}_k$	IND-CPA Decryption via key k
$\times$	Elliptic Curve Scalar Multiplication (Emul)
KDF	Key Derivation Function

Elliptic Curve (EC) points are shown in bold

BPV Pre-computation Technique: We use BPV generator [9], which reduces the computational cost of an Emul to a few EC additions with the expense of a table storage. The *BPV* generator is a tuple of two algorithms (*Offline*, *Online*) defined in Algorithm 1.

Algorithm 1 Boyko-Peinado-Venkatesan (BPV) Generator

$(\Gamma, v, k, F_q, \mathbf{G}, n) \leftarrow BPV.Offline(1^\kappa)$:

1: Set the EC system-wide parameters *params* $\leftarrow (F_q, \mathbf{G}, n)$. We suppress *params* afterwards for the sake of brevity.
2: Generate *BPV* parameters (v, k), where k and v are the number of pairs to be pre-computed and the number of elements to be randomly selected out k pairs, respectively, for $2 < v < k$.
3: $r'_i \stackrel{\$}{\leftarrow} \mathbb{Z}_n^*$, $\mathbf{R'_i} \leftarrow r'_i \times \mathbf{G}$, $i = 0, \ldots, k-1$.
4: Set pre-computation table $\Gamma = \{r'_i, \mathbf{R'_i}\}_{i=0}^{k-1}$.

$(r, \mathbf{R}) \leftarrow BPV.Online(\Gamma, v, k, F_q, \mathbf{G}, n)$:

1: Generate a random set $S \subset [0, k-1]$, where $|S| = v$.
2: $r \leftarrow \sum_{i \in S} r'_i \bmod n$, $\mathbf{R} \leftarrow \sum_{i \in S} \mathbf{R'_i}$.

FourQ Curve: FourQ is a high-security, high-performance elliptic curve that is proposed by Costello et. al [10]. FourQ synergizes some well-known EC optimizations to offer high-speed EC scalar multiplication and EC addition while preserving 128-bit security.

Standard Techniques: Our standard cryptographic framework consists of broadly standardized techniques.

We select secp256k1 curve to implement our standard public key cryptography services, such as ECDH [12], ECDSA [13], and ECIES [14]. Secp256k1 is a NIST recommended curve [15], which is frequently used in practice. We implemented a key exchange, a signature and an integrated scheme to secure a variety of applications. We also implemented some of the most well-known symmetric key cryptography techniques such as AES as the block cipher [16], AES-GCM as the authenticated encryption [17] and HMAC with SHA-256 for MAC. Note that, an open-source implementation and in-depth energy analysis of such standard cryptographic framework for small aerial drones are not currently available (to the best of our knowledge).

3 Proposed Cryptographic Framework

3.1 An Ultra-Low Energy PKI Cryptographic Framework

One may consider adopting traditional pre-computation methods to reduce the computation overhead of standard cryptographic techniques. However, traditional online/offline pre-computation methods require linear storage and therefore they may not be feasible for small aerial drones equipped with lightweight microcontrollers. In addition, it is shown that the regeneration cost of these pre-computation techniques may incur more computation cost than following the original protocol [18]. Hence, there is a need for an approach that can reduce the cost of EC scalar multiplication without incurring heavy storage overhead.

Main Idea: We propose an integrated approach that only requires small-constant storage overhead and highly optimized EC addition operations as opposed to a full EC scalar multiplication. To achieve this objective, we present the first (to the best of our knowledge) instantiation of BPV on FourQ curve, that is ideal for BPV due to its very fast EC additions. With BPV, we reduce EC scalar multiplication operations of our target schemes/protocols to only a few EC addition operations. The integration of FourQ into BPV amplifies the performance gain as it is an addition efficient curve. As it will be showcased in Section 4, this strategy enables almost 35× more efficient operations than its standard schemes (with standard curves such as secp256k1). The storage cost of such an energy efficiency gain is only a constant-size storage of 64KB (for parameters $v = 16, k = 1024$, see Section 2) keying material. With the current capabilities of low-end processors, even on the small aerial drones (such as Crazyflie 2.0 [3]), it is feasible to store 65KB. We then use BPV-FourQ to instantiate two main cryptographic schemes, which creates the following efficient digital signature and integrated encryption suite:

(i) We integrate our optimized BPV-FourQ into Schnorr signature scheme to gain computational efficiency. This transformation is presented in Algorithm 2.

