Practical Security for Disconnected Nodes

Aaditeshwar Seth Urs Hengartner Srinivasan Keshav
School of Computer Science
University of Waterloo, Canada
{a3seth, uhengart, keshav}@uwaterloo.ca

Abstract

Endpoints in a delay tolerant network (DTN) [5] must deal with long periods of disconnection, large end-to-end communication delays, and opportunistic communication over intermittent links. This makes traditional security mechanisms inefficient and sometimes unsuitable. We study three specific problems that arise naturally in this context: initiation of a secure channel by a disconnected user using an opportunistic connection, mutual authentication over an opportunistic link, and protection of disconnected users from attacks initiated by compromised identities. We propose a security architecture for DTN based on Hierarchical Identity Based Cryptography (HIBC) that provides efficient and practical solutions to these problems.

1. Introduction

The emerging field of delay tolerant networks (DTN) [5] has recently attracted much attention. Such networks arise in inter-planetary Internets, sensor and ad-hoc networks, and other 'challenged' scenarios when connectivity is intermittent, such as in rural and underwater communication networks. In a DTN, client applications running on mobile or fixed devices opportunistically exploit connectivity over intermittent links. Mobile routers can also provide connectivity by acting as 'data mules' to carry data to and from servers that may reside in the Internet. Provision of security in such situations is a daunting task: traditional mechanisms are not well suited to environments where nodes may be disconnected for long periods of time and end-to-end communication is usually not possible. In this paper, we describe practical solutions to three problems faced by such disconnected nodes: (1) establishing a secure channel with another node (2) mutual authentication over an opportunistic link and (3) protection from users and infrastructure nodes whose credentials have been revoked or compromised. We solve these problems by extending well-known techniques in Hierarchical Identity-based Cryptography (HIBC) [3]. We start by describing a use case and explaining problems with the use of traditional security mechanisms for this use case. This is followed by an overview of the Delay Tolerant Networking architecture in Section 3, and the threat model in Section 4. Section 5, 6, and 7 describe in detail our proposals to mitigate these attacks.

2. Use case

Consider the following scenario: suppose a bus with a WiFi-based router and local storage drives past a user with a Personal Digital Assistant (PDA) with wireless capabilities. Recent studies [11] have shown that during the short time that the bus and the PDA are within range of each other, it is possible to opportunistically transfer tens of megabytes of data on the wireless link. However, to make this practical, the following three problems need to be solved. First, the PDA user must be able to establish a cryptographically-strong secure channel with some endpoint, for instance, with a mail server to download or upload mail. Second, the user and the bus must authenticate each other, so that the user is assured that the bus is not a rogue, and the bus knows how to bill for usage. Finally, if the permissions of either party to the exchange are revoked, these guarantees should continue to hold. Note that a solution for this use case is generally applicable broadly to any communication involving opportunistic links.

The state-of-the-art techniques to provide these assurances include a combination of Public Key Infrastructure (PKI) certificates issued by trusted third parties and Certificate Revocation Lists (CRLs) [13]. With PKI, a sender can establish a secure channel by encrypting data with a one-time session key, and encrypting the session key in turn with the recipient's public key. Mutual authentication is assured by means of certificates issued to the bus and the user by a mutually trusted third party. Finally, CRLs allow any entity to become aware of other entities whose private keys have been compromised.

However, not all these mechanisms work well in a disconnected environment. A disconnected sender cannot efficiently use PKI because finding out the recipient's public key requires an end-to-end round trip to a central or replicated lookup database, substantially delaying actual data transmission. Mutual authentication by means of certificates is certainly feasible using PKI certificates, but it requires authenticating parties to carry certificates from mutually trusted authorities. Finally, certificate revocation lists are unsuitable when updates can be excessively delayed and there are severe resource constraints on storage and link capacities. In our work, we propose solutions that eliminate these problems.

Our contribution is the development of practical cryptosystems for disconnected environments using HIBC [3] for creating secure channels, providing mutual authentication, and key revocation. Our solution is novel in that it explores the practical aspects related to deployment of DTN in remote rural or disconnected areas. This includes procedures for

initial key establishment and roaming among different service providers. We also describe a simple technique to prevent a user's identity from being compromised due to the loss or theft of a mobile device.

3. DTN overview

We use the Delay Tolerant Networking (DTN) [5] architecture as the basis of our design. This architecture has the following salient features:

1. Intermediate persistent storage
2. Use of opportunistic links
3. Data is sent in the form of self-identifying *bundles*

We now present some definitions relevant to our work.

1. *Region:* A region is a collection of mutually reachable DTN routers, determined by administrative policies, communication protocols, naming conventions, or connection types. The Internet is a single DTN region.
2. *Gateway:* This is a DTN router with interfaces on more than one region. An *Internet gateway* is a DTN router with at least one interface to the Internet region.
3. *Custodian:* This is a DTN router that acts as always-available proxy for intermittently connected hosts. Custodians opportunistically receive bundles from disconnected hosts, forward them to other custodians, and deliver them to a receiver whenever the receiver connects to the network.
4. *Local DTN router:* This is the DTN router that communicates directly with an endpoint. A local DTN router may or may not be a custodian as well.

