



# Secure Space Network Communications

Scott Burleigh  
Jet Propulsion Laboratory  
California Institute of Technology

6 May 2015

This research was carried out at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration. (c) 2015 California Institute of Technology. Government sponsorship acknowledged.



# Threat Analysis (1 of 4)

---

- Reference: “SECURITY THREATS AGAINST SPACE MISSIONS INFORMATIONAL REPORT”, CCSDS 350.1-G-1, Green Book, October 2006.
- Active threats:
  - Data corruption
    - Possible catastrophic loss if commands are corrupted.
  - Ground facility physical attack
    - Possible mission loss.
  - Interception of data
    - Possible exposure of vulnerabilities.



# Threat Analysis (2 of 4)

---

- More active threats:
  - Jamming
    - Failure of communications, leading to errors in ground operations and/or incorrect spacecraft behavior.
  - Masquerade
    - Possible transmission of malicious commands or insertion of malicious data into spacecraft memory.
  - Replay
    - Possible operational error due to receiving multiple copies of a non-idempotent command.
  - Software threats
    - Possible introduction of operational error.



# Threat Analysis (3 of 4)

---

- More active threats:
  - Unauthorized access
    - Possible transmission of malicious commands.
- Passive threats:
  - Tapping of communication links
  - Exploitation of software vulnerabilities
  - Traffic analysis
- Threat sources:
  - Terrorists and criminals
  - Foreign intelligence services



# Threat Analysis (4 of 4)

---

- More threat sources:
  - Terrorists and criminals
  - Foreign intelligence services
  - Subversives or political activists
  - Computer hackers
  - Software and hardware failures
  - Commercial competitors
  - Dishonest maintenance or systems personnel
  - Inadvertent actions of staff members
  - Disgruntled staff members



# Link-layer Security

---

- Reference: “SPACE DATA LINK SECURITY PROTOCOL DRAFT RECOMMENDED STANDARD”, CCSDS 355.0-R-3, Red Book, October 2013.
- Authentication, encryption, or both.
  - Header prepended to data area of transfer frame carries security parameters.
  - Trailer appended to data area of transfer frame carries message authentication code.
- Works with all CCSDS link-layer protocols.



# Network-layer Security

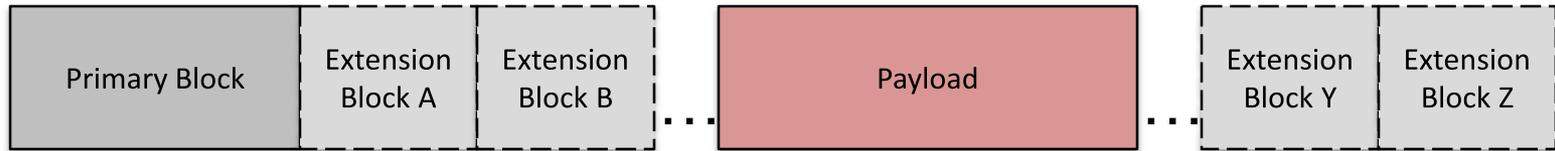
---

- Delay-Tolerant Networking (DTN) protocols developed by the Internet Research Task Force enable automatic networked communications over links where continuous immediate end-to-end data exchange may be impossible, e.g., space data links.
- Security over a DTN-based network is provided by security extension blocks inserted into DTN data “bundles”: RFC 6257, currently being updated in the new DTN Working Group of the Internet Engineering Task Force. Authentication, integrity, confidentiality.