Algorithm 2 BPV-FourQ-Schnorr Signature

$(\Gamma, y, \mathbf{Y}) \leftarrow \mathit{BPV\text{-}FourQ\text{-}Schnorr.Kg}(1^\kappa)$:

1: Generate parameters and BPV table as $(\Gamma, v, k, F_q, \mathbf{G}, n) \leftarrow \mathit{BPV.Offline}(1^\kappa)$.

2: Generate private/public key pair $(y \xleftarrow{\$} \mathbb{Z}_n^*, \mathbf{Y} \leftarrow y \times \mathbf{G})$

$(s, e) \leftarrow \mathit{BPV\text{-}FourQ\text{-}Schnorr.Sig}(m, y, \Gamma)$:

1: $(r, \mathbf{R}) \leftarrow \mathit{BPV.Online}(\Gamma, v, k, F_q, \mathbf{G}, n)$

2: $e \leftarrow H(m||\mathbf{R}),\quad s \leftarrow (r - e \cdot y) \bmod n$ where H is a full domain cryptographic hash function $H : \{0,1\}^* \rightarrow \mathbb{Z}_n^*$.

$b \leftarrow \mathit{BPV\text{-}FourQ\text{-}Schnorr.Ver}(m, \langle s, e\rangle, \mathbf{Y})$:

1: $\mathbf{R}' \leftarrow e \times \mathbf{Y} + s \times \mathbf{G}$

2: If $e = H(m||\mathbf{R}')$ then set $b = 1$ as *valid*, else $b = 0$.

(ii) Similarly, we instantiate ECDH and ECIES with BPV-FourQ. In the ECDH variant, each node first derives its private/public key pair with an EC scalar multiplication, and then one more EC scalar multiplication is performed in order to obtain the shared key [12]. We adopt BPV protocol so that each node derives its private/public key pair with only EC additions. This decreases the computation time and energy consumption of the protocol by almost 1.5×, in the expense of storing a 64KB table. Moreover, it is almost 28× more efficient than its standard counterpart (ECDH on secp256k1, see section 4). In ECIES protocol, node 1 first derives the shared secret by generating an ephemeral private/public key pair and then computing the shared secret using node 2's public key [14]. Then, she generates encryption and MAC keys from this shared secret with a pre-determined key derivation function. We adopt BPV to ECIES by transforming the first EC scalar multiplication (where the ephemeral private/public key pair is generated) to EC additions, as depicted in Algorithm 3.

Algorithm 3 BPV-FourQ-ECIES Encryption

$\underline{(\Gamma, y, \mathbf{Y}) \leftarrow \textit{BPV-FourQ-ECIES.Kg}(1^\kappa)}$:

1. Generate parameters and BPV table as $(\Gamma, v, k, F_q, \mathbf{G}, n) \leftarrow \textit{BPV.Offline}(1^\kappa)$.
2. Generate private/public key pair $(y \stackrel{\$}{\leftarrow} \mathbb{Z}_n^*, \mathbf{Y} \leftarrow y \times \mathbf{G})$

$\underline{(c, d, \mathbf{R}) \leftarrow \textit{BPV-FourQ-ECIES.Enc}(m, Y, \Gamma)}$:

1. $(r, \mathbf{R}) \leftarrow \textit{BPV.Online}(\Gamma, v, k, F_q, \mathbf{G}, n)$
2. Generate shared secret as $\mathbf{T} \leftarrow r \times \mathbf{Y}$
3. $(k_{enc}, k_{MAC}) \leftarrow KDF(\mathbf{T})$
4. $c \leftarrow \mathcal{E}_{k_{enc}}(m)$
5. $d \leftarrow MAC_{k_{MAC}}(c)$

$\underline{m \leftarrow \textit{BPV-FourQ-ECIES.Dec}(y, \langle c, d, R \rangle)}$:

