Dronecrypt - An Efficient Cryptographic Framework for Small Aerial Drones

Muslum Ozgur Ozmen
Oregon State University
Corvallis, Oregon, USA
ozmenmu@oregonstate.edu

Attila A. Yavuz
University of South Florida
Tampa, FL, USA
attilaayavuz@usf.edu

***Abstract*—Aerial drones are becoming an integral part of application domains including but not limited to, military operations, 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 inefficient in terms of energy and time consumption, especially for small aerial drones with resource-limited processors. Therefore, there is a significant need for an efficient cryptographic framework that can meet the requirements of small aerial drones.**

***We propose an improved 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 optimized public key infrastructure (PKI) based framework for small aerial drones, which provides energy efficient techniques by harnessing special precomputation methods and optimized elliptic curves. (ii) We also integrate recent light-weight 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 analysis. Our experiments showed that *our improved cryptographic framework achieves up to 35$\times$ lower energy consumption than its standard counterpart*.**

I. Introduction

Aerial drones are emerging mobile cyber-physical systems with potential applications such as military operations, package delivery, reconnaissance, environmental monitoring and disaster recovery/response [1], [2], [3], [4]. Due to the significant financial and strategic value involved, aerial drones are expected to be exclusively targeted by attackers. These attacks are especially critical for military networks where some sensitive information can be extracted. For instance, in 4th December 2011, a US drone (RQ-170 Sentinel) was captured by Iranian forces. A strong theory that explains how it was captured supposes that the navigation system of the drone was attacked [4], [5]. Another incident is the keylogging attack that targeted a US drone fleet in September 2011 [4]. This attack can cause the exposure of the control station and potentially result in the full control on the drone.

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 to refrain from mentioned attacks. These services are mainly guaranteed via fundamental cryptographic techniques such as symmetric ciphers and PKI. Symmetric key cryptography is highly preferred due to its efficiency and well-analyzed security. However, public key primitives can also be highly useful for certain drone applications. For instance, public key cryptography offers scalability, public verifiability, and non-repudiation that are critical for large and distributed drone networks (e.g., military operations, search and rescue).

Energy and bandwidth constraints pose critical limitations towards the deployment of standard cryptographic techniques on small aerial drones [6]. In the following, we outline recent cryptographic techniques that are considered for aerial drones and highlight some obstacles against practical deployments.

A. Related Work

Initial approaches to secure the aerial drones include implementation of well-known protocols (e.g., RSA and AES) on FPGAs [7], [8]. However, they showed that standard techniques take a lot of computation time and consume high amount of energy on small aerial drones that are further confirmed by our experiments (see Section V).

Seo et al. [9] proposed a security framework that implemented some symmetric ciphers with white-box cryptography to mitigate the impacts of drone capture attacks [9]. However, the insecurity of this line of research has been later shown in [10]. Won et al. proposed Certificateless Signcryption based protocols for mid-size aerial drones (e.g., AR.Drone 2.0) in [11]. 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.

Precomputations and optimizations of public key cryptography algorithms were considered on power constrained nodes. These include sole implementations of certain primitives (e.g., digital signature [26]) and improvements on a few asymmetric primitives (e.g., [12]). However, these techniques do not harbor the optimized elliptic curves (e.g., FourQ [14]) with the precomputation techniques to offer a comprehensive framework.

One may observe that the following issues should be addressed towards a practical deployment of cryptographic techniques on small aerial drones (i) The existing standard primitives and cryptographic protocols are inefficient in terms of energy/time consumption for small aerial drones that operates with resource-limited microprocessors (see Figure 1). (ii) The existing cryptographic protocols for aerial drones

Work done in part while Attila A. Yavuz was at Oregon State University, Corvallis, OR.