Consider the following rural connectivity scenario. A PDA user in a village without any Internet connectivity offloads all her data into a village kiosk. When a bus drives past this Internet kiosk, it picks up the data locally stored at the kiosk, and also picks up data from other PDA users and kiosks on the same bus route. The bus then enters into a city and offloads all its data into an identical kiosk located at the bus station. This kiosk is connected to the Internet over a slow DSL connection, and uploads all this data into a proxy. The proxy reassembles the data and dispatches it to legacy servers like email or content servers. In this scenario, the village kiosk is a *fixed local DTN router* for PDA users, the bus is a *mobile DTN router*, the bus station kiosk is a *gateway* between the *village region* and the *Internet region*, and the proxy is the *custodian*.

We now outline some extensions to the basic DTN architecture to support user mobility; these are described in more detail in Reference [1]. We assume that mobile hosts are identified using an opaque globally unique identifier (GUID). We also assume that every DTN router has a 'default' entry that allows bundles to be forwarded (eventually) to an Internet gateway. The Internet region maintains a registry called the *Home Location Register (HLR)* that maps the mobile's GUID (*I*) to its current region (*R*). Each region maintains a *Visitor Location Register (VLR)* that stores a mapping and path from the GUIDs (*I*) of all hosts currently in the region to that host's custodian DTN router (*C*). Finally, each custodian maintains a *Local Location Register (LLR)* that maps from the GUID to the best last-hop fixed or mobile DTN router (*M*) for each mobile.

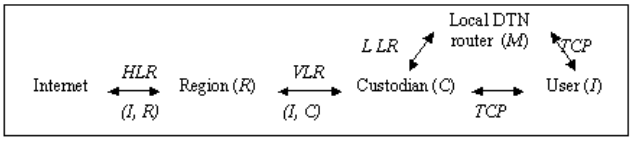

Fig. 1: Three stage hierarchy of lookups

Routing tables in a region are established using reverse-path-forwarding when a REGISTER message sent by a host propagates towards its closest Internet gateway. To send data, a host sends an 'unbound' bundle that is propagated using default paths to an Internet-accessible gateway, which locates the host's current region using the HLR and forwards the bundle to one of the region's gateways. Routing within a region from the gateway to the host uses the routing tables established using during host registration

This three-stage lookup hierarchy is shown in Fig. 1. When a mobile device moves, its location information is updated, if necessary, in the appropriate location registers using a REGISTER message.

4. Threat model

We assume the following threat model:

1. Rogue DTN routers may pretend to be valid DTN nodes.
2. DTN routers may be physically hijacked and compromised, but this is eventually detected.
3. End-systems can be hijacked or can turn malicious.
4. Eavesdroppers can potentially overhear wireless communication and break WEP-like mechanisms.

Given this threat model, consider a user who would like to conduct a secure transaction, such as a bank transaction, over a DTN. Because infrastructure nodes cannot be trusted, every opportunistic link must include a phase of mutual authentication. Second, users will want to establish end-to-end secure channels to prevent eavesdropping. Third, the infrastructure must protect itself from rogue routers. Finally, all nodes should protect themselves from nodes that were detected as being hijacked or declared malicious (the actual detection is outside the scope of this paper). This requires techniques to establish end-to-end secure channels, perform mutual authentication, and prevent communication with revoked nodes. We now describe these mechanisms.

5. Establishing secure channels

5.1. Hierarchical Identity Based Cryptography

Boneh and Franklin [4] proposed the first practical Identity Based Cryptography (IBC) scheme and many variations have subsequently been described in the literature. Unlike traditional PKI, where a user obtains the public/private key pair from a certifying authority, public keys in IBC can be any string, but private keys are obtained from a trusted authority called the Private Key Generator (PKG). Hierarchical IBC extends IBC by establishing a cooperative hierarchy of PKGs. The top-level PKG is called the root PKG, and the other PKGs are called domain PKGs, each of which inherits the first part of its public ID from its parent. A detailed description of Hierarchical Identity Based Encryption (HIDE) and Hierarchical Identity Based Signature (HIDS) is given in [3, 10], and shown in Fig. 2. In the rest of the paper, we represent the public key of a user at level t in the key hierarchy as *username@ID...*ID_{t-1}. This indicates that the parent PKG of the user is the domain PKG at level *t-1* with a public identifier of (ID_1 … ID_{t-1}).