1. $\mathbf{T}' \leftarrow y \times \mathbf{R} \bmod p$
2. $(k_{enc}, k_{MAC}) \leftarrow KDF(\mathbf{T}')$
3. If $d \neq MAC_{k_{MAC}}(c)$ return INVALID
4. $m \leftarrow \mathcal{D}_{k_{enc}}(c)$

3.2 Ultra-low Energy Symmetric Key Primitives

We improve standard symmetric primitives by getting advantage of lightweight and faster ciphers, and authentication protocols that were not available in existing drone implementations and frameworks. We leverage CHACHA20 as a very fast stream cipher [19], CHACHA-POLY as the authenticated encryption and POLY1305 as the MAC protocol [20]. These schemes provide faster and more energy-efficient encryption/authentication while still offering high-security guarantees [21]. Our experiments confirmed that there are significant improvements for computation time and energy consumption with the adoption of these lightweight and high-speed symmetric primitives. More specifically, standard symmetric primitives are improved up to 7× (see Section 4).

4 Performance Evaluation

4.1 Experimental Setup and Evaluation Metrics

We worked on Crazyflie 2.0 (Figure 2) due to its open-source software and hardware [3]. Crazyflie 2.0 has two microcontrollers: (i) An STM32F405 microcontroller as the main controller that runs all the flight control codes. Both cryptographic frameworks were implemented on this microcontroller due to its capabilities. (ii) An ultra light-weight nRF51822 microcontroller that is responsible for the communication (radio) and power management.

STM32F405 is equipped with an ARM Cortex M-4 architecture and operates at 168 MHz. It is a 32-bit microcontroller with a 192KB SRAM and 1MB flash memory. Although STM32F405 is a resourceful processor, it has very low power consumption. It operates at $3.3V$ and takes $40mA$ of current while operating at 168MHz clock speed. Our evaluation metrics include computation time, storage, communication bandwidth, and energy consumption. We measured the energy consumption with the formula $E = V \cdot I \cdot t$, where $V = 3.3V$, $I = 40mA$ [22] and t is the computation time based on the clock cycles. Moreover, we connected an ammeter between the device and the battery to double check the current taken by the processor and observed insignificant difference with $40mA$.

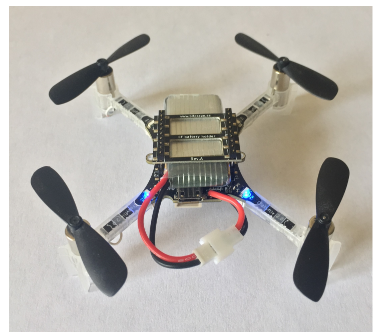

Figure 2: Crazyflie 2.0

We implemented all the schemes with the following libraries. (i) WolfCrypt [23] - All symmetric primitives are implemented using WolfCrypt which is an open-source and easily accessible library. (ii) microECC [24] - We implemented all standard public key services using microECC as it is a very light-weight and open-source library which supports various NIST recommended curves. (iii) FourQ [10] - All optimized public key primitives with algorithmic improvements are implemented with open-source Microsoft FourQ library [10]. We selected BPV parameters as $v = 16$ and $k = 1024$, which sets the size of Γ as 64KB, that can be easily stored in STM32F405. We should note that all libraries used for both standard and optimized frameworks are open-source, therefore they can be easily integrated to various aerial drones.

4.2 Performance/Energy Evaluation and Comparison

Experimental evaluation and comparison of the standard framework and our proposed framework are depicted in Tables 2 and 3, for public key and symmetric cryptography, respectively.

Although energy consumption is critical for small aerial drones, computation time also has some crucial effects on the adaptation of cryptography. For instance, there are time-critical applications that need frequent data transmission (e.g. camera mounted on an aerial drone for video streaming - 24 frames per second are necessary). Cryptographic primitives used to secure such applications should meet this demand by offering high-speed operations. Therefore, we believe CPU Time improvements depicted in Tables 2 and 3 are also critical for small aerial drones.

Below, we present the comparison between the standard framework and our proposed framework with algorithmic improvements with an application to use-cases for small aerial drones.