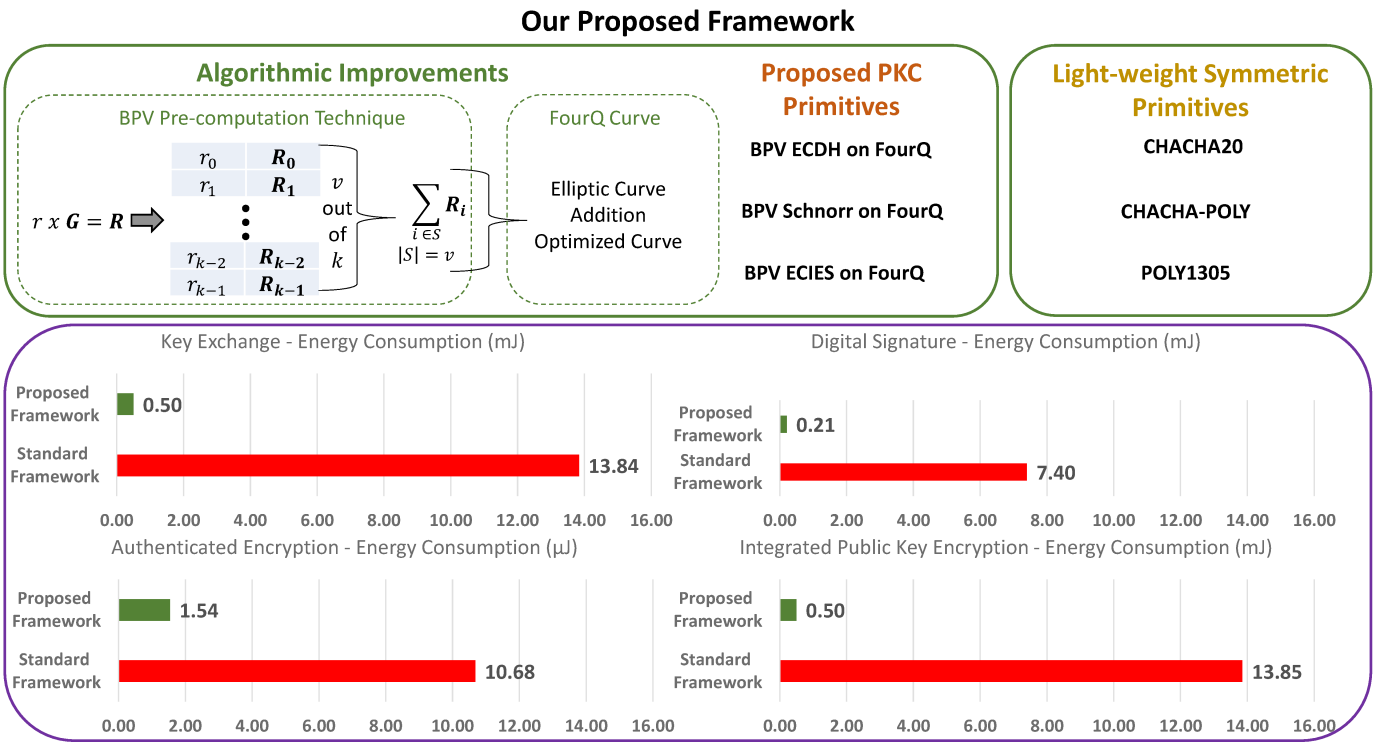

Fig. 1: The performance comparison of the proposed and standard cryptographic frameworks.

only focus on a few specific primitives but do not offer a comprehensive framework that harbors a variety of light-weight symmetric and asymmetric primitives. (iii) To the best of our knowledge, there is no open-source framework, even for some standard cryptographic primitives, which offers a *detailed energy assessment*.

B. Our Contribution

Towards the practical adoption of cryptographic techniques to resource-limited small aerial drones, we propose an improved cryptographic framework by harnessing various cryptographic primitives and optimizations, which provides significantly lower delay and energy consumption for small aerial drones, compared to the deployment of standard cryptographic techniques. We further outline our contributions as follows:

- *An Efficient Framework with Algorithmic Improvements*: Our goal is to reduce the overhead of cryptographic primitives to minimize their energy consumption for small aerial drones. Therefore, we exploit synergies among special precomputation 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) precomputation technique [13] on FourQ curve [14]. BPV technique reduces the cost of an 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 [15]), 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 recent light-weight symmetric primitives into our improved 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 and presented a detailed energy consumption analysis. As demonstrated in Figure 1, for different cryptographic primitives, our experiments showed that proposed improvements enable up to 35× less energy consumption compared to the standard techniques for small aerial drones (see Section V). 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 [7], [8], to the best of our knowledge, no comprehensive open-source cryptographic framework is available for small aerial drones. Towards meeting this need, we implemented our standard and improved cryptographic frameworks. We open-source both to enable a broad test and potential adoption at

https://github.com/ozgurozmen/Dronecrypt

Limitations: The main limitation of our proposed framework is the increased private key size, that is inherited from the BPV precomputation technique. More specifically, our improved PKI primitives require the sender to store a private key of 64 KB. However, this increased private key size translates into significant improvements on computational time and energy consumption. Moreover, only a small portion of this key (1

KB) must be loaded into RAM for computations. Considering the storage capabilities of small aerial drones, we believe this is a desirable trade-off for security-critical applications. On the other hand, if the drone is highly memory-limited and cannot tolerate such storage, standard cryptographic primitives should be preferred.

