Towards Protecting Billions and Billions of Bits on the Interplanetary Internet

Stephen Herwig
William & Mary
smherwig@wm.edu

Abstract—**As multiple nations and enterprises embark on ambitious programs to explore our solar system, the success of their endeavor is intimately tied to the cooperative establishment of an efficient and secure Interplanetary Internet (IPN)—a deep space network designed for the challenges of long-distance and non-continuous communication. Unfortunately, the high latencies and low bandwidth of deep space stymie the IPN's adoption of the Internet's security protocols. In this paper, we advocate the construction of new security protocols specifically designed for the constraints of space networks and based in modern cryptographic constructs for functional encryption. We argue that such protocols could securely support a range of properties beneficial to space communication, including group messaging, in-network processing, and anonymity, and discuss the open questions and research challenges of this proposal.**

I. Introduction

With multiple nations renewing their efforts to explore our solar system, and with businesses expanding their enterprise into space, there is an increasing need for robust communication among the growing collection of deep space instruments and their operators on Earth. Unfortunately, the Internet—the chief model for robust communication on Earth—does not work in deep space: extreme distances, lack of continuous connectivity, high error rates, and asymmetric data links violate key assumptions of the Internet's protocols.

To meet these challenges, NASA, in cooperation with other national space agencies, is leading development of the protocols for the envisioned Interplanetary Internet (IPN)—a collection of interoperable deep space networks that will seamlessly connect with the terrestrial Internet. These protocols embrace an architecture for reliable communication despite long and highly-variable round-trip times, called delay-tolerant networking (DTN) [13], [18]. DTN achieves reliability using a store-and-forward communication model where each node in the path stores a message until it can forward it to the next node, thus overcoming intermittent path discontinuities.

While reliability is essential to the IPN, *security* is paramount: a diverse set of competing organizations will contribute infrastructure to the IPN, and the IPN in turn will carry traffic from mistrusting parties spanning commercial, defense, and multi-national interests. Realizing that the Internet's security protocols [29]—with their interactive, streaming, and unicast design—are likewise incompatible with a DTN setting, the DTN protocols instead specify general-purpose *extensions* [34], [6] for ensuring the confidentiality and integrity of individual messages. However, these extensions leave unspecified the critical aspects of *(1)* how they integrate with the larger networking environment (such as key management and resource naming), and *(2)* how an application would apply them to implement useful paradigms, such as multicast routing or anonymous communication. It is not even clear whether the extensions themselves are sufficient to express these paradigms.

Prior research has primarily focused on the problem of key management within the IPN, with several works [27], [35], [33] proposing the use of identity-based encryption (IBE) [32], [7], [10]—in which a node's name is its public key—to gracefully handle dynamic growth and decentralization in the network. While we believe that IBE is an important primitive for the IPN, we argue that important use cases require even greater flexibility with regard to naming, routing, and in-network functionality.

In this paper, **we propose that it is possible to modify the security extensions for the IPN to express a variety of communication models while also satisfying the latency and bandwidth constraints of space networks.** Through a series of thought experiments, we argue that by composing methods from a branch of public-key cryptography called *functional encryption* with the DTN protocols, we can achieve efficient, non-interactive communication models that embrace the IPN's need for decentralized management and bandwidth conservation.

II. Delay-Tolerant Networking Overview

The DTN architecture [13] comprises an end-to-end, message-oriented, overlay called the Bundle Protocol (BP) [31], [11] that exists above the transport layer. This overlay uses a store-carry-forward model, where each node in the path from source to destination stores a message until it can transmit it to the next node in the path. By default, BP provides an unacknowledged, prioritized (but not guaranteed) unicast message delivery service, but also includes options for reliable delivery. In total, BP improves latency and throughput in environments with frequent periods of discontinuity, while relieving the source of responsibility for end-to-end delivery.

Bundle Format. The protocol data unit of BP is the *bundle*. Each bundle contains two or more *blocks* of data: *(1)* a *primary block* that includes identification and routing information, such as the source, destination, time-to-live, and class of service,

Workshop on Security of Space and Satellite Systems (SpaceSec) 2023
27 February 2023, San Diego, CA, USA
ISBN 1-891562-89-4
https://dx.doi.org/10.14722/spacesec.2023.238733
www.ndss-symposium.org

(2) a *payload block* that carries the application data, and *(3)* optional *extension blocks*, as for carrying security metadata.