• *Digital Signature and Broadcast Authentication*: Digital signatures are commonly used for broadcast authentication and key certification purposes. Some drone use-cases may contain

Table 2: Comparison of Standard and Optimized Public Key Cryptographic Frameworks

Protocol	CPU Cycles	CPU Time (ms)	Memory¶ (Byte)	Bandwidth (Byte)	Energy Consumption (mJ)
Standard Cryptographic Framework					
ECDH	17611413	104.83	32	32	13.84
ECDSA-Sign	9411839	56.02	32	64	7.40
ECDSA-Verify	8169117	48.62	32	64	6.42
ECIES-Encrypt	17625012	104.91	32	32 + C†	13.85
ECIES-Decrypt	8817657	52.49	32	32 + C†	6.93
Optimized Framework with Algorithmic Improvements‡					
BPV ECDH	**636833**	**3.79**	65536	32	**0.50**
BPV Schnorr-Sign	**264011**	**1.57**	65536	64	**0.21**
BPV Schnorr-Verify	**683882**	**4.07**	32	64	**0.54**
BPV ECIES-Encrypt	**638791**	**3.80**	65536	32 + C†	**0.50**
BPV ECIES-Decrypt	**513487**	**3.06**	32	32 + C†	**0.40**

¶ Memory denotes the private key size for sign/encrypt schemes as signer/sender stores it, memory denotes the public key size for verify/decrypt schemes as verifier/receiver stores it.
† C denotes the length of ciphertext and Message Authentication Code.
‡ These primitives are implemented on EC Addition efficient FourQ curve

Table 3: Comparison of Standard and Optimized Symmetric Cryptographic Frameworks

Protocol	(MB/s)	CPU Cycles¶	CPU Time (μs)	Energy Consumption (μJ)
Standard Cryptographic Framework				
AES	0.926	5537	32.96	4.35
AES-GCM	0.377	13599	80.95	10.68
HMAC†	3.338	1536	9.14	1.21
Optimized Framework with Algorithmic Improvements				
CHACHA20	**3.554**	**1443**	**8.59**	**1.13**
CHACHA-POLY	**2.619**	**1958**	**11.65**	**1.54**
POLY1305	**13.709**	**374**	**2.23**	**0.29**

¶ CPU cycles presented here are for a 32-byte message.
† SHA256 is used as the standard hash function for HMAC.

scenarios where the drone needs to broadcast its sensor data such as GPS data for location, photo frames for monitoring and temperature/pressure data for meteorological observation. As there can be multiple servers that need to authenticate the confidentiality of the data in these use-cases, digital signatures should be preferred due to its scalability, public verifiability, and non-repudiation. Moreover, certificates are necessary to protect key exchange schemes from man-in-the-middle attacks. Digital signatures are extensively used in practice for certification. Therefore, whenever a drone makes a key exchange with a server or another drone, certificate verification (digital signature verification) must be performed.

Our proposed framework makes significant improvements over the standard framework. As depicted in Figure 1, energy consumption of digital signature is decreased to 0.21mJ from 7.40mJ with a 35.24× improvement. Moreover, the maximum signing throughput of the standard framework is 17 signatures per second, that may not be sufficient for some use-cases.

- *Integrated Public Key Encryption and Key Exchange*: Although public key encryption is very costly compared to symmetric key encryption, there are various use-cases that it should be preferred. These use-cases mainly include the configurations where multiple drones need to report to a single server. For instance, CrazySwarm [25] consists of 49 drones that work cooperatively with each other and used for object tracking, where all these drones report to a single server. We believe public key encryption can be preferred for applications such as tracking and search/rescue operations. Moreover, our preferred public key encryption scheme, ECIES, provides *forward security*, that is crucial for safety critical applications.

In addition, a key exchange protocol is essential for drones for the management/distribution of symmetric keys. We cannot assume pre-installed keys for most of the applications as the controller/server may change for different purposes and use-cases. Therefore, an efficient key exchange protocol is necessary to be deployed on drones.

