An Opportunistic Batch Bundle Authentication Scheme for Energy Constrained DTNs

Haojin Zhu†, Xiaodong Lin‡, Rongxing Lu§, Xuemin (Sherman) Shen§, Dongsheng Xing† and Zhenfu Cao†
† Shanghai Jiao Tong University, Shanghai, China
{zhu-hj, dsdsdds, zfcao}@cs.sjtu.edu.cn
§University of Waterloo, Waterloo, Ontario, Canada
{rxlu, xshen}@bbcr.uwaterloo.ca
‡ University of Ontario Institute of Technology, Ontario, Canada
Xiaodong.Lin@uoit.ca

Abstract—**Bundle Authentication is a critical security service in Delay Tolerant Networks (DTNs) that ensures authenticity and integrity of bundles during multi-hop transmissions. Public key signatures, which have been suggested in existing bundle security protocol specification, achieve bundle authentication at the cost of an increased computational, transmission overhead and a higher energy consumption, which is not desirable for energy-constrained DTNs. On the other hand, the unique "store-carry-and-forward" transmission characteristic of DTNs implies that bundles from distinct/common senders can be buffered opportunistically at some common intermediate nodes. This "buffering" characteristic distinguishes DTN from any other traditional wireless networks, for which an intermediate cache is not supported. To exploit such a buffering characteristic, in this paper, we propose an Opportunistic Batch Bundle Authentication Scheme (OBBA) to achieve efficient bundle authentication. The proposed scheme adopts batch verification techniques, allowing a computational overhead to be bounded by the number of opportunistic contacts instead of the number of messages. Furthermore, we introduce a novel concept of a fragment authentication tree to minimize communication cost by choosing an optimal tree height. Finally, we implement OBBA in a specific DTN scenario setting: pocket-switched networks on campus. The simulation results in terms of computation time, transmission overhead and power consumption are given to demonstrate the efficiency and effectiveness of the proposed schemes.**

Keywords – **DTN, Bundle Security, Batch Authentication.**

I. Introduction

Delay tolerant networks (DTNs), such as sensor networks with scheduled intermittent connectivity, and pocket-switched networks that allow humans to communicate without network infrastructure [1], are highly partitioned networks that may suffer from frequent disconnectivity. In DTNs, the in-transit messages, also named bundles, can be sent over an existing link and buffered at the next hop until the next link in the path appears (e.g., a new node moves into the range or an existing one wakes up). This message propagation process is usually referred to as the "store-carry-and-forward" strategy, and the routing is decided in an "opportunistic" fashion.

Although a lot of effort has been put into the design of efficient routing algorithms for DTNs [1], [2], little attention has been paid to DTN security. Recently, there is an increasing interest in DTN routing related security issues. This interest has focused on routing misbehavior detection [3] or discouraging selfish behavior [4]; and, do not address the fundamental issue of bundle authentication or access control. In DTNs, malicious routers can arbitrarily insert false information into the bundles. If innocent routers further propagate these forged messages, attackers are able to generate large amounts of unwanted traffic in the network, which is also known as a traffic storm [5]. Since most DTN applications (e.g. pocket-switched networks) depend on resource-constrained mobile devices, extra traffic may pose a serious threat on the DTN operation [6]. Lastly, unauthorized access and utilization of DTN resources are another serious concerns of DTN security [7]. Therefore, to filter bogus bundles as early as possible, a secure yet efficient bundle authentication mechanism should be in place.

The current primary security proposal, *bundle security protocol specification* [8], proposed by delay tolerant networking research group (DTNRG), addresses two major DTN security vulnerabilities, including lack of authenticity of the bundles conveyed in messages and lack of authorization for DTN routers to appropriately access and utilize DTN resources, both of which are related to DTN bundle authentication. Bundle security protocol specification suggests to adopt Payload Integrity Block (PIB) to realize bundle authentication and router authorization by adding a digital signature to each bundle. More specifically, a bundle sender can sign the bundles with its private keys and produce a bundle-specific digital signature. This signature allows receivers as well as intermediate forwarders to confirm the authenticity of the sender, the integrity of the messages and the sender's class-of-service (CoS) rights.

However, when public key signature based bundle authentication protocol is adopted, it has faced a series of performance obstacles: high transmission, computational overhead and energy consumption. Firstly, the size of digital signatures is typically very large, in the order of tens (ECDSA) to hundreds of bytes (RSA), which will introduce extra transmission overhead. Secondly, public key signature verification is typically computationally extensive operations, and thus verifying those individual signatures one by one at each intermediate DTN router will significantly increase the computational overhead of bundle authentication. Lastly, but no less importantly, the high transmission and computational overhead also translates to high energy consumption. Although there have been significant improvements in computing and storage capability of mobile devices, advances in battery technology is still seriously lagging behind, rendering energy resource considerations the fundamental challenge in resource-constrained DTNs. This energy consumption issue becomes more challenging when multi-copy or even flooding based propagation method is employed to enhance the reliability of DTN transmission [1], [2], since the signature transmission and verification should be performed along each data delivery path. Furthermore, the bundle fragmentation issue, which means an intermediate node can split a large bundle into smaller fragments and route different fragments through different forwarding paths to make the best use of limited resources, also increases the authentication cost since each fragment requires an additional signature to make it self-authenticating [9].

On the other hand, the unique "store-carry-and-forward" transmission characteristic of DTNs implies that bundles from distinct/common senders may be buffered at some common intermediate nodes. Such a "buffering" characteristic distinguishes DTN from any other traditional wireless networks, for which intermediate cache is not supported. Our simulations show that, in a high traffic load case, there exist up to 98.25% DTN contacts during which DTN transmission is performed in a batch (two or more bundles are transferred simultaneously). To exploit such an opportunistic buffering characteristic, in this paper, we propose an Opportunistic Batch Bundle Authentication Scheme (OBBA) to reduce the bundle authentication costs. The OBBA scheme is an online/offline protocol, which allows the intermediate nodes to combine the bundles during the offline phase (or carry phase) and efficiently authenticate the combined signatures during online phase (or forwarding phase) in a batch. Similar to "Opportunistic Routing", the proposed scheme can be performed opportunistically at every intermediate node when a batch of buffered bundles need to be authenticated.

The contributions of this paper are summarized as follows.