BP identifies endpoints using URI-based endpoint identifiers (EIDs), which may map to one or more DTN nodes, thereby allowing unicast, anycast, and multicast delivery semantics. The resolution of an EID to a lower-level address may occur at a node on the path other than the source, thus allowing for *late address binding*. Other than the source and destination, a bundle may also specify an EID where nodes should send error and diagnostic messages, as well as the EID that is the current *custodian* for the bundle, and thus responsible for persistently storing the data until receipt of delivery or custody-transfer.

Fragmentation and Reassembly. To allow for communication over low-volume links, BP supports bundle fragmentation and reassembly. Similar to IP, the final destination(s) are responsible for extracting the smaller blocks from incoming bundles and reassembling them into the original bundle.

Security Extensions. The BPSec [34], [6] extension to the Bundle Protocol defines two block types that provide security services for a bundle: the *Block Integrity Block* (BIB), which provides integrity protection for plaintext, and the *Block Confidentiality Block* (BCB), which provides authenticated plaintext confidentiality with additional authenticated data. A bundle may contain several security blocks, with each specifying the blocks that it protects, the cipher suites and public parameters that it uses to protect these blocks, and any *ex situ* results of the security operation, such as a signature or MAC. BPSec allows any node in the path to add, remove, or otherwise process security blocks, and specifies a set of rules for adding security blocks that ensures unambiguous processing.

Threat Model. In this proposal, we adopt BPSec's threat model, which itself reflects the Internet's threat model [28] of an on-path attacker. An on-path attacker cannot compromise the source and destination endpoints, but may control all routing nodes (or some subset) on the communication path. Specifically, an attacker my modify bundles (remove or replace blocks), inject new bundles, drop existing bundles, and subvert the network topology so as to influence routing paths. BP also exposes unique attack surfaces: the potential for long bundle lifetimes opens the possibility for an attacker to perform cryptanalysis before the bundle reaches its destination, while the need for persistent storage in the network provides a vector for resource depletion.

Deployment Status The Bundle Protocol and BPSec extensions are on a path to standardization through official standards defining organizations such as the IETF and the Consultative Committee for Space Data Systems (CCSDS) [1]—a multinational forum of the major space agencies. Operationally, BP remains in the testing phase. In 2008, the British UK-DMC (Disaster Monitoring Constellation) satellite first demonstrated the use of BP; later that year NASA also tested the protocol as part of the Deep Impact comet mission. More recently, in 2022 the Korea Pathfinder Lunar Orbiter tested the use of BP to transfer images and video from the orbiter to Earth.

III. Review of Communication Models

In this section, we motivate important communication models that are difficult to implement using the current BP and BPSec protocols. These models are not exhaustive (for instance, for brevity, we omit discussion of publish-subscribe or content-centric models) but rather exemplary, and serve as an aid for enumerating requirements for potential solutions. Before describing each model, we first identify our initial requirement:

(R1) Ensure Compatibility with BP/BPSec: Given the roughly 15-year effort to refine and standardize BP and BPSec, any additional security extensions should comply with the extension mechanisms of these protocols rather than incur the formalization and adoption of some new protocol.

A. Group Communication

Imagine that an operator wants to send the same message (say, a software update) to all rovers on Mars with confidentiality guarantees. There are three basic approaches to this problem:

1. **Group Key** The rovers share a public key and register in a multicast endpoint. The operator hybrid-encrypts a single message and sends it to the multicast group.

2. **Individual Keys** If a group key is not available, the operator instead sends to the multicast group a bundle containing a BCB for each group member; each BCB contains the symmetric key encrypted under that member's public key.

3. **Unicast Fallback** If a multicast group cannot be formed, the operator must send a separate message to each rover.

The main issue with these approaches is that the operator may not know the identities of all group members—a product of BP's assumption that a node's registration to an endpoint is a local operation. Specifically, BP does not require information about a node's registration to be available at other nodes, and does not include a mechanism for distributing information about registrations. In other words, barring a static registration system (which does not scale) or a distributed IPN name service (which will have intolerable latencies and bandwidth consumption), it is unlikely that the operator will be able to identify group members for the purposes of provisioning or selecting keys. The leads to the following requirement:

(R2) Minimize Round-trip Exchanges: Due to the high latencies of space communication, security protocols must favor non-interactive approaches, where the bundle carries as much metadata as is needed for the network to route and the endpoint to process the data. In particular, protocols should not rely on interactive negotiation, name service queries, or regular contact with a centralized trusted authority.