# General Structure of a DTN “Bundle”

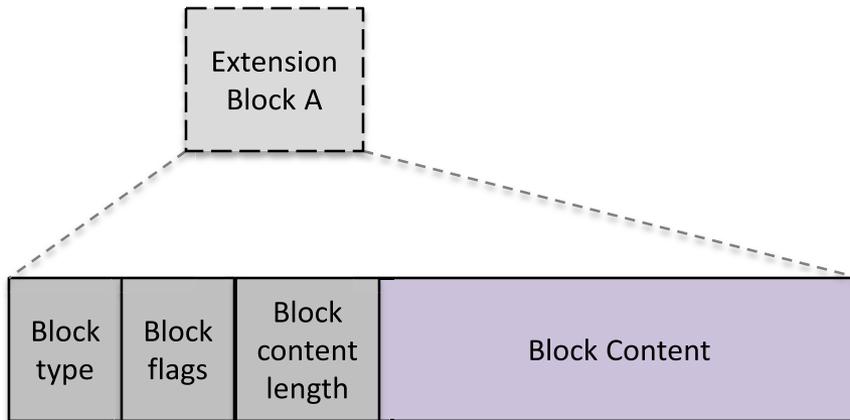
---





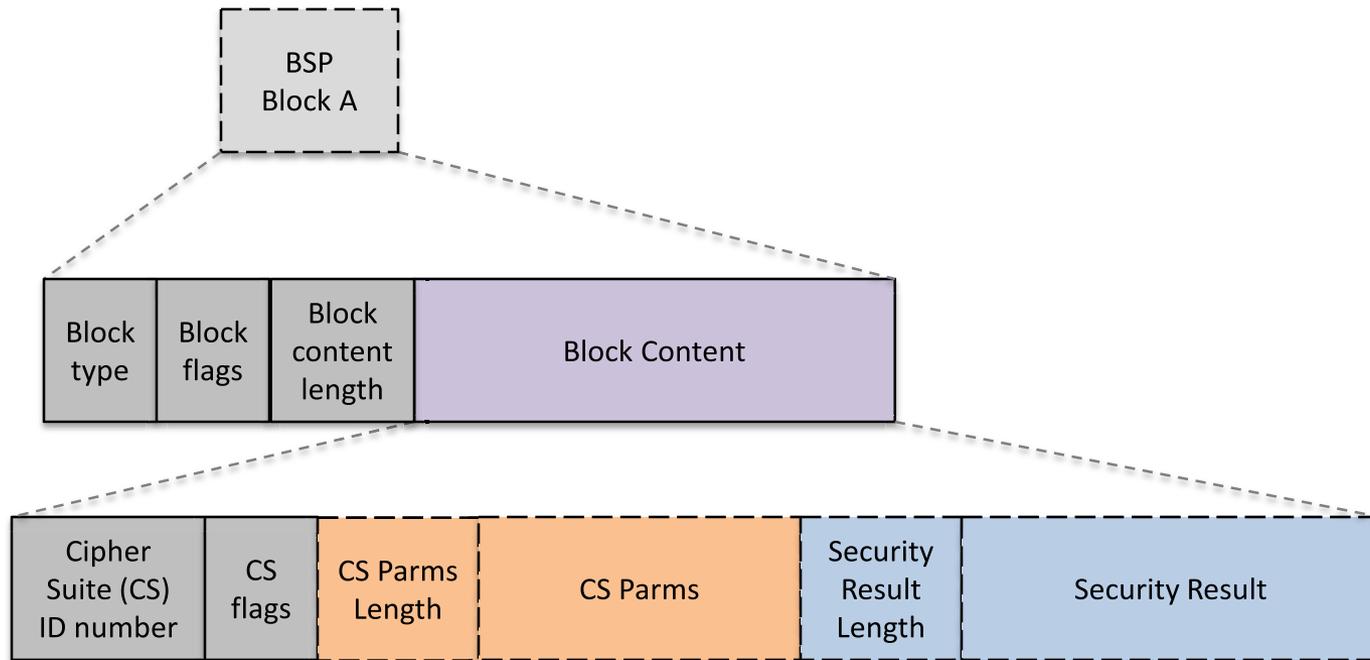
# Extension Block Structure

---



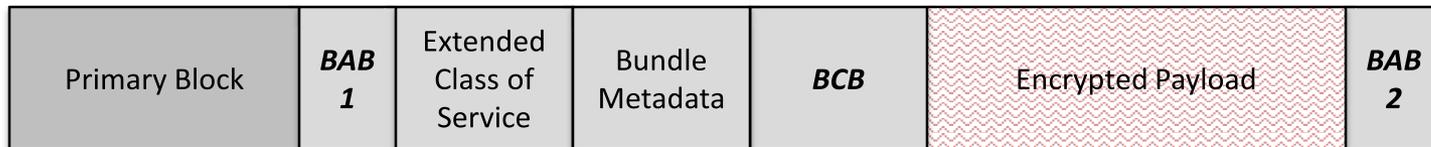
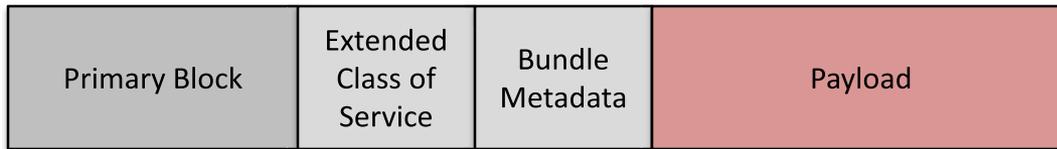