- Firstly, we propose the basic OBBA scheme based on signature batch verification technique. With OBBA, the computational cost of bundle authentication is bounded by the number of opportunistic contacts instead of the number of bundles transferred.
- Secondly, we take advantage of fragment authentication tree (FAT) to reduce the communication overhead when fragmentation issue is considered. Since the communication overhead is determined by the FAT tree height, we discuss how to derive an optimal tree height to minimize the communication overhead.
- Thirdly, we propose an advanced OBBA scheme to achieve both of communication and computational efficiency by seamlessly integrating OBBA and FAT scheme.
- Lastly, we implement the OBBA under a specific application scenario setting: Pocket-switched networks on campus. Detailed simulation results in terms of computational time, transmission overhead and energy consumption, are given to demonstrate the efficiency and effectiveness of the proposed schemes.

To the best of our knowledge, this is the first research effort towards exploiting the unique DTN network characteristics to reduce the security overhead. The remainder of this paper is organized as follows. In Section II, some preliminaries related to DTN security and bundle authentication are reviewed. In Section III, we present the system model, adversary model, and the design goals. In Section IV, the proposed OBBA scheme is presented in details. Simulation results and performance analysis are given in Section V, followed by the conclusion in Section VI.

II. PRELIMINARIES

A. DTN Security and Bundle Authentication

It has been widely recognized that security issue is one of the major challenges for DTN deployment [7]. Due to resource-scarcity characteristic of DTNs, a general motivation for DTN security is to prevent the attackers from unauthorized accessing and utilizing of DTN resources. In the current "Bundle Security Protocol Specification" [8], it has introduced the concept of Payload Integrity Block (PIB) to enable the destination as well as any node between the source and the destination to verify the authenticity and integrity of the bundle. Currently, there is only a mandatory ciphersuite for PIB defined in the latest bundle security specification, which is based on digital signatures using RSA.

Even though public key signature based bundle authentication solutions adopted in current Bundle Security Protocol Specification has provided a general framework to secure DTNs and it also has the great advantage of providing interoperability for various standards, there are still two open issues: fragment authentication issue and performance issue.

- *Fragmentation Issue:* Fragmentation is a critical issue in DTNs. Due to limited contact duration, when a message is large, it may not be possible to send the entire message at once. One possible solution is to split the message into smaller pieces and let each become its own bundle, or "fragment bundle", and send some pieces of a large message through the current link and rest of the message through another link later to make the best use of limited resources. Due to fragmentation, traditional authentication scheme, e.g., the sender generates the signature over an entire message, may not work well since the intermediate receiver cannot authenticate any of the received fragments if it has not yet received the entire message. To address this problem, one approach called "toilet paper" was proposed in [9]. The main idea is to make each fragment self-authenticating by attaching a signature to the end of each fragment separately. However, this approach may lead to a more serious performance issue since the intermediate nodes have to put more computational and transmission efforts in transmitting and verifying a growing number of signatures.
- *Performance Issue*: Due to the resource-scarcity characteristic of DTN, how to minimize the security cost and improve the bundle authentication efficiency becomes

another critical problem for DTN security. The signature based individual bundle authentication scheme may face the challenges of expensive computational cost and transmission cost. Efficiency issue is extremely important in DTNs because the multi-copy routing/forwarding is very common in DTNs and the fragmentation issue also makes this problem more challenging.

Since the above described two issues are closely related, we aims to address these two issues together. The objective of this paper is to minimize the computational and transmission overhead by exploiting the bundle buffering characteristics.

B. Identity-based Cryptography and Pairing Technique

Even though the current bundle security specification is still based on traditional public key cryptography (PKC) such as RSA, there is an increased interest in adopting more advanced cryptographical results, such as identity-based cryptography (IBC), in DTNs [10]. The main idea of IBC is to make an entity's public key directly derivable from its publicly known identity information such as e-mail address and thus eliminate the need for public-key certificate transmission and storage. Generally speaking, PKC and IBC have their own advantages as well as weaknesses. Since the focus of this paper is bundle authentication, we skip discussions on comparison of PKC and IBC; see [10] and [8] for more details. OBBA aims to exploit the unique network characteristics of DTNs to propose a general batch bundle authentication framework to reduce bundle authentication cost. Such a framework can be based on various PKC and IBC signatures supporting batch verification. Since most of the signatures supporting batch verification are based on pairing techniques, we briefly introduce the concept of pairing as follows: Let $\mathbb{G}$ be a cyclic additive group and $\mathbb{G}_T$ be a cyclic multiplicative group of the same order q, i.e., $|\mathbb{G}| = |\mathbb{G}_T| = q$. Let P be a generator of $\mathbb{G}$. We further assume that $\hat{e} : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$ be an efficient admissible bilinear map with the following properties:

- Bilinear: for $a, b \in \mathbb{Z}_q^*$, $\hat{e}(aP, bP) = \hat{e}(P, P)^{ab}$.
- Non-degenerate: $\hat{e}(P, P) \neq 1_{\mathbb{G}_T}$.
- Computable: there is an efficient algorithm to compute $\hat{e}(P_1, Q_1)$ for any $P_1, Q_1 \in \mathbb{G}$.

III. Models and Design Goals

This section describes our system and adversary models, followed by design goals.

A. System Model

We consider a general DTN forwarding model, under which a source node can deliver packets to a destination node via one or multiple paths depending on any particular forwarding algorithm [2]. Specifically, for a given intermediate node, it may contemporarily receive bundles from multiple bundle senders via one or multiple hops. The received bundles will be buffered until the next link in the path appears. At the system initialization phase, we follow a general assumption such as [11] that an *Offline Security Manager (OSM)* exists to take charge of public key certificate issuing for traditional PKC (e.g., RSA and ECDSA) or private key generation for IBC. As shown in [8], key revocation issue is still an open problem in DTN, which is out of the scope of this paper.

B. Adversary Model

We consider that the adversary's goal is to inject bogus messages into the network, attempt to deceive other DTN nodes, gain unauthorized access to DTN resources, or exhaust constrained network resources, especially energy resources. However, we do not consider that the adversary is able to compromise DTN nodes.

C. Design Goals

Our design security goal is straightforward: all messages relayed should be authenticated so that the bogus ones inserted by any illegitimate DTN users (or external attackers) can be efficiently rejected/filtered as early as possible. We also focus on minimizing the overheads of the security design. Especially, computational cost, communication overhead as well as energy efficiency (with respect to both communication and computation) are given priority to cope with the resource-constrained nature of DTNs.

IV. The Proposed Scheme

In this section, firstly, we propose a basic OBBA scheme which aims to minimize the computational overhead by exploiting the bundle buffering opportunities. Then, we take advantage of fragment authentication tree technique (FAT) technique to further reduce the communication overhead. Lastly, we propose an advanced OBBA scheme by integrating OBBA and FAT.