Beyond the group enumeration problem, the latter two approaches also have the regrettable property that bandwidth consumption scales with the size of the group. Hence, our next requirement:

(R3) Minimize Bandwidth: Due to the limited capacity of deep space links, additional security and privacy features must impose small bandwidth overheads.

B. In-Network Processing

Imagine a low-powered device on Jupiter's moon Europa that periodically sends encrypted oceanic measurements to a laboratory on Earth for analysis. Since the device is low-powered, the device first routes the message to a nearby high-powered orbiter that is then responsible for transmitting the measurements to Earth. Due to the potentially high volume of sensor data and the limited bandwidth between Jupiter and Earth, the orbiter should only transmit the "valuable" measurements and discard the others. Unfortunately, as the measurements are end-to-end encrypted to the laboratory, the orbiter has no way of determining which data is worth sending. This motivates our next requirement:

(R4) Delegate Functions to the Network: It may be infeasible (from a processing power perspective) or inefficient (from a bandwidth perspective) to process messages at the endpoints. Thus, security protocols should support the secure offloading of processing to services provided by the network itself.

C. Anonymity

Imagine that an operator is communicating with a defense spacecraft and wishes to hide the fact that she is communicating with this craft from the other deep space network nodes (which might belong to other nations) that route her message. Simply encrypting the payload block is insufficient, as the bundle's primary block contains the source and destination in plain sight for routing purposes. What the operator desires is a type of anonymity known as *unlinkable communication*: a source-destination pair is *unlinkable* if no one other than the two endpoints can identify *both* the source and destination. Under such a model, a routing node can observe that communication is taking place, and perhaps can observe one endpoint, but cannot determine that any two parties are communicating.

To achieve unlinkable communication, a DTN node may use the experimental Bundle-in-Bundle Encapsulation (BIBE) [12] specification to tunnel a BCB-encrypted bundle as the payload of an outer bundle. This procedure may be applied recursively to achieve a design similar to Tor's onion routing [17]. However, unlike Tor (which relies on a global census of relays from which the source selects a routing path), the operator may be unable to source-route a bundle due to having only a partial view of the network, or partial knowledge of the routing constraints, such as the link volumes and contact patterns between nodes. This leads to our last requirement:

(R5) Tolerate Partial Network View: Security protocols must assume that a global view of the network topology and conditions is unavailable.

IV. Approach

To address these challenges, we propose to extend BP/BPSec (**R1**) with functional encryption constructs, as these constructs are non-interactive (**R2**), have efficient implementations (**R3**), and allow for delegation of computation (**R4**) and decentralization of management (**R5**).

A. Functional Encryption

Overview. Functional encryption (FE) [8], [9] is a branch of public-key cryptography where a decryption key enables a user to learn a specific *function* of the encrypted data and nothing else. In an FE system, Alice has a public encryption key pk and a master secret key msk; given the description of some function f, Alice can use msk to generate a derived secret key sk_f associated with f. If Alice encrypts a message x with pk, and Bob decrypts this message with sk_f, Bob learns only $f(x)$ (and possibly f itself) rather than the entire message x. Functional encryption thus allows Alice to selectively share data according to an access policy expressed by f, and is a generalization of traditional public-key encryption [16], identity-based encryption (IBE) [32], and attribute-based encryption (ABE) [22], [36], [3].

In Practice. Although theoretical FE constructions exist for an arbitrary f, all practical FE schemes [2], [4], [19] restrict f to be either a linear or quadratic function, including some schemes [25] that are quantum-resistant. Several practical FE constructions also incorporate mechanisms for delegation [30] or decentralization [14], [5]. In general terms, delegation allows a user with sk_f to generate a key for a function f', where f' is more "restrictive" then f. Often this restriction takes a hierarchical form, as in a hierarchical IBE (HIBE) [20], [23], [10] scheme where a root authority generates keys for top-level domains, and each top-level domain then generates keys for their subdomains. Decentralized solutions allow f to apply to data from multiple authorities, such as ABE schemes [24], [15] where f embeds a policy over attributes from different authorities.

B. Design

Our proposed design for extensible security in the Bundle Protocol incorporates FE for two purposes: endpoint naming and in-network evaluation of bundle-specified programs.

