Architecture for Delay-Tolerant Key Administration

IETF100 DTN Working Group
November 16, 2017

Scott Burleigh (Scott.Burleigh@jpl.nasa.gov)
David Horres (David.C.Horres@jpl.nasa.gov)
Kapali Viswanathan (kapaleeswaran.viswanathan@boeing.com)
Michael W. Benson (michael.w.benson@boeing.com)
Fred L. Templin (fred.l.templin@boeing.com)

Motivation

• On-demand & interactive communication cannot be assumed in DTN
• SSL and Online Certificate Status Protocol (OCSP) require on-demand & interactive communication
• A DTN-friendly public-key distribution and revocation protocol suite is needed

Delay-Tolerant Key Administration (DTKA)

- **Key Owner**: has private key & claims ownership of corresponding public key
- **Key Authority**: verifies ownership of public key & authenticates ownership of public key for Key Owner
- **Key User**: uses public key authentication by Key Authority to securely communicate with Key Owner

**Figure 1: Abstract Data-Flow-Diagram for DTKA**

![Diagram](https://tools.ietf.org/html/draft-burleigh-dtnwg-dtka-00)
A “Time Synchronization Mechanism” like the Network Time Protocol (NTP) allowed drift in the order of seconds. UTC offsets may be present.

System Security Configuration:
Public key of each DTKA Key Agent is securely configured into every Agent, Owner and User in the application domain.
Avoiding single-point-of-trust

Role of Key Authority is distributed among a set of Key Agents
- To prevent single-points of failure
- i.e. A pre-defined set of Key Agents constitute the Key Authority
- More on this topic later in the discussion

Figure 1: Abstract Data-Flow-Diagram for DTKA

https://tools.ietf.org/html/draft-burleigh-dtnwg-dtk-00
Node/key registration

1. (NodeID, Key) | Key
Owner +-----------------> Authority
(Node ID)

2. ListOfAuthenticated
   (Node ID, Key) | Key
   +-----------------> User

3. Secure Communications

Figure 1: Abstract Data-Flow-Diagram for DTKA

(Node ID, Key) Authentication
Out-of-band bootstrapping
In-band bootstrapping
Rely on physical security of communication channel
Rely on previously certified public keys for authentication

Periodic key updates

- Key Authority dispatches periodic updates to list of authenticated keys
- Instantaneous receipt of periodic key updates is not assumed

Communication protected using Bundle Integrity Block of DTN employing the Key Authority’s private key
Message Format for Key Updates

Figure 1: Abstract Data-Flow-Diagram for DTKA

Bulletin | Key information message (KIM): 
Hash | {([Node ID, Effective Time, Public Key], KIM | ... KIM
assert/revoke/roll-over)

Figure 3: Bulletin

Code blocks formed by each Key Authority

```
| Bulletin | Key information message (KIM): |
| Hash     | {{[Node ID, Effective Time, Public Key], KIM, ... KIM assert/revoke/roll-over}} |
```

Figure 3: Bulletin

**Bulletin formation using (Q+k) erasure coding @ Key Authority X**

```
| Bulletin | Code Block | Code Blocks |
| Hash     | Numbers    |            |
```

Figure 4: Message Format for Code Blocks

Code block assignment for Key Authorities

- **Scheme**
  - A (Q=7, k=1) erasure code is used
  - 5-out-of-8 (t-out-of-n) Key Authorities needs to be received bulletin reconstruction

Security configuration:
All DTKA entities are configured to accept only the designated code blocks from the key agents. Eg: Code block 7 is accepted only from KAs 6, 7, and 8

Table 1: Example: Code Blocks Assignments for Key Authorities

<table>
<thead>
<tr>
<th>Code Block Numbers (0 to (Q + k - 1))</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>KA 1</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>KA 2</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>KA 3</td>
<td></td>
<td>x</td>
<td>x</td>
<td>x</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>KA 4</td>
<td></td>
<td></td>
<td>x</td>
<td>x</td>
<td>x</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>KA 5</td>
<td></td>
<td>x</td>
<td>x</td>
<td>x</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>KA 6</td>
<td></td>
<td>x</td>
<td></td>
<td></td>
<td></td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>KA 7</td>
<td>x</td>
<td>x</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>KA 8</td>
<td>x</td>
<td></td>
<td>x</td>
<td>x</td>
<td></td>
<td></td>
<td></td>
<td>x</td>
</tr>
</tbody>
</table>

Summary

• Delay-Tolerant Key Administration (DTKA) is a public-key distribution and revocation mechanism for DTN
• Uses Bundle Protocol Security’s Bundle Integrity Block (BIB) to provide authentication
• Uses \((Q,k)\) erasure coding to distribute the trust in Key Authority among multiple Key Agents
  • Realizes the no-single point of failure requirement for DTKA
• Internet Draft has detailed design for the DTKA protocol suite
Thank you!
Additional slides
Node/key registration

**Figure 5: Interaction Diagram 1: Node Registration**

- **DTKA-KO [Node ID]**

- **DTKA-KA (KAx)**

- Physicaly authenticated channel (USB,...)

- TRUE = Verify (Public Key, s, [Node ID, Effective Time, Public Key,...])

Authentication

- Out-of-band bootstrapping
- Rely on physical security of communication channel

---

Node/key revocation

{m, s such that
s = Sign(PrivateKey[KAx], m) and
m = [KAx, Revoke, [Node ID, Effective Time, Pub Key]]}

TRUE = Verify(Public Key[KAx], s, [Node ID, Effective Time, New Public Key, ...])

Figure 6: Interaction Diagram 1.1: Key Revocation
Node/key rollover

Figure 7: Interaction Diagram 1.2: Key Rollover

| DTKA-KO [Node ID] | DTKA-KA (KAx) |

\{ [Node ID, Effective Time, New Public Key, s] \\
\text{such that } s = \text{Sign}(\text{Old Private Key}, [\text{Node ID}, \text{Effective Time}, \text{New Public Key}, ...]) \}

\begin{align*}
\text{TRUE} &= \text{Verify}(\text{Old Public Key, s, [Node ID,} \\
& \text{Effective Time, New Public Key, ...}])
\end{align*}

Out-of-band bootstrapping

Rely on physical security of communication channel

In-band bootstrapping

Rely on previously certified public keys for authentication

Key distribution

Figure 1: Abstract Data-Flow-Diagram for DTKA

Figure 4: Message Format for Code Blocks

(Q+k) erasure decoding with DTKA rules for accepting code blocks

Figure 8: Interaction Diagram 2: Bulk Key Distribution