A. The Basic OBBA

The main computational cost for authenticating the bundles comes from verifying a set of bundle-specific signatures issued by different bundle senders. On the other hand, the unique "Store-Carry-and-Forward" transmission strategy of DTN implies that the bundles can be verified in batch instead of one by one. To design OBBA, we firstly introduce the concept of batch verification as follows.

1) Signature Batch Verification Technique: Batch Verification is a promising technique which allows the signature verifier to quickly verify a set of digital signatures on different messages from different sources. A general batch verification technique can be described as follows: given n signatures on n distinct messages by n distinct users, using a batch verification algorithm, it is possible for a batch verifier to combine these multiple signatures into a single signature and then verify it. This single signature will convince the verifier that the n users indeed sign the n original messages[1]. BLS signature is a typical public key signature which supports batch verifying signatures from the same signer, but, in the multiple signer case, the verification cost will grow in line with the number of signers

[1] We would like to distinguish the signature batch verification from an aggregate signature in the sense that not all aggregate signature schemes support batch verification [12].

[13]. In 2007, another public key signature, CHP scheme [12], was proposed as the first short signature supporting batch verification, which provides security equivalent to 1024-bit RSA at a cost of only 160 bits and allows the total number of dominant operations (pairing) independent of the number of signatures to verify. Unfortunately, in CHP, only signatures from the same time period can be batch verified, which makes it unsuitable for DTNs due to long transmission delay and lack of global synchronization. A recent empirical study in [14] shows that some existing identity-based signatures including Chch scheme [15] and Hess [16] scheme are among the most efficient signatures supporting batch verification and also allow the batch verification cost independent of the number of signatures to verify. Without loss of generality, we take Chch as an example to show how the batch verification works, which can be summarized as follows.

1) *System Parameter:* Choose a random number $s \in Z_q^*$ as the system private key and compute $P_{pub} = sP$ as the system public key. Let $H_1 : \{0,1\}^* \times \mathbb{G} \to \mathbb{Z}_q^*$ and $H_2 : \{0,1\}^* \to \mathbb{G}$ be two hash functions.
2) *Sign:* For a particular DTN node ω_i, given the private key $sk_i = sH_2(\omega_i)$ corresponding to the public key ω_i, which is hash of node's identity, and a bundle B_j, choose a number r; compute $U_{ij} \leftarrow rH_2(\omega_i)$, $h_j \leftarrow H_1(\mathcal{B}_j, U_{ij})$ and $V_{ij} \leftarrow (r + h_j)sk_i$. $\sigma_{ij} = (U_{ij}, V_{ij})$ is the signature.
3) *IndividualVerify:* Given the node identity ω_i, bundle $\mathcal{B}_j$ and the signature σ_{ij}, compute $h_j \leftarrow H_1(\mathcal{B}_j, U_{ij})$ and accept it as a valid signature if $\hat{e}(P, V_{ij}) = \hat{e}(P_{pub}, U_{ij} + h_j H_2(\omega_i))$.
4) *SigCombine:* Given a set of nodes $\{\omega_i | 1 \leq i \leq k\}$, each of which generates signatures $\{\sigma_i^j | 1 \leq j \leq n_i\}$ on bundles $\{\mathcal{B}_i^j | 1 \leq j \leq n_i\}$, compute $V_{Batch} = \sum_{i=1}^{k} \sum_{j=1}^{n_i} V_{ij}$, $U_{Batch} = \sum_{i=1}^{k} \sum_{j=1}^{n_i} U_{ij} + h_j pk_i$. $\sigma' = (V_{Batch}, U_{Batch})$ is the combined signature.
5) *SigBatchVerify:* Given the combined signature V_{Batch} and U_{Batch}, the bundle set $\{\mathcal{B}_i^j | 1 \leq i \leq k, 1 \leq j \leq n_i\}$ on which it is based for all senders $\{\omega_i | 1 \leq i \leq k\}$, the verifier can authenticate the bundles by checking if $\hat{e}(P, V_{Batch}) = \hat{e}(P_{pub}, U_{Batch})$ holds.

Note that the signature combination can be performed incrementally and the computational costs are measured by the most expensive pairing operations. It is obvious that, given η unauthenticated signatures, in the *SigBatchVerify* phase, the computational cost is bounded by 2 pairings, which is a significant improvement over 2η pairings required by individual verification. Similar to Chch, the computational cost of Hess is also bounded by 2 pairings. It is important to point out that, though OBBA can be built on any signature supporting batch verification, different signatures with different verification efficiency or signature size make a difference to overall system performance. A detailed performance comparison between Chch and Hess is given in Section V.

2) The Design of Basic OBBA: To employ the batch signature techniques to reduce the computational overhead, one critical issue is when to authenticate the bundles. Due to the bounded verification cost, to maximize the effect of batch verification, one basic strategy is that an intermediate node tries to collect as many bundle signatures as possible before *SigBatchVerify* is performed. Basically, OBBA is an online/offline batch bundle authentication algorithm to exploit the unique "store-carry-and-forward" characteristic of DTN transmission. In this algorithm, we use a function *IsDownstreamNode*(τ) to indicate if an opportunistic contact τ is a downstream node. *IsDownstreamNode*(τ) is determined by a specific DTN routing protocol. Each node maintains two buffers, which store the authenticated messages and unauthenticated messages, respectively. As shown in Algorithm 1, during the message carrying process (offline phase), the intermediate node can perform the *SigCombine* operation to combine the unauthenticated bundle signatures incrementally. The *SigBatchVerify* operation is only triggered whenever the current node starts to transmit bundles to a downstream node, which is regarded as online phase.

Algorithm 1: The Basic OBBA

```
for unauthenticate bundles in buffer do
    Perform SigCombine to combine the signatures;
end for                                    //Offline Phase
for An Opportunistic Contact $\tau$ do
    if IsDownstreamNode($\tau$) then
        Perform the SigBatchVerify;
        Clear the unauthenticated message buffer and move
        the messages to the authenticated message buffer;
        Route the selected bundles to $\tau$;
    else
        Retrieve messages from $\tau$ and store them in the
        unauthenticated message buffer;
    end if
end for                                    //Online Phase
return valid;
```

The computational complexity of basic OBBA is analyzed as follows. Assume that n is the total number of opportunistic contacts within a specific interval, π is the average batch size of each transmission, C_B and C_I refer to the cost of performing a batch authentication and individual authentication, respectively. The computational cost of OBBA can be bounded by $O(n * C_B)$. Note that the worst case happens when each contact incurs bidirectional transmissions. In practice, the computational cost is expected to be further reduced if not evey contact incurs bidirectional transmissions. If using Chch or Hess scheme as the building block, we can obtain $C_B = C_I = 2C_{Pairing}$, where $C_{Pairing}$ refers to the computational cost for one pairing operation. In other words, the computational complexity of OBBA is bounded by the number of opportunistic contacts while irrelevant of number of message transferred. Furthermore, compared with the computational cost of individual authentication $O(\pi * n * C_I)$, OBBA is expected to achieve a better performance gain in case of a higher average batch size π. More detailed discussions on batch size distribution will be given in Section V.A.