Endpoint Naming For Bundle Protocol endpoint identifiers, we propose a URI format that composes HIBE with decentralized ABE. We envision an HIBE scheme where public keys may be expressed in a domain name syntax, such as `curiosity.mars.nasa.gov`, so as to reflect both the administrative zones within the IPN as well as the delegation of key authority. A user that sends a bundle to this EID hybrid-encrypts the payload with the public key "`curiosity.mars.nasa.gov`." Attributes are also domain names; for example, a URI of ipn://curiosity.mars.nasa.gov?camera.es indicates that the payload is encrypted with two layers: once to the identity curiosity.mars.nasa.gov, and once to the attribute camera.es. (Here we imagine communicating with some Spanish-developed camera application executing on the Curiosity rover.)

A salient feature of this scheme is wildcarding: a URI such as ipn://*.mars.nasa.gov indicates a multicast group of all child domains of mars.nasa.gov (implying that such child domains have two keys: a unique key for their fully-qualified domain name, and a shared one for the wildcard name). Wildcards may also have attributes, as in ipn://*.mars.nasa.gov?rovers, which indicates a multicast group of all child domains of mars.nasa.gov that have the rovers.mars.nasa.gov attribute (here, using a shorthand that specifies an attribute relative to the domain name). A companion syntax, ipn://@.mars.nasa.gov?rovers, indicates anycast delivery semantics: the network must deliver the bundle to one such rover.

Bundle-specified FE Programs. While our naming scheme supports *rich destination identifiers*, we also need intermediary nodes to provide *rich services* over a bundle's encrypted data. We propose the development of a BPSec extension block that specifies the following: an FE operation, the bundle block(s) that the operation targets, and a small program that invokes the FE operation and interprets the result. We imagine that such programs will target eBPF—a small RISC-like assembly language and associated bytecode extensively used in the Linux kernel—and that routing nodes will have an eBPF virtual machine to safely execute these user-defined network packet filters. Returning to the earlier example of a device on Europa transmitting a measurement x, we imagine that the device sends a bundle containing both sk_f and an eBPF program that computes $f(x)$, compares this result to some threshold value, and returns a status code indicating whether the router should forward or drop the bundle.

Anonymity Revisited. Using our proposed HIBE-ABE naming scheme, we posit an anonymity system where the sender selects a routing path among a set of EIDs, and uses Bundle-In-Bundle-Encapsulation to onion-encrypt her message to the EIDs of the path. Using our URIs, the sender need not know the domain name of each relay in the path, but can specify instead an acceptable set of relays via attributes and anycast-style wildcarding.

Of course, this path selection method may potentially result in degenerate routing choices that fail to make geographical progress towards the final destination. To ensure progress, we assume the sender has prior knowledge of the destination's approximate geographical location x for a time interval that includes the likely delivery of the message. Within each layer of the onion-encryption, the sender also includes an encryption of x and the FE operation sk_f. Upon decapsulating the bundle, a forwarder computes $f(x)$, which outputs a new geographic location of some sender-chosen proximity to x, thus allowing the forwarder to optimally choose the next hop among the set of possible next hops.

V. Open Questions

Q1: Efficient Routing Table Construction The primary routing method in DTN is Contact Graph Routing (CGR)—a system that computes routes through a time-varying topology of scheduled communication contacts using forwarding costs such as a contact's latency, volume, and willingness to serve as the custodian for a bundle. Given our proposal for semantically rich EIDs, a natural set of questions is: *How can we construct routing tables and multicast trees that efficiently manage large sets of domain names and attributes? How should routing algorithms cope with partial knowledge of an endpoint's attributes? How can we effectively route when some attributes may be non-public?*

Q2: Efficient Path Propagation Currently, operators compute and distribute CGR tables ahead of time. While this is practical for small networks, it clearly will not scale with the IPN's growth. Unfortunately, truly dynamic routing protocols that rely on distributed path advertisement may be ineffective due to the latency and bandwidth constraints of space—their advertisements might lose currency while in flight. Thus, a related question is: *How can we advertise paths securely and opportunistically, in a way that leverages the existing client communication in the network?*

Q3: Safe Execution of User-Defined FE Programs Although the focus of our proposal is protecting endpoint communication, it is also critical to protect the routing infrastructure. Our call for routers to implement an eBPF virtual machine and execute untrusted code represents a significant increase in attack surface. *How can we verify and enforce the safety of these programs while preserving their expressive power?* We imagine that recent efforts [21], [26] in applying formal methods to eBPF will be beneficial.