II. Preliminaries

TABLE I: 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 [13], 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 $(\mathit{Offline}, \mathit{Online})$ defined in Algorithm 1.

Algorithm 1 Boyko-Peinado-Venkatesan (BPV) Generator

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

1: Set the EC system-wide parameters $params \leftarrow (F_q, \mathbf{G}, n)$.
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: $r'_i \xleftarrow{\$} \mathbb{Z}^*_n, \quad \mathbf{R'_i} \leftarrow r'_i \times \mathbf{G}$, $i = 0, \ldots, k-1$.
4: Set pre-computation table $\Gamma = \{r'_i, \mathbf{R'_i}\}_{i=0}^{k-1}$.

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

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 [14]. 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 [16], ECDSA [17], and ECIES [18]. Secp256k1 is a NIST recommended curve [19], which is frequently used in practice. We implemented a key exchange, a digital 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 [20], AES-GCM as the authenticated encryption [21] 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).

III. Proposed Framework

A. Low Cost PKC Primitives

One may consider adopting traditional pre-computation methods to reduce the computation overhead of cryptographic techniques. However, traditional online/offline pre-computation methods require linear storage. Thus, they may not be feasible for small aerial drones equipped with light-weight microcontrollers. In addition, once the pre-computed tokens are depleted, they must be regenerated. It is shown that the regeneration cost of these pre-computation techniques may incur even more cost than following the original protocol [22].

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 an instantiation of BPV on FourQ curve, which 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 V, this strategy enables almost $35\times$ more efficient operations compared to the 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 II) keying material. With the current capabilities of low-end processors, even on the small aerial drones (such as Crazyflie 2.0 [6]), it is feasible to store such a private key. We then use BPV-FourQ to instantiate three main cryptographic schemes, which creates the following efficient digital signature, key exchange, 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 $(\Gamma, v, k, F_q, \mathbf{G}, n) \leftarrow BPV.\mathit{Offline}(1^\kappa)$.
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: $(r, \mathbf{R}) \leftarrow BPV.\mathit{Online}(\Gamma, v, k, F_q, \mathbf{G}, n)$
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 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 [16]. 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 $1.5\times$, in the expense of storing a

64KB table. Moreover, it is almost 28× more efficient than its standard counterpart (ECDH on secp256k1, see Section V).

(iii) 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 [18]. 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

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

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

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

1: $(r, \mathbf{R}) \leftarrow BPV.Online(\Gamma, v, k, F_q, \mathbf{G}, n)$
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: $d \leftarrow MAC_{k_{MAC}}(c)$

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

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: $m \leftarrow \mathcal{D}_{k_{enc}}(c)$

B. Low Cost Symmetric Key Primitives

We integrate light-weight ciphers and symmetric authentication mechanisms into our framework. Specifically, we use CHACHA20 as a very fast stream cipher [23], CHACHA-POLY as the authenticated encryption and POLY1305 as the MAC protocol [24]. These schemes provide faster and energy-efficient encryption/authentication while still offering high-security guarantees [25]. Our experiments confirmed that the adoption of these light-weight ciphers offer significant improvements in terms of computation time and energy consumption. For instance, these light-weight symmetric techniques enable up-to 7× improvement over standard symmetric ciphers (e.g., AES) for small aerial drones (see Section V).

IV. Security of the Proposed Framework

The security of our improved PKC suite relies on the security of the optimization techniques (BPV precomputation technique and FourQ curve), since the underlying schemes (Schnorr, ECDH and ECIES) are all well-studied and standardized. BPV technique takes advantage of the random walks on Cayley graphs over Abelian groups and it relies on the hardness of hidden subset sum problem. Boyko et al, show that any attack to BPV technique would require solving non-linear lattice problems [13]. BPV technique in elliptic curve based schemes (as adopted in our proposed framework) were later investigated in [26], with an integration to ECDSA. This integration relies on the affine hidden subset sum problem. Since the BPV technique is used in our schemes without any modification, it inherits its security guarantees presented in [13], [26]. As discussed in Section II, FourQ is a special elliptic curve that combines multiple optimizations to enable fast operations. While offering high computational efficiency, FourQ still preserves 128-bit security level that is the same with standardized secp256k1 curve [14].