3) Detection of Invalid Bundle Signatures: Batch authentication scheme may be vulnerable to invalid signature injection attack, which is defined as a variant of bogus message flooding attack in which a malicious DTN node may arbitrarily inject forged bundles and invalid bundle supporting signatures into the legitimate bundles. Under invalid signature injection attack, if there is even a single invalid signature in the batch, the batch verifier will be rejected with high probability. To counter this attack, recently, a new method based on recursive "divide-and-conquer" was proposed in [17] and it only requires average $O(w)$ products of pairings to identify w invalid signatures within a batch of size π.

Although there are extensive research efforts on efficiently finding invalid signature in the batch authentication, we argue that the impact of invalid signatures attack on OBBA is limited because invalid signatures could be detected and filtered within one hop. For those nodes which are not under the attack, they can still take the advantage of OBBA to reduce the authentication cost. Therefore, from a system point of view, OBBA still outperforms the traditional individual authentication.

4) Supporting Fragment Authentication: To support fragment authentication, one naive approach is that before transmission, the bundle sender proactively splits a bundle into multiple base fragments (or *proactive fragmentation*) and appends a signature to each fragment, which enables each fragment self-authenticated (toilet paper approach) [9]. This naive approach may dramatically increase the fragment signatures required and thus significantly introduce the computational and transmission overhead. Although the proposed basic OBBA can reduce the verification cost at the intermediate nodes, it cannot reduce the transmission cost incurred by the fragment authentication. For example, a specific size of bundle is split into n base fragment at the source node. Given the bundle size $50Mb$ and base fragment size $500Kb$, the bundle will be split into 100 fragment at the source node. This means that 100 fragment supporting signatures are required for fragment authentication and $100 * L_{sig}$ extra transmission overhead is introduced, where L_{sig} is the size of a supporting signature. The above analysis clearly indicates that more advanced schemes are needed to further reduce the transmission cost.

B. Utilizing Fragment Authentication Tree (FAT) to Achieve Efficient Fragment Authentication

To reduce the number of signatures required and provide fragment authentication, one promising approach is that the sender collects unsigned fragments, builds a Merkle hash tree on them [18], signs the root of the tree and thus generates one signature for all unsigned fragments, instead of one for each fragment. A Merkle tree (also called binary hash tree) is a complete binary tree equipped with a function hash and an assignment Ω, which maps a set of nodes to a set of fixed-size strings. We denote a Merkle hash tree built on the base fragments as *Fragment Authentication Tree (FAT)*. In a fragment tree, a leaf of the tree is the hash of the fragment, and the value of an internal tree node is the hash value of the concatenation of the values of its two children.

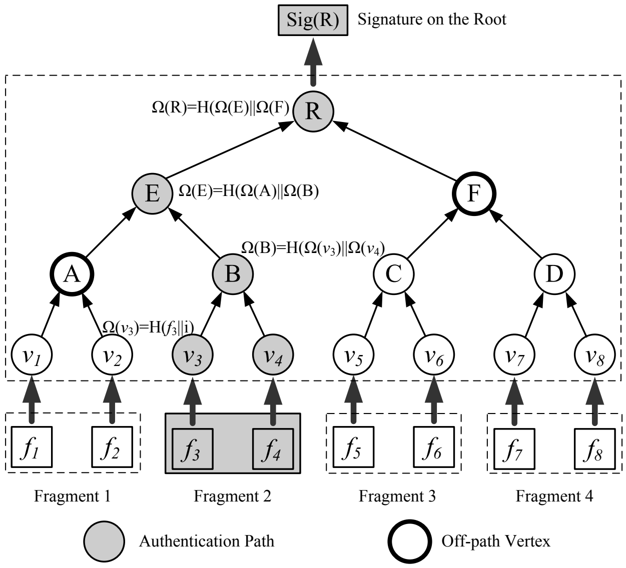


Fig. 1. An example of FAT tree and Fragment 2's off path vertices

1) FAT Building: To build a FAT, given m base fragments $\{f_i | 1 \leq i \leq m\}$ at the source, the bundle sender constructs m leaves $\{v_i = H(f_i||i) | i = 1, \ldots, m\}$ with each leaf corresponding to a base fragment. The bundle sender then builds a complete FAT with these leaves. The value of each internal tree node Ω is defined as follows:

$$\Omega(V) = H(\Omega(V_{left})||\Omega(V_{right})))$$

where V denotes an internal tree node, and V_{left} and V_{right} are V's two children. Fig. 1 shows an example to construct such a FAT with 8 fragments. After building it, the sender generates a signature on the root of tree by computing $Sig(R)$, where R denotes the hash value of the root.

If there is no reactive fragmentation during the propagation, FAT can naturally support batch authentication on fragments: each intermediate node can batch authenticate the base fragments by reconstructing the corresponding FAT and then checking the authenticity of the signature on the root.

2) Supporting Reactive Fragment Authentication: FAT can also support reactive fragment authentication at a cost of increased communication overhead. Reactive fragmentation is triggered when a connection breaks during a message transfer between two intermediate nodes. Fig. 1 also shows the reactive fragmentation: a full FAT tree comprised of m base fragments $\{f_i | i = 1, \ldots, m\}$ is split into k non-overlapping fragments $\{F_1, F_2, \ldots, F_k\}$ and each fragment F_i becomes a fragment bundle in the subsequent transmission and is forwarded along a specific path. We define the reactive fragmentation size s as $s = m/k$, which is used to describe how many base fragments included in each new fragment bundle due to reactive fragmentation. In Fig. 1, $m = 8$, $k = 4$ and $s = 2$, respectively. To authenticate such a fragment bundle, the receiver must recompute the sequence of FAT vertices between the leaf vertices and the root. For example, in Fig. 1, the receiver of fragment F_2 needs to compute the vertices v_3, v_4, B, E, and R. To perform this series of computations, each node must receive all the off-path vertices of its leaf vertex. The off-path vertices of a leaf vertex f_i are the sibling vertices of all nodes on the path from f_i to the root of the tree [19]. This means that the verifier must receive all the child vertices of B, E, and R respectively, which correspond to the set $\{A, F\}$.

The transmission of off-path vertices will increase the transmission overhead of the FAT scheme. Specifically, the following theorem states that the transmission overhead will be determined by the FAT tree height, reactive fragment probability and fragment size.