Q4: Revocation and Forward Secrecy A fundamental concern in any credential system is revocation. *How can we revoke identity and attribute keys, and how can clients efficiently validate whether a key has been revoked?* As revocation is a response to key compromise, a related question is *how to ensure that our hybrid-encryption schemes provide forward secrecy.* For both cases, a notion for expiration may be useful, but any scheme based on expiration needs to address the problem of devices that lack an accurate source of time.

VI. Conclusion

In this paper, we underscored the importance of security for the incipient IPN, and argued that the IPN's existing security protocol, BPSec, is either insufficient or non-optimal for a number of important use cases. Using functional encryption primitives, we sketched an approach for extending BPSec to express a variety of useful communication models, including multicast communication, in-network processing, and anonymous communication. We anticipate the biggest challenge with this proposal is that routing protocols must handle the increased complexity of routing over a semantically-rich space of endpoint identifiers.

References

[1] "CCSDS Bundle Protocol Specification," Sep. 2015, recommended Standard CCSDS 734.2-B-1. [Online]. Available: https://public.ccsds.org/Pubs/734x2b1.pdf

[2] M. Abdalla, F. Bourse, A. De Caro, and D. Pointcheval, "Simple functional encryption schemes for inner products," in *International Workshop on Public Key Cryptography (PKC)*, 2015.

[3] S. Agrawal and M. Chase, "FAME: Fast attribute-based message encryption," in *ACM Conference on Computer and Communications Security (CCS)*, 2017.

[4] S. Agrawal, B. Libert, and D. Stehlé, "Fully secure functional encryption for inner products, from standard assumptions," in *International Cryptology Conference (CRYPTO)*, 2016.

[5] M. Ambrona, D. Fiore, and C. Soriente, "Controlled functional encryption revisited: Multi-authority extensions and efficient schemes for quadratic functions," in *Privacy Enhancing Technologies Symposium (PETS)*, 2021.

[6] E. J. Birrane and K. McKeever, "Bundle Protocol Security (BPSec)," RFC 9172, Jan. 2022. [Online]. Available: https://www.rfc-editor.org/info/rfc9172

[7] D. Boneh, X. Boyen, and E.-J. Goh, "Hierarchical identity based encryption with constant size ciphertext," in *International Conference on the Theory and Applications of Cryptographic Techniques (EUROCRYPT)*, 2005.

[8] D. Boneh, A. Sahai, and B. Waters, "Functional encryption: Definitions and challenges," in *Theory of Cryptography Conference (TCC)*, 2011.

[9] ——, "Functional encryption: A new vision for public-key cryptography," *Communications of the ACM*, vol. 55, no. 11, nov 2012.

[10] X. Boyen and B. Waters, "Anonymous hierarchical identity-based encryption (without random oracles)," in *International Cryptology Conference (CRYPTO)*, 2006.

[11] S. Burleigh, K. Fall, and E. J. Birrane, "Bundle Protocol Version 7," RFC 9171, Internet Engineering Task Force, Jan. 2022. [Online]. Available: https://www.rfc-editor.org/info/rfc9171

[12] S. C. Burleigh, "Bundle-in-Bundle Encapsulation," Internet Engineering Task Force, Internet-Draft draft-ietf-dtn-bibect-03, Feb. 2020, work in Progress. [Online]. Available: https://datatracker.ietf.org/doc/draft-ietf-dtn-bibect/03/

[13] V. Cerf, S. Burleigh, A. Hooke, L. Torgerson, R. Durst, K. Scott, K. Fall, and H. Weiss, "Delay-tolerant networking architecture," RFC 4838, Internet Engineering Task Force, Apr. 2007. [Online]. Available: https://www.ietf.org/rfc/rfc4838.txt

[14] J. Chotard, E. Dufour-Sans, R. Gay, D. H. Phan, and D. Pointcheval, "Dynamic decentralized functional encryption," in *International Cryptology Conference (CRYPTO)*, 2020.

[15] P. Datta, I. Komargodski, and B. Waters, "Decentralized multi-authority ABE for DNFs from LWE," in *International Conference on the Theory and Applications of Cryptographic Techniques (EUROCRYPT)*, 2021.