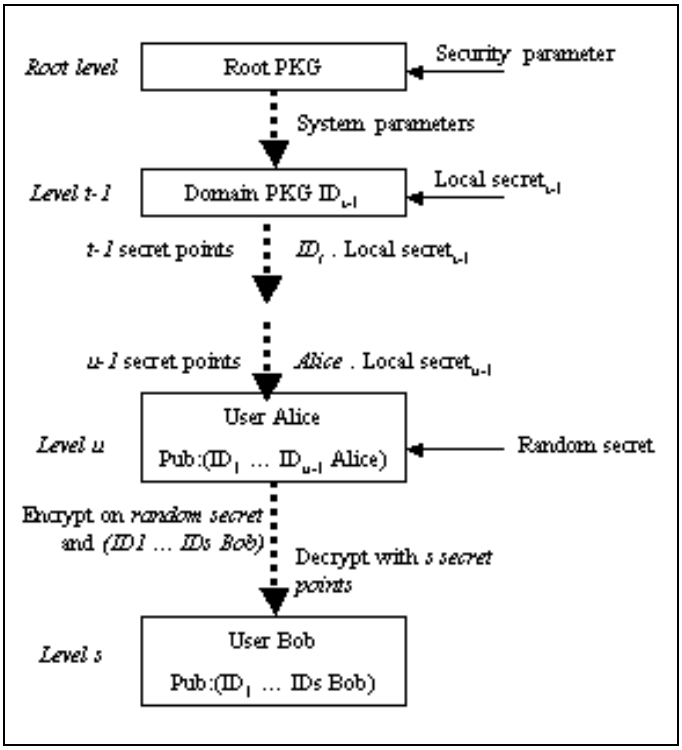

Fig. 2: Hierarchical Identity Based Encryption

Identity Based Cryptography (IBC) is ideally suited for creating a secure channel in a disconnected environment because the public key of an entity can simply be its public ID, and hence a lookup step is not required. For example, the public ID for a user can be the email address of the user itself. Another advantage is that the possession of a valid private key implies that the certification authority has certified the identity. Therefore, a valid signature serves as an assurance of authentication. Finally, a user can freely self-generate certificates signed with its private key that are universally verifiable.

It is well known that HIBC suffers from lack of forward secrecy, meaning that the compromise of a key can compromise earlier transactions as well, which were conducted using that key. A forward-secure HIBC scheme is proposed in [8]. We propose a simpler algorithm for forward-secure HIBC in Section 5.4.

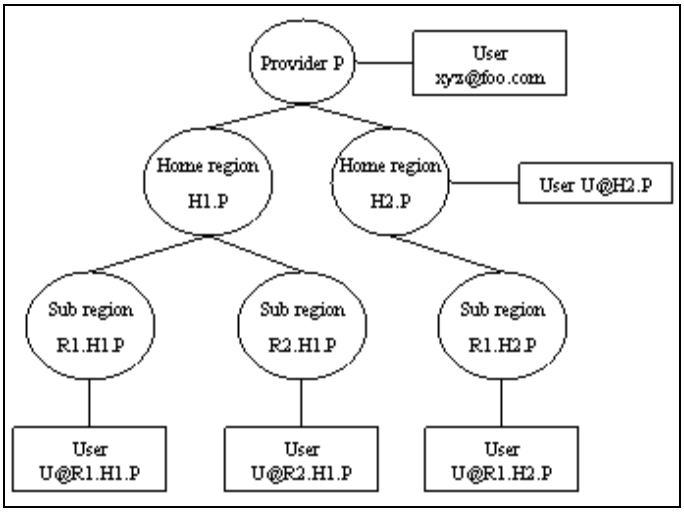

Fig. 3: Hierarchy of layout of regions

HIBC also suffers from the problem that if a PKG is compromised then it can yield all the private keys generated for lower level PKGs and users, which can be maliciously used to decrypt messages. A combination of IBC and PKI in [7] has been shown to avoid this problem, but we cannot use this scheme because it does not allow the public key to be chosen freely. Rather, the public key of a user is derived from the IBC portion of her private key. In this paper, we assume that the PKG nodes are trusted and cannot be compromised. We are investigating alternative solutions to this problem in ongoing work.

5.2. Using HIBC in DTN

We incorporate HIBC into DTN as follows: As shown in Fig. 3, an arbitrary tree-like hierarchy is imposed on the DTN regions, based on administrative structures and policies. Each provider maintains its own top-level PKG, preferably in its partition of the DTN Internet region. Every sub-region has its own domain-level PKG; alternatively, location registers in the sub-region should be able to default-route key pair requests to a parent PKG. A user can request a public ID and private key either from his nearest regional PKG (for instance, user *U@R1.H1.P* shown in Fig. 3, requests his public ID from the PKG in sub-region *R1.H1.P*), or directly from the top-level PKG. The procedure of acquiring public-private key pairs is explained below in Section 5.3, and needs to be executed only once for new users who need a DTN identity. Each DTN router also maintains a unique identity for itself.

The public ID for a DTN node in the region *R1.H1.P* can be written as *DTN-IPaddress@R1.H1.P*.