Theorem 1: *Given a FAT tree with height h, which corresponds to 2^{h-1} leave vertices, and reactive fragmentation size s (for simplicity, we assume $s \le 2^{h-2}$), the transmission cost incurred by authenticating total N fragments sent by a sender is*

$$T_1 = ((h - 1 - \lfloor log_2 s \rfloor)L_{hash} + L_{Sig})\lceil \frac{2^{h-1}}{s} \rceil \lceil \frac{N}{2^{h-1}} \rceil, \quad (1)$$

where L_{hash} and L_{sig} refer to the length of a hash value and a signature, respectively.

Proof: When $s = 1$, the number of nodes along the authentication path equals the height $h - 1$, which means the number of off-path vertices equal $h - 1$. For subsequent $s = 2, \ldots, 2^i, \ldots, 2^{h-2}$, the number of off-path vertices will decrease by one whenever the height of subtree formed by the fragment size increases by one. Therefore, we can obtain the number of off-path vertices as $h - 1 - \lfloor log_2 s \rfloor$, which contributes to the transmission overhead as $(h - 1 - \lfloor log_2 s \rfloor)L_{hash}$. So, the transmission cost for a fragment is $(h-1-\lfloor log_2 s \rfloor)L_{hash}+L_{Sig}$. Since there are $\lceil \frac{N}{2^{h-1}} \rceil$ FATs and each FAT is split into $\lceil \frac{2^{h-1}}{s} \rceil$ fragments, the total transmission cost is $((h - 1 - \lfloor log_2 s \rfloor)L_{hash} + L_{Sig})\lceil \frac{2^{h-1}}{s} \rceil \lceil \frac{N}{2^{h-1}} \rceil$. ■

From Theorem 1, it is obvious that, given the fixed fragmentation probability and the fragment size, the transmission cost will grow with the tree height. On the other hand, if there is no reactive fragmentation, the sender only needs to send a root signature to verify the whole hash tree. Therefore, the transmission cost without fragmentation is

$$T_2 = L_{Sig} * \lceil \frac{N}{2^{h-1}} \rceil, \quad (2)$$

In this case, the transmission overhead will decrease when tree height h grows. Therefore, there exists an optimal tree height to minimize the average transmission overhead, which will be discussed in the following section.

3) Finding the Optimal FAT Tree Height to Minimize Transmission Overhead: We estimate the average transmission cost for fragment authentication under a simplified fragmentation model. Let each message traverse K hops before arriving the destination and that message have the chance of p to be fragmented at any intermediate node for the fragment size s but will not be further fragmented after the first fragmentation [2]. Therefore, we can obtain the average transmission cost for a message of size N as follows:

$$T = \sum_{k=0}^{K-1} ((K - k)T_1 + kT_2)(1 - p)^k p + KT_2 * (1 - p)^K \quad (3)$$

$$= (K - A) * T_1 + A * T_2 \quad (4)$$

where $A = \frac{1-p}{p} * (1 - (1 - p)^K)$.

The following theorem gives the optimal FAT tree height h to achieve a minimal transmission overhead.

Theorem 2: *Given the reactive fragment size s, the fragmentation probability p and the average hop number K, the optimal FAT tree height h for achieving the minimal transmission overhead is*

$$h = 1 + log_2 \frac{sA \ln 2L_{Sig}}{(K - A)L_{hash}} \quad (5)$$

where $A = \frac{1-p}{p} * (1 - (1 - p)^K)$.

Proof: Since $T = (K - A) * T_1 + A * T_2$, to minimize the transmission overhead T, we have

$$\frac{dT}{dh} = (K - A)L_{hash}\lceil \frac{2^{h-1}}{s} \rceil \lceil \frac{N}{2^{h-1}} \rceil - \frac{\ln 2NAL_{Sig}}{2^{h-1}}$$

Since $\lceil \frac{2^{h-1}}{s} \rceil \lceil \frac{N}{2^{h-1}} \rceil \approx \frac{N}{s}$, it is easy to check that the derivative is 0 when $h = 1 + log_2 \frac{sA \ln 2L_{Sig}}{(K-A)L_{hash}}$. Note that, in practice, h must be an integer. ■

4) Using Learning to Approximate p, K And s in Practice: In order to optimize the communication metric, we need the global information such as the p, K and s. This can be achieved by a history learning process [21]. For example, each node records the number of fragmented bundles, fragment size and total received bundles during a specific past time duration. It also periodically updates and broadcasts its fragmentation information. The node calculates the overall approximation of the p, s and K based on its local record and the received neighboring information. So that, all nodes will have the global and accurate view about the network history. Note that this history can be limited to some time duration if the network size is large.

C. An Advanced Scheme: A Hybrid Batch Bundle Authentication Scheme (OBBA-FAT)

From previous discussions, we know that the basic OBBA can dramatically reduce the bundle authentication computational cost, and FAT can achieve the optimal transmission efficiency. Therefore, combining the OBBA and FAT into a hybrid bundle authentication scheme (OBBA-FAT) could achieve the optimal computation and transmission efficiency. Specifically, OBBA-FAT works as follows.

The source node $\mathcal{S}_i$ chooses an optimal tree height h_i according to the estimated reactive fragmentation probability,

[2] A recent study shows that the fragment size may follow a certain distribution in practice [20]. Therefore, a smaller fragment size will decrease the possibility of being re-fragmented. A more advanced fragmentation model deserves further investigation. In this study, for the simplicity of presentation, we adopt a simplified fragmentation model to introduce our approach.