[16] W. Diffie and M. Hellman, "New directions in cryptography," *IEEE Transactions on Information Theory*, vol. 22, no. 6, 1976.

[17] R. Dingledine, N. Mathewson, and P. Syverson, "Tor: The Second-Generation Onion Router," in *USENIX Security Symposium*, 2004.

[18] K. Fall, "A delay-tolerant network architecture for challenged internets," in *ACM SIGCOMM*, 2003.

[19] R. Gay, "A new paradigm for public-key functional encryption for degree-2 polynomials," in *International Workshop on Public Key Cryptography (PKC)*, 2020.

[20] C. Gentry and A. Silverberg, "Hierarchical id-based cryptography," in *International Conference on the Theory and Application of Cryptology and Information Security (ASIACRYPT)*, 2002.

[21] E. Gershuni, N. Amit, A. Gurfinkel, N. Narodytska, J. A. Navas, N. Rinetzky, L. Ryzhyk, and M. Sagiv, "Simple and precise static analysis of untrusted linux kernel extensions," in *ACM SIGPLAN's Conference on Programming Language Design and Implementation (PLDI)*, 2019.

[22] V. Goyal, O. Pandey, A. Sahai, and B. Waters, "Attribute-based encryption for fine-grained access control of encrypted data," in *ACM Conference on Computer and Communications Security (CCS)*, 2006.

[23] J. Horwitz and B. Lynn, "Toward hierarchical identity-based encryption," in *International Conference on the Theory and Applications of Cryptographic Techniques (EUROCRYPT)*, 2002.

[24] A. Lewko and B. Waters, "Decentralizing attribute-based encryption," in *International Conference on the Theory and Applications of Cryptographic Techniques (EUROCRYPT)*, 2011.

[25] J. M. B. Mera, A. Karmakar, T. Marc, and A. Soleimanian, "Efficient lattice-based inner-product functional encryption," in *International Workshop on Public Key Cryptography (PKC)*, 2022.

[26] L. Nelson, J. V. Geffen, E. Torlak, and X. Wang, "Specification and verification in the field: Applying formal methods to BPF just-in-time compilers in the Linux kernel," in *Symposium on Operating Systems Design and Implementation (OSDI)*, 2020.

[27] R. Patra, S. Surana, and S. Nedevschi, "Hierarchical identity based cryptography for end-to-end security in DTNs," in *IEEE International Conference on Intelligent Computer Communication and Processing (ICCP)*, 2008.

[28] E. Rescorla and B. Korver, "Guidelines for Writing RFC Text on Security Considerations," RFC 3552, Internet Engineering Task Force, Jul. 2003. [Online]. Available: https://www.rfc-editor.org/rfc/rfc3552

[29] E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.3," RFC 8446, Aug. 2018. [Online]. Available: https://www.rfc-editor.org/info/rfc8446

[30] A. Sahai, H. Seyalioglu, and B. Waters, "Dynamic credentials and ciphertext delegation for attribute-based encryption," in *International Cryptology Conference (CRYPTO)*, 2012.

[31] K. Scott and S. Burleigh, "Bundle protocol specification," RFC 5050, Internet Engineering Task Force, Nov. 2007. [Online]. Available: https://www.ietf.org/rfc/rfc5050.txt

[32] A. Shamir, "Identity-based cryptosystems and signature schemes," in *International Cryptology Conference (CRYPTO)*, 1984.

[33] G. Srivastava, R. Agrawal, K. Singh, R. Tripathi, and K. Naik, "A hierarchical identity-based security for delay tolerant networks using lattice-based cryptography," *Peer-to-Peer Networking and Applications*, vol. 13, 01 2020.

[34] S. Symington, S. Farrell, H. Weiss, and P. Lovell, "Bundle security protocol specification," RFC 6257, Internet Engineering Task Force, May 2011. [Online]. Available: https://www.ietf.org/rfc/rfc6257.txt

[35] W. L. Van Besien, "Dynamic, non-interactive key management for the Bundle Protocol," in *ACM Workshop on Challenged Networks (CHANTS)*, 2010.

[36] B. Waters, "Ciphertext-policy attribute-based encryption: An expressive, efficient, and provably secure realization," in *International Workshop on Public Key Cryptography (PKC)*, 2008.