HIBC allows the creation of an end-to-end secure channel: the sender encrypts all data with the public key of the recipient, and only the recipient can decrypt the data. This provides confidentiality, integrity, and authenticated access. Besides allowing end-to-end secure channels, HIBC also protects the infrastructure from a class of attacks on the location management subsystem. Recall that a mobile host sends control messages whenever it changes its location. We use HIDS between the mobile host and the location registers being updated, with the system parameters of the mobile host's HIBC system piggybacked on the message. This ensures safety from fabrication of control messages, redirection attacks, and the creation of dead-ends by unauthorized updating of location registers. Finally, custodians in DTN send messages to the end-systems when custody has been transferred. We require custodian DTN nodes to sign these messages for the bundles that they take custody of, in order to ensure safety from spoofing. End users can store these acknowledgements for auditing. Since the HIDS scheme itself ensures non-repudiation, the audit logs can be used as proof of custody transfer.

A typical communication session in our system involves the following steps:

1. *Initial setup*: When a mobile host desires to become a member of the DTN system hosted by a certain provider, it acquires the software and the security parameters from the resellers or distribution agents of that provider. This is modeled on how cellular subscriptions are created when customers buy SIM cards from outlets of cellular companies.
2. *Link association*: When a mobile host and a mobile DTN router establish an opportunistic wireless connection, the mobile tries to connect to the router on a well-known port. Similarly, whenever a mobile host acquires a connection into a public network, it tries to connect to its DTN custodian or gateway.
3. *Mutual authentication*: Once a connection has been established between the mobile host and the DTN router, both participate in a mutual authentication procedure to ensure that malicious users and rogue DTN routers are not involved in the communication session.
4. *Location management and routing*: DTN routers carry authenticated routing and location management information from DTN custodians or gateways, which they give to mobile hosts. The mobiles then use application specific policies to select their custodians or gateways, and send appropriate control messages to the DTN routers.
5. *Data transfer and billing*: Subscription plans can be created to enable data based billing. Thus, both mobile hosts and DTN routers collect mutually authenticated statistics on the amount of data transferred between the two entities. These statistics are non-repudiable, and can be verified later if the need arises.
6. *Roaming access*: Much like roaming access provided across different cellular networks, it can be enabled across DTNs hosted by different providers. Mobile hosts and DTN routers exchange system parameters to be able to communicate with each other, along with collection of verifiable billing statistics.

We explain these steps in the subsequent sections. The *link association* procedure is explained in detail in [1].

5.3. Establishment of system parameters

IBC requires each new user to obtain a public-private key pair from the top-level or a domain-level PKG. Users can form their public ID by concatenating a desired user-name with the region name (*U@R1.H1.P*) of a domain PKG, or can request the root PKG to use any well-known ID like their email address as their public key. PKGs then push the new private keys to the users. Once the mobile host acquires the key pairs, it does not need to initiate any more interactions with the PKG, and only relies on the PKG to *push* time-based keys on a scheduled basis (see Section 7).

A new user that is directly connected to the PKG obtains its private/public key pair by communicating with the PKG over a standard secure channel mechanism like SSL. However, if the new user is in a disconnected region, it cannot communicate with the PKG. How then should it obtain its keys? We show this process in Fig. 4. We propose that USB storage devices (such as the popular '*USB keys*') be used by the PKG to distribute keys through authorized distribution agents to disconnected end users. For instance, these pre-loaded USB keys could be given to a kiosk operator who authenticates a user first-hand and then hands over a USB key (similar to the way SIM cards are handed out for cell phones today). These storage devices carry a pair of (*UID, Symmetric key*) that has been generated by the PKG. During setup phase, mobile hosts send their desired username and UID to the distribution agent. The agent signs the tuple, and sends back the signature along with the system parameters of the HIBC scheme being used. The system parameters are themselves signed by a well-known certifying authority like Verisign, to ensure the authenticity of the provider. The user verifies the signatures to confirm that the provider is real, and the agent has matched the desired username with the UID. The user then authenticates the signature, username, and UID by a MAC on the symmetric key, and addresses it to the PKG.

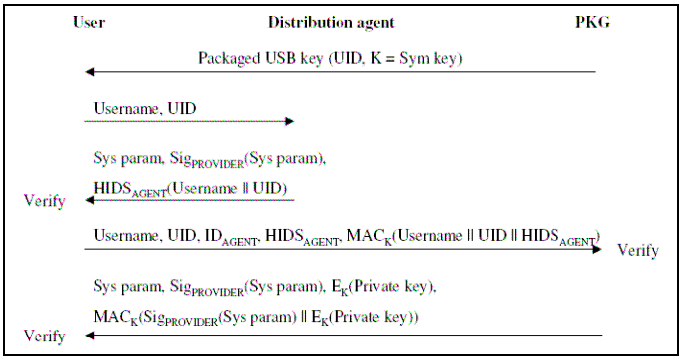

Fig. 4: Establishment of security parameters for new users

The PKG looks up this UID to verify the MAC, and uses the requested username to determine the user's public key. It then computes the public-private key pairs for the mobile host and sends them back to the user encrypted on the same symmetric key. Because the symmetric key is a one-time secret shared only by the new user and the PKG, this assures the security of this communication. To prevent the kiosk operator or distribution agents from tampering with the USB storage device, the device itself can be wrapped in a tamper-resistant package (such as a sealed cellophane wrapper), which can be verified by visual inspection.