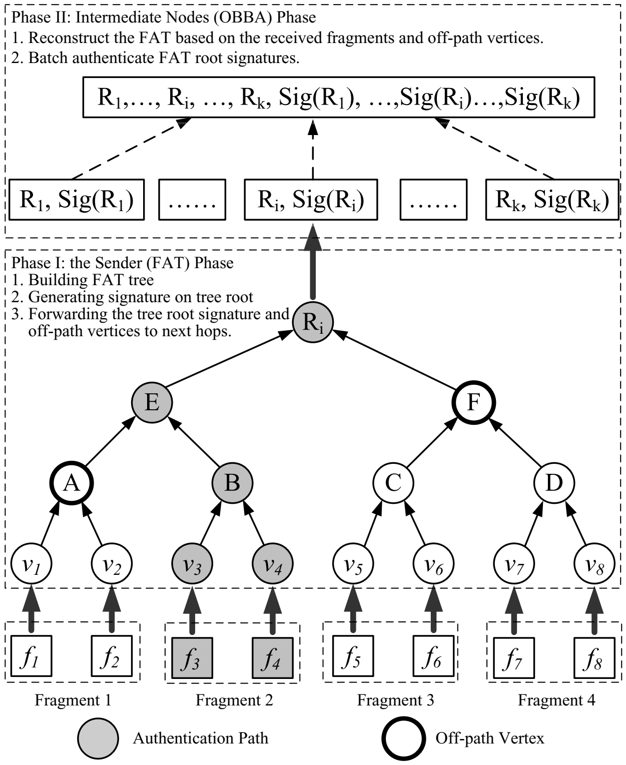

Phase II: Intermediate Nodes (OBBA) Phase
1. Reconstruct the FAT based on the received fragments and off-path vertices.
2. Batch authenticate FAT root signatures.
R1,…, Ri, …, Rk, Sig(R1), …,Sig(Ri)…,Sig(Rk)
R1, Sig(R1)
……
Ri, Sig(Ri)
……
Rk, Sig(Rk)
Phase I: the Sender (FAT) Phase
1. Building FAT tree
2. Generating signature on tree root
3. Forwarding the tree root signature and off-path vertices to next hops.
Ri
E
F
A
B
C
D
v1
v2
v3
v4
v5
v6
v7
v8
f1
f2
f3
f4
f5
f6
f7
f8
Fragment 1
Fragment 2
Fragment 3
Fragment 4
Authentication Path
Off-path Vertex

Fig. 2. An example of advanced OBBA-FAT approach

fragment size and average bundle forwarding hops. Then, $\mathcal{S}_i$ generates fragment authentication trees FAT$_j$ for every 2^{h_i-1} base fragments as introduced in Section IV-B and obtains the corresponding tree root $\{R_j, j = 1, \ldots, \lceil \frac{N}{2h_i} \rceil\}$, where $\lceil \frac{N}{2h_i} \rceil$ is the total FAT number. Finally, $\mathcal{S}_i$ generates a signature for each root as defined in Section IV-A and obtains the signatures $\sigma_j = Sign(R_j)$, where $j = 1, \ldots, \lceil \frac{N}{2h_i} \rceil$. Here, $\{R_j, \sigma_j, j = 1, \ldots, \lceil \frac{N}{2h_i} \rceil\}$ constitutes the authentication blocks for fragment authentication.

The intermediate node $\mathcal{N}$ receives fragments and authentication blocks $\{R_j, \sigma_j, j = 1, \ldots, M\}$ from multiple senders, where M denotes the total number of FATs. Upon receiving them, node $\mathcal{N}$ reconstructs the FAT trees and obtains the root values. Note that, if a fragment of the bundle is received, node $\mathcal{N}$ needs the offpath vertices to rebuild the FAT trees. If the constructed FAT roots R'_j equals to the received FAT root R_j, $\mathcal{N}$ performs *SigCombine* to combine the signatures $\{\sigma_j, j = 1, \ldots, M\}$. The above described operations are performed during the offline phase. In the online phase when an opportunistic contact appears, $\mathcal{N}$ performs the *SigBatchVerify* operation to batch authenticate the bundles. The above described advanced batch bundle authentication is illustrated in Fig. 2.

V. Simulations and Performance Evaluation

We implement the OBBA scheme on a public available DTN simulator *Opportunistic Networking Environment (ONE) Simulator* [22] and evaluate its performance under a specific application scenario: Pocket-Switched Networks on campus (PSN-Cam), which enables students as well as the faculty members to communicate with each other by using bluetooth enabled handheld devices within the campus. PSN-Cam provides a cost-effective alternative to the infrastructure based wireless networks (e.g., cellular networks or WLAN), which may suffer from limited bandwidth, high communication fee or limited coverage range. Implementing OBBA above PSN-Cam could ensure that only authorized or legitimate users can access to the PSN-Cam network while unauthorized bundles could be filtered efficiently. We run simulation with 100 mobile nodes uniformly deployed in an area of 4000 by 4000 meters. The average speed of each node varies from 1 km/h $\sim$ 1.5km/h and the transmission coverage of each node is 10 m. The maps adopted in the study are extracted from the campus maps of both of Shanghai Jiao Tong University and University of Waterloo. Each mobile node is first randomly scattered on one position of the roads and move towards another randomly selected position along the paths in the map.

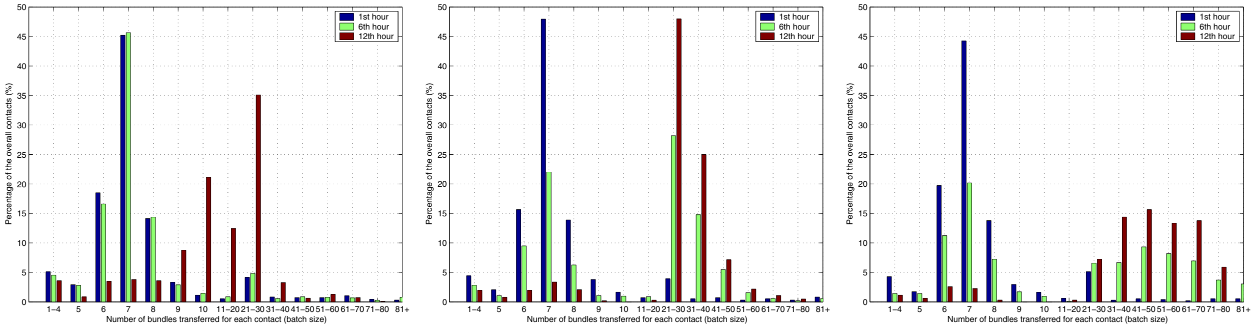
Based on the above scenario setting, we implement the OBBA on top of a typical multi-copy DTN routing protocol, Spray and Wait routing (SW) protocol [2]. As we have pointed out, any signatures supporting batch verifications can be applied in OBBA to improve the bundle authentication efficiency. In this simulation, we choose two signatures supporting batch verification: Chch scheme and Hess scheme, as the cryptographic choice of OBBA, denoted as Chch-OBBA and Hess-OBBA, respectively. We also compare them with two individual bundle authentication schemes based on ECDSA and RSA, denoted as ECDSA-IBA and RSA-IBA, respectively.

A. Bundle Size Distribution

One of the fundamental assumptions of OBBA is that the bundles are buffered and batch transmitted in DTNs. To justify our assumption, we record each opportunistic contact and its corresponding transferred message number, which have been shown in Fig. 3. We are interested in those contacts during which more than 2 bundles are transmitted since these contacts provide opportunities for batch authentication. We also denote the number of bundles contemporarily transmitted in a contact as the *batch size*.