Our optimized framework with algorithmic improvements achieves significantly lower energy consumption and faster encryption/key exchange than the standard framework. BPV-FourQ-ECIES is 27.70× more energy efficient than the standard ECIES protocol. Moreover, BPV-FourQ-ECDH also improved its standard counterpart by 27.68×. Besides the energy efficiency, our optimized protocols offer significantly faster computation time which is essential for time-critical applications.

• *Light-weight Symmetric Primitives*: Symmetric key cryptography can be used for various use-cases on drones, due to their low energy consumption. For instance, securing command and control channel is the minimum requirement for a safe operation of any aerial drone system. This channel can be easily secured via symmetric key primitives. We suggest using an authenticated encryption to secure this channel to prevent any kind of attacks from the communication channel that aims to take control over the aerial drone. Moreover, when a single aerial drone reports its sensor data to one or a few base stations, symmetric key primitives can be preferred over digital signature/public key encryption schemes. Therefore, we believe that adopting lightweight symmetric primitives is a significant improvement for aerial drones.

Although the energy consumption of symmetric primitives is minimal compared to the public key primitives, it is essential to optimize the energy consumption as the frequency of encryption/authentication needs to be much higher. 100 messages per second are necessary to have a stable flight, which means 100 encryption/authentication operations per second [3]. When authenticated encryption schemes are used to secure this channel as suggested, our proposed framework offers 6.95× lower energy consumption. Our proposed framework still achieves 3.84× and 4.11× lower energy consumption for encryption and authentication, respectively. Therefore, our proposed framework offers significant energy improvement over its standard counterpart for symmetric key primitives.

Discussions: We showed that our optimized framework provides significant improvements over its standard counterpart, in terms of both energy consumption and computation time. We believe our optimized framework can meet the demands of various aerial drone configurations with different needs. Hence, it should be adopted to these configurations to minimize the energy overhead of cryptography. Moreover, computation time also poses a critical limitation for some aerial drone configurations. Signing throughput with our framework can be as high as 636 messages per second and forward secure public-key encryption throughput can be as high as 263 messages per second. Although for most use-cases such high throughputs are not necessary, we show that our framework can support up to these numbers for some extreme cases. Therefore we believe our framework can meet the demands of applications where very fast authentication/encryption is necessary.

5 Conclusion

In this paper, we show that standard cryptographic techniques may not meet with the demands of energy and computational efficiency requirements on small aerial drones. Our proposed framework meets these requirements and includes a vast range of cryptographic techniques that can be used to secure numerous aerial drone applications and configurations. Our proposed framework with algorithmic improvements achieves up to 35× higher speed and better energy efficiency when compared to its standard counterpart. Moreover, we put the engineering effort and to fully implement and present the (to the best of our knowledge) first full-fledged open-source framework with algorithmic improvements targeting small aerial drones.

References

[1] R. Clarke, "Understanding the drone epidemic," *Computer Law & Security Review*, vol. 30, no. 3, pp. 230 – 246, 2014. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0267364914000545

[2] M. Saska, T. Krajník, J. Faigl, V. Vonásek, and L. Peuil, "Low cost mav platform ar-drone in experimental verifications of methods for vision based autonomous navigation," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Oct 2012, pp. 4808–4809.

[3] A. Bitcraze, "Crazyflie 2.0 nano quadcopter," 2016. [Online]. Available: http://www.bitcraze.io

[4] J. A. Steinmann, R. F. Babiceanu, and R. Seker, "Uas security: Encryption key negotiation for partitioned data," in *2016 Integrated Communications Navigation and Surveillance (ICNS)*, April 2016, pp. 1E4–1–1E4–7.

[5] A. Shoufan, H. AlNoon, and J. Baek, *Secure Communication in Civil Drones*. Cham: Springer International Publishing, 2015, pp. 177–195. [Online]. Available: https://doi.org/10.1007/978-3-319-27668-7_11

[6] S.-H. Seo, J. Won, E. Bertino, Y. Kang, and D. Choi, "A security framework for a drone delivery service," in *Proceedings of the 2Nd Workshop on Micro Aerial Vehicle Networks, Systems, and Applications for Civilian Use*, ser. DroNet '16. ACM, 2016, pp. 29–34.