Note that because we require an authorized agent to distribute the USB keys, we assume that if a user desires to use a well-known ID like his email address as his public ID, then the authorized agent can verify that the email address being requested by the user is indeed the user's own email address.

Note that a new user is actually unreachable because no location table entries exist for that node's UID! How can the PKG send a reply to this user? If we allowed temporary unverified entries in the location registers, we would open a security hole that could be used for a DoS attack. So, for this special case, we use source routing. Specifically, when the new user's message is sent to the PKG, it accumulates a certified route. The PKG simply reverses this route and source routes this reply. This allows the new user to be added to the network without trust violations. Once the user is added, it can REGISTER its location with a signed message.

5.4. Preventing identity theft due to loss of mobile devices

Users will generally access the DTN infrastructure through mobile devices like cell phones and PDAs. However, such devices can be easily lost, which also implies a loss of the identity. Our solution is to never keep the actual private key on the PDA, but to extend the key hierarchy by another level that is time-based. In other words, public and private keys for (ID_1 … ID_t) are extended to (ID_1 … ID_t *Date*). Note that this method of generating time-based keys is different from the key-revocation method explained in Section 7. Here, the tree is extended an additional level (ID_1 … ID_t → ID_1 … ID_t *Date*) which can be done by the user acting as a PKG for itself, but in the key-revocation method the public key is changed by concatenating a timestamp (ID_1 … ID_t → ID_1 … *Date-*ID_t), and hence this can be done only by the actual PKG. Furthermore, the granularity of change is likely to be finer for local time-based keys.

These time-based keys are generated for each new day in a secure location, such as on a desktop, and downloaded to the PDA periodically, say every few days. Thus, even if the PDA is lost, an intruder will gain access for only a limited amount of time (i.e until the keys on the PDA remain valid). Even this access can be prevented by prompt action to quarantine all resources belonging to the user till the time-based keys have expired. The advantage with time-based keys is that the duration of compromise is limited. Frequent updates can bring down the duration of exposure to arbitrarily small values. We believe this is an adequate practical solution for forward secrecy in IBC systems.

6. Mutual authentication

We now consider the case when a local DTN router meets a disconnected node. How can the two nodes mutually authenticate themselves? Two cases arise:

6.1. Challenge-Response

If the local DTN router belongs to the same provider as that of the user, a 1.5 RTT challenge-response protocol is used to verify the authenticity of the user and the local DTN router [13], shown in Fig. 5.

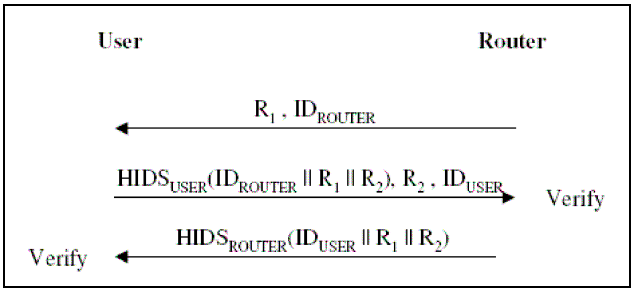

Fig. 5: Mutual authentication challenge response protocol

This protocol verifies that both the user and the DTN router are who they claim to be, and they possess valid private keys because they are able to decrypt each other's random messages. The infrastructure is assumed to belong to a Trusted Computing Base (TCB), and hence it relies on the ingress Local DTN router to authenticate the user. Per-hop authentication can also be done if there is a high risk of the ingress routers to get compromised, and to push fake traffic into the network. In addition, all DTN nodes have the *"DTN"* string as a prefix in their public ID, which can be used as an additional safeguard.

6.2. Billing

Data based billing is supported by negotiation of signed tokens between a user and the DTN router, before either party commits to sending or receiving data bundles. These tokens are sent to the DTN provider's billing server to bill the user. Incorrect billing is avoided due to the non-repudiation properties of HIBC. The detailed procedure is as follows, and illustrated in Fig. 6.

i. Authentication tokens are created and signed by users through HIDS for each bundle or group of bundles. These tokens contain the user identifier and the identifier number of the bundles. The signed tokens and the bundles are sent by the end-host to the local DTN router.

ii. The router verifies that the identifier numbers of all the bundles are included in the token, and sends a signed acknowledgement back to the user.

iii. The user stores the signed acknowledgement for auditing purposes to detect incorrect billing.

iv. The router dispatches the data to the desired destinations, and sends the signed tokens to the DTN provider's billing server in order to impose charges on the user.

v. The signed tokens prevent any tampering attacks before presenting the tokens to the billing server. To avoid replay attacks, all authentication tokens carry the sequence number specified by the user as well. The billing server has to be careful in collecting and ordering the tokens according to the sequence number before analyzing them.