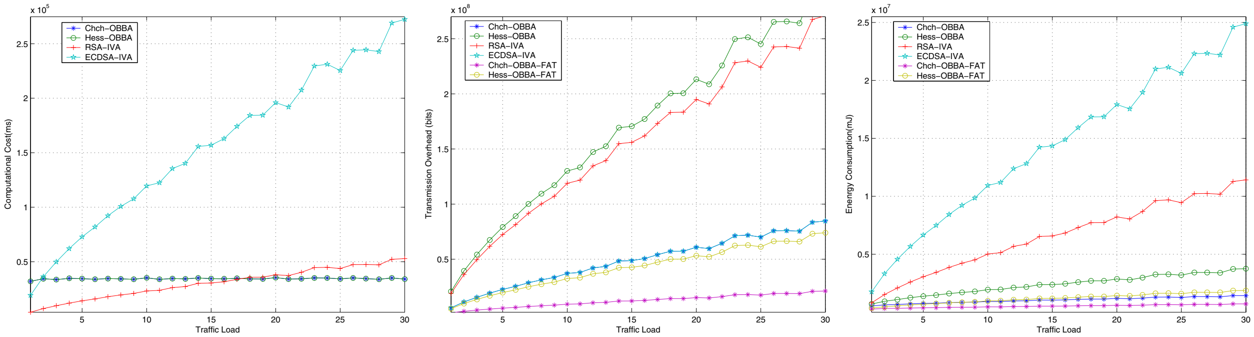
In Fig. 3-a, we show the batch size distribution under a specific network traffic setting (e.g., three messages generated for each message generation interval). It is observed that the batch transmission is dominant in the DTN transmission. For example, even in the first hour, there are more than 92.02% contacts during which more than 5 messages are transferred and the average batch size is 8. Such a percentage and average batch size will grow along with simulation time. In the 6th and 12th hour, the percentage of batch size which is larger than 5 grows to 92.68% and 95.58%, respectively, and average batch sizes increase to 10 and 12, respectively. This is because the number of buffered messages will increase with times, which increases the possibility of batch transmission.

In Fig. 3-b, we investigate two other factors which may have impact on the batch size distribution: the network traffic and the forwarding copies of DTN routing. We increase the network traffic by changing the number of bundles generated



(a) Normal Traffic (b) 3x Network Traffic & Forwarding Copies (c) 3x Network Traffic with Fragmentation

Fig. 3. Batch Size Distribution under Different Cases



(a) Computational Cost (b) Transmission Overhead (c) Energy Consumption

Fig. 4. Performance comparison of OBBA and IVA variants

per traffic generating interval from 3 to 10 and, at the same time, increasing the forwarding copy number from 6 to 12. We find that the average batch size is dramatically increased to 36 and the percentage of batch size larger than 5 is also increased to 97.24%.

Next, we evaluate the impact of fragmentation on the batch size distribution. Intuitively, as long as the sender proactively splits the bundle into multiple fragment bundles, the increased number of bundles will lead to a higher batch transmission possibility, which has been demonstrated in the simulation results. In Fig. 3-c, it is observed that the average batch size is increased to more than 80 and the percentage of batch size larger than 5 is more than 98.25%.

In summary, the above discussions justify our motivation for batch authentication. In the following section, we will study the effect of batch authentication on the overall authentication performance in terms of three metrics: computational latency, transmission overhead and energy consumption.

B. Computational Cost

In this section, we analyze the computation cost of the OBBA to further demonstrate the suitability of the proposed scheme. The computational cost for various signatures as well as the signature sizes are summarized in Table I, as shown in which, given the batch size η which corresponds to η bundle supporting signatures, only two pairing operations are needed for Chch-OBBA and Hess-OBBA.

TABLE I
SUMMARY OF VARIOUS SIGNATURES SCHEMES

Type	Scheme	Size	IndividualVerify	SigBatchVerify
PKC	RSA	1024	$\eta \times T_{\text{RSA}}$	N/A
PKC	ECDSA	320	$\eta \times T_{\text{ECDSA}}$	N/A
IBC	ChCh	320	$2\eta \times T_{\text{pair}}$	$2 \times T_{\text{pair}}$
IBC	Hess	1120	$2\eta \times T_{\text{pair}}$	$2 \times T_{\text{pair}}$

* Let η be the number of signatures to verify. In this simulation, we set $T_{\text{RSA}} = 0.1ms$, $T_{\text{ECDSA}} = 1.03ms$, and $T_{\text{Pair}} = 3.47ms$, which are evaluated on an Intel Celeron M CPU 1.73GHz machine with 2GB RAM running Ubuntu 9.04 x86_64 based on cryptographic library MIRACL [23].

Further, we evaluate the computational cost of different bundle authentication schemes under different traffic loads, which start from one bundle generated per generation interval to 30 bundles generated. The simulation results are shown in Fig. 4.a. It is observed that, when traffic load is low, Chch-OBBA and Hess-OBBA incur the same computational cost, which is a little higher than RSA-IVA and much less than ECDSA-IVA. However, computational cost of RSA-IVA grows along with increase of the traffic load and becomes higher than Chch-OBBA and Hess-OBBA after traffic load is larger than 15. On the contrary, the computational cost of Chch-OBBA and Hess-OBBA keep comparably stable. This is because computational cost of OBBA is bounded by the opportunistic contact number, which is only determined by node's traces instead of the message number. This further demonstrates the effectiveness of the OBBA.

C. Transmission Overhead

The transmission efficiency can be categorized into two cases: no reactive fragmentation case and fragmentation case.

In no reactive fragmentation case, the transmission overhead is determined by the number of supporting signatures. In Fig. 4.b, it is observed that Chch-OBBA and ECDSA-IVA have the same transmission overhead, which are much less than Hess-OBBA and RSA-IVA. Hess-OBBA has the higher communication cost due to the largest signature size. To minimize the signatures required, in OBBA-FAT, the sender can build a fragment authentication tree on the fragments and then only need to generate a signature on the FAT tree root. In the case of reactive fragmentation, according to Equation 4, the transmission overhead is determined by the hash value size L_{hash}, signature size L_{Sig}, fragment size s, fragmentation probability p, forwarding hops K and the FAT tree height h. In the simulation, we set $L_{hash} = 64$ bits, $N = 4096$, $s = 2, 3, 4$, $p = 0.1$ and $K = 5$. In Fig. 4-b, it is observed that the transmission costs of Chch-OBBA-FAT and Hess-OBBA-FAT are further reduced due to less supporting signatures required.