# General Security Block Structure





# BSP Example





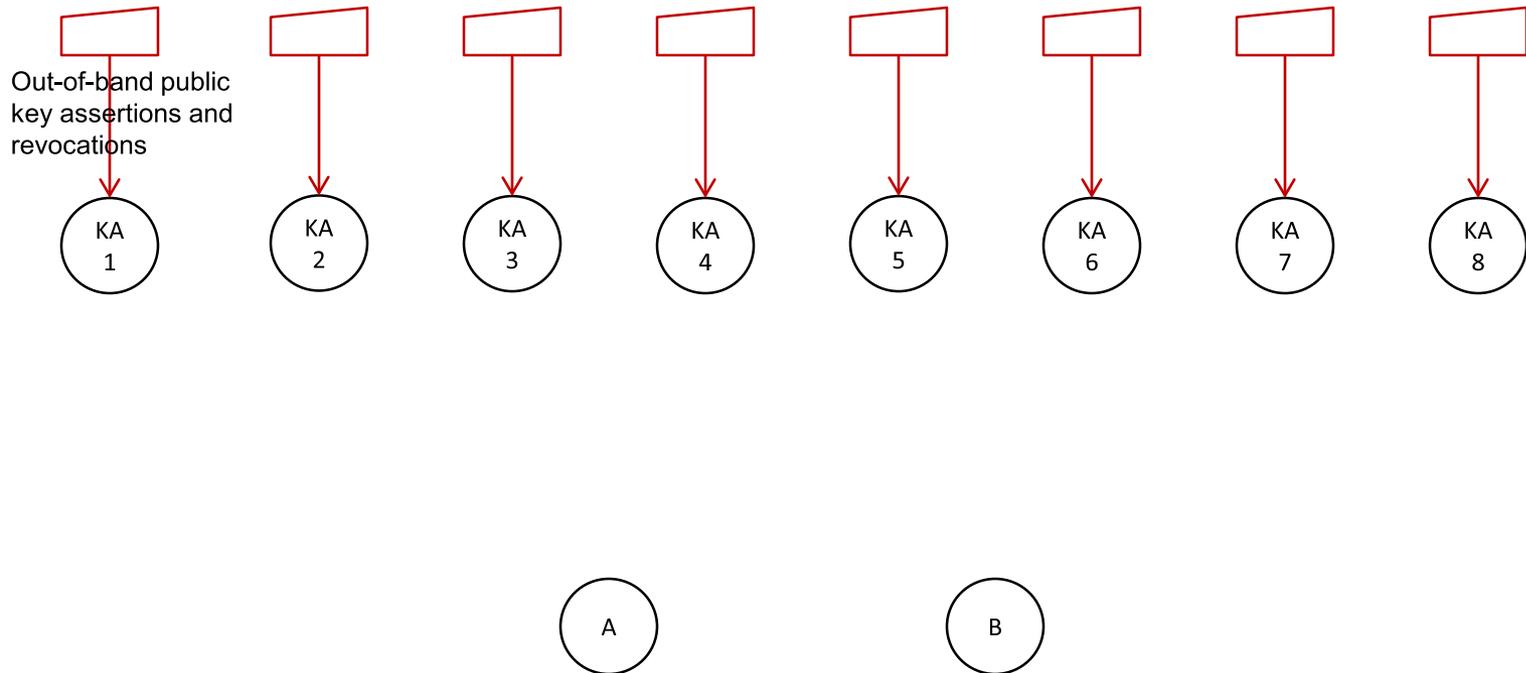
# Key Management in DTN

---

- As in other DTN mechanisms, negotiation (e.g., IKE or TLS) is undesirable: it might never complete in time for the key to be usable. Not delay-tolerant.
  - Keys should be time-qualified and “pushed” in advance of the time at which they will be needed.
- Autonomous generation and assertion of all keys by all nodes would be delay-tolerant but unreliable: in concept, any node is potentially subject to compromise. We need to control keys so that they can be revoked.
- But any key authority node could be destroyed or, worse, compromised.