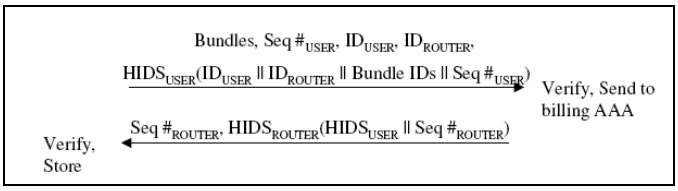

Fig. 6: Data based billing

6.3. Roaming access

The local DTN router can also belong to some other provider referred to as the *roaming provider*. This can occur if a bus drives past a PDA in a remote region, or if the PDA is taken into a remote kiosk, where the bus and the kiosk belong to a provider other than the home provider of the user. There are two cases in such a scenario:

6.3.1. If guest access is allowed

We explain this through an example illustrated in Fig. 7 that uses the notion of chains of trust. It illustrates a scenario where Bob, who is a user of provider P1, roams to access service from a kiosk that is owned by provider P2. To authenticate Bob, the kiosk uses P1's system parameters signed by P2, and Bob's public key signed by P1. Since the kiosk trusts P2, it infers that P2 has allowed access to P1's users because P2 signed the system parameters of P1. Now, since Bob is a valid P1 user, hence the kiosk grants access to Bob through the chain of trust. Similarly, Bob verifies that he can trust P2's kiosks by looking at P2's system parameters signed by P1, and the kiosk's public key signed by P2. This shows that our scheme works well despite the entities being disconnected from each other. Note that HIBC is not necessarily required for this scheme, and PKI is usable as well.

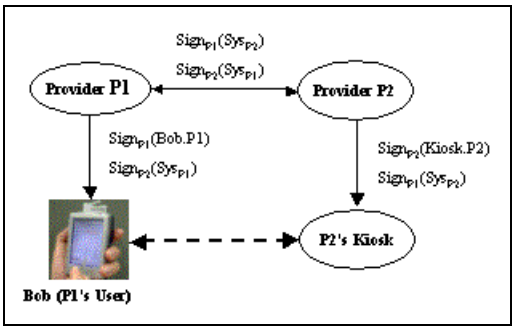

Fig. 7: Chain of trust

We slightly modify the first two steps in the 1.5 RTT challenge-response protocol shown in Fig. 5 for the single provider case. As explained in the example above, along with their respective public keys, both entities also furnish their respective system parameters signed by a well-known signing authority like Verisign, or signed by the correspondent entity's home provider. Once the system parameters have been negotiated and verified, the same protocol can be used to authenticate both parties.

Billing is done in a similar fashion by extending the scheme described in Section 6.2. Both users and DTN routers also include the identifier of their respective providers in the token signatures. Each entity signs the tokens in its own HIBC system, but verifies the tokens sent by the other entity in that entity's HIBC system. This is possible because signature verification does not require both entities to belong to the same HIBC system, but only requires knowledge of the public system parameters. This method can support 'guest access' to allow data based *post-payment* of roaming services [10].

Note that the above mechanism can be used only for sending data through a roaming provider's network. However, if data needs to be received in a roaming provider's network as well, then appropriate routing tables need to be set up in the roaming HLR, VLRs, and LLR as well. This can be achieved in two ways. (a) The user signs the control messages through HIDS, and also attaches the system parameters of their HIBC system to the messages so that anybody can verify the message authenticity. (b) The user is granted a temporary time-based identity by the local DTN router in the roaming network, similar to the roaming token method explained next. In either case, the home HLR of the user redirects all incoming data requests to the roaming HLR, much like the way a mobile IP home agent operates. Note that if control message encryption is desired as well, then only the second method can be used.

6.3.2. If roaming tokens are granted in advance

In this scheme, the soon-to-be-mobile user is given, in advance, a time-based private key and system parameters of a roaming provider. The can be done through an initial inter-federated secure communication between the user and the home provider, and the home provider and the roaming provider. The same 1.5 RTT challenge-response protocol is then used for authentication purposes between the roaming user and the local DTN router of the third-party provider. The temporary identity is used to set up routing tables in the roaming provider's network, so that the user can even receive data in the roaming network. The roaming token method is meant to support data based *pre-payment* of roaming services.

6.4. Mutual authentication of routing information

We can use HIBC to guarantee authenticity of routing information given to a mobile host by a mobile local router. Note that the information required by the user consists of the identity of the custodians that the mobile DTN router visits, the regions of these custodians, and routing and scheduling information of the mobile DTN router. We ensure that this information is correct and up-to-date by double signing tuples of *(Information, Custodian DTN node, Mobile local DTN router, Current time)*, first by the mobile DTN node, and then by the custodian DTN router (For example, the signature may look like $HIDS_{Custodian\text{-}DTN}$ *(*$HIDS_{Mobile\text{-}DTN}$ *(Information, Custodian-DTN@R1.H1.P, Mobile-DTN@R1.H1.P, Time))*). The double signing and embedding of identities of the DTN nodes within the tokens makes the scheme secure against imposter attacks even if some eavesdropper or rogue DTN router intercepts the signed tuples and tries to replay them.