D. Energy consumption

Energy consumption is a major concern for DTN security design since the DTN nodes are typically battery-powered devices such as cell phones and laptop computers. The energy consumption incurred by bundle authentication includes both the computational energy cost and transmission energy cost. As for the transmission energy consumption, as reported in [24], a Chipcon CC1000 radio used in Crossbow MICA2DOT motes consumes 28.6 and 59.2 μJ to receive and transmit one byte, respectively. Therefore, we can obtain the transmission energy cost for each byte per hop as $(28.6 + 59.2)\ \mu J$.

As for the computational energy consumption, according to [25], the computation of the Tate pairing on PXA255 roughly needs the energy consumption 25.5 mJ. Due to the batch bundle authentication, the energy consumption of signature verification can be defined as $25.5 * 2$ mJ. For the ECDSA-IVA scheme, as reported in [24], it takes 45.09 mJ to verify an ECDSA-160 signature. By jointly considering the transmission and computational overhead, we obtain the simulation results on energy consumption for each bundle propagation and en-route authentication in Fig. 4-c. It is observed that the energy consumptions incurred by Chch-OBBA and Hess-OBBA are still much less that those of ECDSA-IVA and RSA-IVA. The energy cost of Chch-OBBA-FAT and Hess-OBBA-FAT are further reduced by using the introduced FAT technique. What's more important, the energy consumption of OBBA variants keep relatively stable even in the high traffic load case. This further demonstrates the effectiveness of the proposed OBBA scheme.

VI. Conclusions and Future Work

In this paper, by exploiting the unique bundle buffering characteristic, we have proposed an efficient opportunistic batch bundle authentication scheme (OBBA) for DTNs. The scheme can effectively reduce the transmission cost as well as computational cost and thus minimize the energy consumption incurred by bundle authentication. Since the computational cost is determined by the number of opportunistic contacts instead of messages transferred, OBBA is particularly suitable for energy constrained DTNs, especially under the high traffic load. The performance gain of OBBA is expected to be further improved along with the research progress of cryptographic techniques (especially for non-pairing based batch verification). OBBA can be directly applied to other DTN security solutions such as credit-based incentive schemes in DTNs [4] or encounter-ticket based secure routing scheme [3] to reduce the security overhead. Our future work includes secure routing and privacy preservation in DTNs.

References

[1] W. Gao, Q. Li, B. Zhao and G. Cao, "Multicasting in Delay Tolerant Networks: A Social Network Perspective," Proc. *ACM Mobihoc'09*, 2009.

[2] T. Spyropoulos, K. Psounis and C. S. Raghavendra, "Efficient routing in intermittently connected mobile networks: the multiple-copy cast," *IEEE/ACM Trans. on Networking*, vol. 16, no. 1, pp. 77-89, Feb. 2008.

[3] F. Li, A. Srinivasan, and J. Wu, "Thwarting Blackhole Attacks in Disruption-Tolerant Networks using Encounter Tickets," Proc. of *INFOCOM'09*, 2009.

[4] H. Zhu, X. Lin, R. Lu, Y. Fan, and X. Shen, "SMART: A Secure Multi-Layer Credit based Incentive Scheme for Delay-Tolerant Networks," *IEEE Trans. on Vehicular Technology*, vol.58, no. 8, pp. 4628-4639, 2009.

[5] N. Asokan, K. Kostianinen, P. Ginzboorg, J. Ott, and C. Luo, "Towards securing disruption-tolerant networking," Nokia Research Center, Tech. Rep. NRC-TR-2007-007.

[6] S. Farrell and V. Cahill, "Security consideartons in space and delay tolerant networks," Proc. of *SMC-IT'06*, July 2006.

[7] K. Fall and S. Farrell, "DTN: An architectural retrospective," *IEEE J. Sel. Areas Commun.*, vol. 26, no. 5, pp. 828-836, June 2008.

[8] S. Symington, S. Farrell, H. Weiss and P. Lovell, "Bundle security protocol specification," draft-irtf-dtnrg-bundle-security-08.txt, work-in-progress, March, 2009.

[9] DTNRG. Delay tolerant networking research group: dtn-interest mailing list archive, April 2005. Available from http:// mailman.dtnrg.org/pipermail/dtn-interest/2005-April/.

[10] N. Asokan, K. Kostiainen, P. Ginzboorg, J. Ott and Cheng Luo, "Applicability of identity-based cryptography for disruption-tolerant networking," Proc. of *MobiOpp'07*, June 2007.

[11] A. Kate, G. Zaverucha and Urs Hengartner, "Anonymity and security in delay tolerant networks," Proc. of *SecureComm 2007*, Sept. 2007.

[12] J.L. Camenisch, S. Hohenberger, M.Pedersen,"Batch verification of short signatures," EUROCRYPT 2007, LNCS, vol. 4515, pp.246-263, 2007.

[13] D. Boneh, B. Lynn, H. Shacham, "Short signature from the Weil Pairing," ASIACRYPT 2001, vol.2248, pp.514-532, 2001.

[14] A. L. Ferrara, M. Green, S. Huhenberger and M. Pedersen, "Practical Short Signature Batch Verification," Proc. of *CT-RSA 2009*, 2009.

[15] J. C. Cha and J. H. Cheon, "An identity-based signature from gap Diffie-Hellman groups," Proc. of *PKC'03*, vol. 2567, pp. 18-30. 2003.

[16] F.Hess, "Efficient identity-based signature schemes based on pairing,", SAC 2002, LNCS, vol. 2595, pp. 310-324, 2002.

[17] B. J. Matt, "Identification of Multiple Invalid Signatures in Pairing-Based Batched Signatures," PKC 2009, 2009.

[18] R. Merkle, "Protocols for public key cryptosystems," Proc. of *IEEE S&P*, pp. 122-133, 1980.

[19] H. Chan and A. Perrig, "Efficient security primitives derived from a secure aggregation algorithm," Proc. of *CCS'08*, 2008.

[20] M. Pitkanen, M. Keranen and J. Ott, "Message fragmentation in opportunistic DTNs," in Proc. of *WoWMoM 2008*, 2008.

[21] A. Krifa, C. Barakat, and T. Spyropoulos, "Optimal buffer management policies for delay tolerant networks," Proc. of *SECON'08*, 2008.

[22] The One Simulator: http://www.netlab.tkk.fi/tutkimus/dtn/theone/.

[23] Multiprecision Integer and Rational Arithmetic C/C++ Library (MIRACL).

[24] K. Ren, W. Lou and Y. Zhang, "Multi-user broadcast authentication in wireless sensor networks," Proc. of *SECON'07*, 2007.

[25] Y. Zhang, W. Liu,W. Lou, and Y. Fang, "Location-based compromise-tolerant security mechanisms for wireless sensor networks," *IEEE J. Sel. Areas Commun.*, vol. 24, no.2, pp.247-260, 2006.