The security of the light-weight ciphers and symmetric authentication mechanisms are studied in [25]. Their security is equivalent to their well-known standard counterparts such as AES. Moreover, we used CHACHA20, that is the 20 round CHACHA scheme to ensure its security [25]. Therefore, the security of our proposed framework is well-analyzed and comparable to the standard framework.

V. Performance Evaluation

A. Experimental Setup and Evaluation Metrics

We worked on Crazyflie 2.0 (Figure 2) due to its open-source software and hardware [6]. 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 is responsible for the communication (radio) and power management.

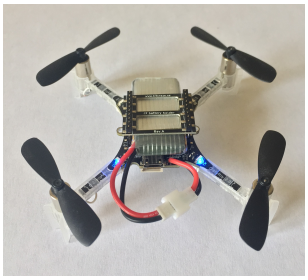

Fig. 2: Crazyflie 2.0

STM32F405 is equipped with an ARM Cortex M-4 architecture and operates at 168 MHz. It is a 32-bit microcontroller with a 192 KB SRAM and 1 MB 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 168 MHz. 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 $V = 3.3V$, $I = 40mA$ [27] 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 $40mA$, while running at 168 MHz.

We used the following libraries in our implementations. (i) All symmetric primitives are implemented using WolfCrypt [28]. (ii) We implemented all standard public key services using microECC [29] as it is a light-weight and open-source library. (iii) All optimized public key primitives are implemented with open-source Microsoft FourQ library [14]. We selected BPV parameters as $v = 16$ and $k = 1024$, which sets the size of Γ as 64 KB, that can be easily stored in STM32F405. We note that all libraries used for both standard and optimized frameworks are open-source, therefore they can be easily integrated to various aerial drones.

B. Performance Evaluation and Comparison

Experimental comparison of the standard framework and our proposed framework are presented in Tables II and III, for public key and symmetric cryptography, respectively.

TABLE II: 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| + |MAC|	13.85
ECIES-Decrypt	8817657	52.49	32	32 + |c| + |MAC|	6.93
Optimized Cryptographic Framework with Algorithmic Improvements					
BPV ECDH on FourQ	**636833**	**3.79**	65536	32	**0.50**
BPV Schnorr-Sign on FourQ	**264011**	**1.57**	65536	64	**0.21**
BPV Schnorr-Verify on FourQ	**683882**	**4.07**	32	64	**0.54**
BPV ECIES-Encrypt on FourQ	**638791**	**3.80**	65536	32 + |c| + |MAC|	**0.50**
BPV ECIES-Decrypt on FourQ	**513487**	**3.06**	32	32 + |c| + |MAC|	**0.40**

¶ Memory denotes the private key size for sign/encrypt schemes and the public key size for verify/decrypt schemes. We want to note that even though the total memory required for BPV schemes is 64 KB (that includes $t = 1024$ components), only 1 KB ($v = 16$) of it is loaded into RAM for a single operation.

TABLE III: 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 Cryptographic Framework				
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.

Although energy consumption is critical for small aerial drones, computation time also has some crucial effects on the adoption of cryptographic protocols in practice. For instance, there are time-critical applications that need frequent data transmission (e.g. camera mounted on an aerial drone for video surveillance - 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 II and III are also critical for small aerial drones.

Below, we present the comparison between the standard framework and our proposed framework, 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. Drones may be used in 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. Since digital signatures offer scalability, public verifiability, and non-repudiation properties that are lacked by symmetric key authentication mechanisms, they should be preferred for broadcast authentication. 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 offers 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 messages per second, that may not be sufficient for real-time use-cases.

• *Integrated Public Key Encryption and Key Exchange*: Although public key encryption is costly compared to symmetric key encryption, there are various use-cases that it can be useful. One of such cases could be when multiple drones need to report to a single server. We believe public key encryption can be useful for applications such as military drone fleets, tracking and search/rescue operations, due to the scalability concerns of symmetric key primitives. Moreover, ECIES provides forward security, which is useful for security-critical applications such as military operations.