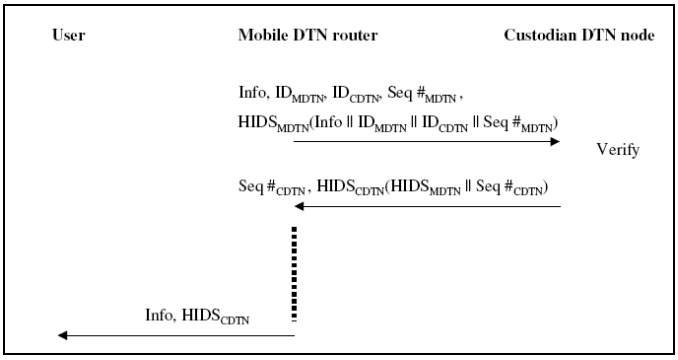

Fig. 8: Mutual authentication of routing information

7. Distributed key revocation

Instead of using CRLs, we use time-based keys [2] for revoking the rights of compromised infrastructure nodes and malicious users. We assume that clocks of all the entities are synchronized within a small margin of error. Shortly before the end of a *refresh interval*, a PKG updates every endpoint with new timestamped private keys. The refresh interval should be much larger than the allowed margin of errors in clock synchronizations. The interval can be interpreted as representing a tradeoff between the scalability of refreshes and the level of security needed, with a faster refresh implying a higher security level. The interval will also depend on the schedules of intermediate links, and can be adjusted to a maximum threshold limit that depends on the operational environment.

Our system automatically concatenates all public IDs with the last refresh time, transparent to the communicating entities, as described below. Provision is made for a small lag period before and after the time instance of change during which both old and new keys can be used. Each entity tells its correspondent node its public key and last time of change, as well as a certificate signed by the root PKG (or a well-known third party certifying authority) with the value of the refresh interval and the public ID of the entity. This allows the correspondent node to correctly decide if the time-based key of the entity has expired since the last time of change because both the time-based private key and the refresh interval are unforgeable.

A time-based key is considered expired one refresh interval after the last time of change. A user with an expired key is automatically refused access into the trusted computing base at the ingress local DTN router. Similarly, no valid user or infrastructure node accepts communication from an infrastructure node with an expired key. This means that once the infrastructure detects the compromise of any node in the system, that node can be excluded from the trusted computing base after one refresh interval.

It might seem that situations can arise when keys are not updated in time, and users may not be able to send or receive messages. However, certain reservations can be made to minimize the chances for the occurrence of such situations. (a) The problem is unlikely to arise when receiving data because most data will come in through the Internet, and the latest keys can be piggy backed on the data. (b) As part of future work, we plan to extend the TCA location management system [1] with mobility prediction. This will allow timestamped private keys to be sufficiently duplicated and flooded in the neighborhood of the user so that that whenever the user has data to send or receive, she will connect to some DTN node, and this DTN node will already possess the latest private keys for the user. (c) In certain emergency situations it should be possible for the mobile to use the symmetric key obtained during the initial setup procedure.

Note that using time-based keys imposes an additional computational and communication overhead on the entire system. However, it has the property that it is fail-safe, that is, a failure in the system does not compromise security. In contrast, CRLs are less expensive, but open a security hole in case of a failure. Developing efficient techniques to disseminate time-based keys is an interesting area for future work. Also, note that these time-based keys are likely not the same as the keys used for end-to-end encryption, authentication, and integrity. End-to-end IBC keys will usually not be time-based because a sender has no way to know a recipient's last refresh time.

8. Cross domain secure communication

We have so far assumed end-to-end communication to exist between users of the same provider. Thus, the sender encrypts data on the public key of the receiver, and the receiver decrypts the data by its private key. In IBC, the public key of an entity can simply be a public ID like its email address, and hence a lookup step is not required. This is of substantial value for disconnected operations because had PKI been used, then a round trip lookup for the public key of the receiver could have caused considerable delay. However, if the receiver belongs to some other DTN provider, then the sender will be forced to do a lookup for the system parameters of the receiver's HIBC system; the sender can then encrypt data on the public key of the receiver in the system parameters of the receiver. Note, a lookup will not be required if each DTN router can carry the system parameters of most popular DTN providers. Here, we explain the procedure in the case when the receiver's system parameters are not available readily with the local DTN router.