[7] J. W. Bos, C. Hubain, W. Michiels, and P. Teuwen, *Differential Computation Analysis: Hiding Your White-Box Designs is Not Enough*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2016, pp. 215–236. [Online]. Available: https://doi.org/10.1007/978-3-662-53140-2_11

[8] J. Won, S. H. Seo, and E. Bertino, "Certificateless cryptographic protocols for efficient drone-based smart city applications," *IEEE Access*, vol. 5, pp. 3721–3749, 2017.

[9] V. Boyko, M. Peinado, and R. Venkatesan, "Speeding up discrete log and factoring based schemes via precomputations," in *Advances in Cryptology — EUROCRYPT'98: International Conference on the Theory and Application of Cryptographic Techniques Espoo, Finland, May 31 – June 4, 1998 Proceedings*. Berlin, Heidelberg: Springer Berlin Heidelberg, 1998, pp. 221–235.

[10] C. Costello and P. Longa, *FourQ: Four-Dimensional Decompositions on a Q-curve over the Mersenne Prime*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2015, pp. 214–235. [Online]. Available: http://dx.doi.org/10.1007/978-3-662-48797-6_10

[11] D. J. Bernstein, *Curve25519: New Diffie-Hellman Speed Records*. Springer Berlin Heidelberg, 2006, pp. 207–228. [Online]. Available: http://dx.doi.org/10.1007/11745853_14

[12] W. Diffie and M. Hellman, "New directions in cryptography," *IEEE Transactions on Information Theory*, vol. IT-22, pp. 644–654, November 1976.

[13] *ANSI X9.62-1998: Public Key Cryptography for the Financial Services Industry: The Elliptic Curve Digital Signature Algorithm (ECDSA)*, American Bankers Association, 1999.

[14] D. Pointcheval, "Psec-3: Provably secure elliptic curve encryption scheme," 2000.

[15] National Institute of Standards and Technology, "Recommended elliptic curves for federal government use," August 1999.

[16] NIST, "Announcing the advanced encryption standard (AES)," FIPS 197, November 2001.

[17] D. A. McGrew and J. Viega, "The security and performance of the galois/counter mode of operation (full version)," Cryptology ePrint Archive, Report 2004/193, 2004, http://eprint.iacr.org/2004/193.

[18] A. Singla, A. Mudgerikar, I. Papapanagiotou, and A. A. Yavuz, "Haa: Hardware-accelerated authentication for internet of things in mission critical vehicular networks," in *MILCOM 2015 - 2015 IEEE Military Communications Conference*, Oct 2015, pp. 1298–1304.

[19] D. J. Bernstein, "New stream cipher designs," M. Robshaw and O. Billet, Eds. Berlin, Heidelberg: Springer-Verlag, 2008, ch. The Salsa20 Family of Stream Ciphers, pp. 84–97. [Online]. Available: http://dx.doi.org/10.1007/978-3-540-68351-3_8

[20] Y. Nir and A. Langley, "ChaCha20 and Poly1305 for IETF Protocols," RFC 7539, May 2015. [Online]. Available: https://rfc-editor.org/rfc/rfc7539.txt

[21] G. Procter, "A security analysis of the composition of chacha20 and poly1305," Cryptology ePrint Archive, Report 2014/613, 2014, http://eprint.iacr.org/2014/613.

[22] STMicroelectronics, "Using stm32f4 mcu power modes with best dynamic efficiency," Tech. Rep., May 2014.

[23] wolfSSL, Github Repository, 2017. [Online]. Available: https://github.com/wolfSSL/wolfssl/tree/master/wolfcrypt

[24] K. MacKay, "micro-ecc: Ecdh and ecdsa for 8-bit, 32-bit, and 64-bit processors," Github Repository, 2013. [Online]. Available: https://github.com/kmackay/micro-ecc

[25] J. A. Preiss, W. Honig, G. S. Sukhatme, and N. Ayanian, "Crazyswarm: A large nano-quadcopter swarm," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, May 2017, pp. 3299–3304.