A key exchange protocol is essential for aerial drone networks for the management and distribution of symmetric keys. That is, it might not be always possible to assume pre-installed symmetric keys for all drones as the controller/server may change for different purposes/use-cases. Therefore, an efficient key exchange protocol is useful to be deployed on drones.

Our optimized framework with algorithmic improvements achieves significantly lower energy consumption and faster encryption/key exchange than that of the standard framework. BPV-FourQ-ECIES is 27.70× more energy efficient than standard ECIES protocol. BPV-FourQ-ECDH also improved its standard counterpart by 27.68×. Besides the energy efficiency, our optimized protocols offer fast computation that is important 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 the 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 symmetric encryption to secure this communication channel. Moreover, when a single aerial drone reports its sensor data to one or a few base stations, symmetric key primitives can be preferred over public key schemes.

Although the energy consumption of symmetric primitives is minimal compared to PKC, it is still useful to optimize their energy consumption as their use with high message throughputs might be much higher. For example, 100 messages

per second are necessary to have a stable flight, which means 100 encryption/authentication operations per second [6]. 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 sole encryption and authentication, respectively. Therefore, our proposed framework offers significant energy improvement over its standard counterpart for symmetric key primitives.

Discussions: We would like to note that our experiments were performed on the drone processor while flight control codes were inactive. When these codes are active, it would potentially increase the run-time of both frameworks. Since the impact of such a slow-down would be similar, we believe the improvements of our framework will remain. More importantly, the message throughput that both framework can support will decrease. Considering that the standard framework is already struggling meeting with the requirements of time-critical use-cases with current configurations, it will suffer more from this decrease in throughput. Therefore, we believe that, in a real-life deployment, the benefits of our improved framework would even increase and our framework can meet the needs of stringent requirements of real-time applications.

VI. Conclusion

In this paper, to the best our knowledge, we proposed the first open-source low energy cryptographic framework tailored for small aerial drones with an in-depth energy consumption analysis. Our framework integrates algorithmic optimizations to improve the performance of standard PKC techniques, all supported with light-weight symmetric ciphers. We have implemented and deployed our optimized framework on Crazyflie 2.0, and compared its performance with the standard techniques. Our experiments confirmed that our improved framework offers up-to 35× higher speed and better energy efficiency compared with its standard counterpart. Therefore, our improved framework can meet some of the stringent energy and delay requirements of aerial drone networks with only a small impact on the battery life and end-to-end delay. We open-source our cryptographic framework for public testing, improvement and adoption purposes.

Acknowledgment. We would like to thank the anonymous reviewers for their insightful comments and suggestions. This work is supported by NSF CAREER Award CNS-1652389.

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] P. Perazzo, F. B. Sorbelli, M. Conti, G. Dini, and C. M. Pinotti, "Drone path planning for secure positioning and secure position verification," *IEEE Transactions on Mobile Computing*, vol. 16, no. 9, pp. 2478–2493, Sept 2017.

[4] K. Hartmann and C. Steup, "The vulnerability of uavs to cyber attacks - an approach to the risk assessment," in *2013 5th International Conference on Cyber Conflict (CYCON 2013)*, June 2013, pp. 1–23.

[5] R. Altawy and A. M. Youssef, "Security, privacy, and safety aspects of civilian drones: A survey," *ACM Trans. Cyber-Phys. Syst.*, vol. 1, no. 2, pp. 7:1–7:25, Nov. 2016. [Online]. Available: http://doi.acm.org/10.1145/3001836

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

[7] 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.

[8] 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

[9] 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.

[10] 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

[11] 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.

[12] M. O. Ozmen and A. A. Yavuz, "Low-cost standard public key cryptography services for wireless iot systems," in *Proceedings of the 2017 Workshop on Internet of Things Security and Privacy*, ser. IoTS&P '17. New York, NY, USA: ACM, 2017, pp. 65–70. [Online]. Available: http://doi.acm.org/10.1145/3139937.3139940

[13] 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.

[14] 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

[15] 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

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

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

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

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

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

[21] 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.

[22] 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.

[23] 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

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

[25] 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.

[26] G. Ateniese, G. Bianchi, A. T. Capossele, C. Petrioli, and D. Spenza, "Low-cost standard signatures for energy-harvesting wireless sensor networks," *ACM Trans. Embed. Comput. Syst.*, vol. 16, no. 3, pp. 64:1–64:23, apr 2017.

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

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

[29] 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