This is shown in Fig. 9, and explained as follows.

i. The sender encrypts data in its own HIBC system by deriving a temporary session key to be used as a public key, to reduce the possibility of replay attacks. This key can be based on a monotonically increasing sequence number, or as shown in Fig. 9, it can be uniquely based on the message signature itself. The encrypted data is sent directly to the receiver, but the receiver cannot decrypt the data until it possesses the corresponding private key. Note, the data path for the encrypted message from the sender to the receiver can be derived through hot-potato like routing algorithms that can work in a disconnected manner.

ii. The private key for decrypting the message is generated by the PKG of the sender, and sent securely to the PKG of the receiver. Presumably, both of the PKGs will be present in the Internet, and therefore SSL-like mechanisms can be used to securely exchange information.

iii. The receiver PKG then encrypts the private key and sends it to the receiver in the normal manner in which it would send any kind of control message updates to the receiver.

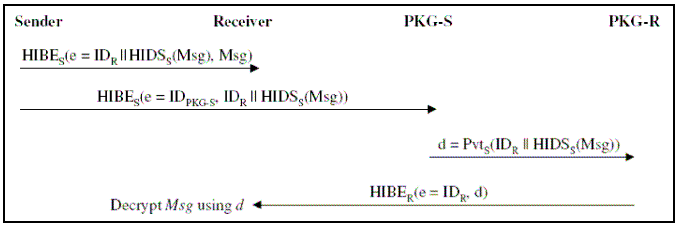

Fig. 9: Cross-domain secure communication [$\mathbf{HIBE_S(e, M)}$ stands for encrypting $\boldsymbol{M}$ on $\boldsymbol{e}$ in the HIBC system of $\boldsymbol{S}$]

9. Discussion and related work

Our work presents practical solutions to providing secure channels, mutual authentication, and rights revocation in networks with disconnected nodes. We exploit several inherent strengths of HIBC, and, assuming security of the PKGs, provide simple solutions to these challenging problems. Security in disconnected environments has only recently been studied in the literature. Related work includes:

HIBC for DTN: The use of HIBC for DTN as well as the use of time-based keys for access control was proposed in [2]. In contrast, we provide *practical* schemes for key dissemination, mutual authentication, secure location updates, audited custody transfer, and time-based forward secrecy and revocation.

Offline authorization frameworks for ubiquitous computing: The 'Lobby' system proposed in [12] is typical of secure ubiquitous computing architectures like UPnP and Jini. These architectures provide an authorization framework that support offline mobile devices. These schemes require all policy-enforcement-points (or 'Lobby's) to periodically connect to a central database and renew their user and role based ACLs. Our work supplements such architectures by providing a general platform over which fine-grained trust models can be built to provide policy-based access control at the ingress points to the DTN TCB.

10. Acknowledgements

We gratefully acknowledge Kevin Fall at Intel Research, Berkeley for his seminal suggestions on the use of Identity Based Cryptography for security in disconnected environments. We also benefited from extensive discussions with Stephen Fung at U. Waterloo.

11. References

[1] A. Seth, P. Darragh, S. Keshav. "A Generalized Architecture for Tetherless Computing in Disconnected Networks," *Work in progress: http://mindstream.watsmore.net*

[2] K. Fall, "Identity Based Cryptosystem for Secure Delay Tolerant Networking," *Manuscript.* Intel Research, Berkeley, December 2003.

[3] Craig Gentry, Alice Silverberg. "Hierarchical ID-Based Cryptography," In *Proc. International Conference on the Theory and Application of Cryptography and Information Security*, 2002.

[4] D. Boneh, M. Franklin. "Identity Based Encryption from the Weil Pairing," In *Proc. Crypto 2001 Lecture Notes in Computer Science, Vol 2139, Springer Verlag*, 2001.

[5] K. Fall. "A Delay Tolerant Network Architecture for Challenged Internets," In *Proc. SIGCOMM*, 2003.

[6] B. Lampson. "Computer Security in the Real World," *IEEE Computer*, June 2004.

[7] C. Gentry. "Certificate based encryption and the certificate revocation problem," *Proc EUROCRYPT, pp 272-293*, 2003.

[8] D. Yao, N. Fazio, Y. Dodis, A. Lysyanskasa. "ID-Based Encryption for Complex Hierarchies with Applications to Forward Security and Broadcast Encryption", In *Proc. ACM Conference on Computer and Communications Security,* 2004.

[9] J. Mirkovic, S. Dietrich, D. Dietrich, P. Reiher. "Internet Denial of Service: Attack and Defence Mechanisms", *Prentice Hall PTR,* 2005.

[10] A. Seth, S. Fung, S. Keshav. "A Secure Tetherless Computing Architecture", *Tech report, University of Waterloo, Canada.*

[11] J. Ott, D. Kutscher. “A Disconnection-Tolerant Transport for Drive-thru Internet Environments”, In *Proc. Infocom,* 2005.
[12] K. Zhang, T. Kindberg. “An Authorization Infrastructure for Nomadic Computing”, In *Proc. ACM Symposium on Access Control Models and Technologies,* 2002.
[13] A. Menezes, P. Oorschot, S. Vanstone. “Handbook of Applied Cryptography”, *CRC Press,* October 1996