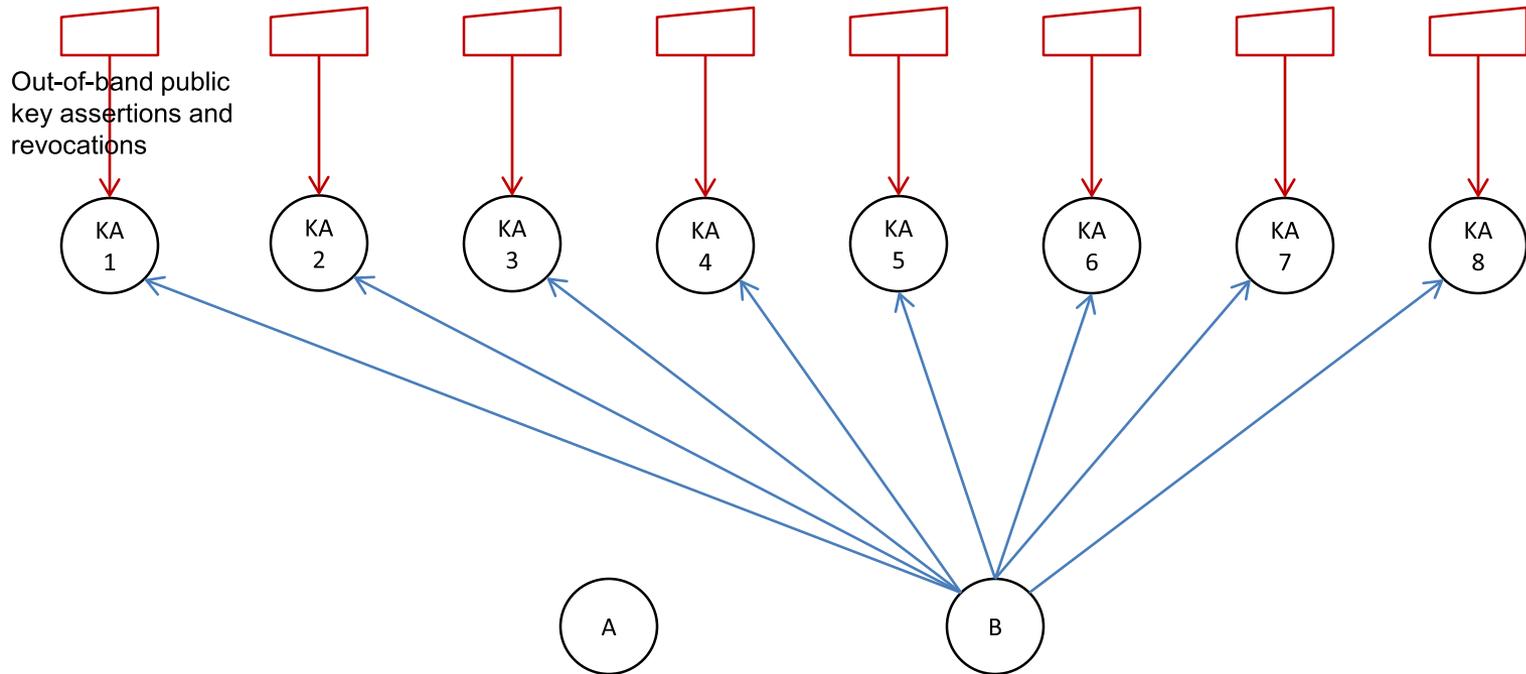
# A Key Management System for DTN



Assume  $N$  key authority (KA) nodes in the network (here  $N=8$ ). All KA nodes have all current key information for the network, as far as possible. All credible assertions of key revocation and reinstatement are provided to KAs by an out-of-band mechanism, such as a human network security analyst.



# Assertion of Public Key by Node B



Each node generates its own public/private key pairs periodically and multicasts its public keys to all KA nodes in advance of effective time.



# Assertion of Public Key by Node B

---

Sign in current private key of B

Message from B: effective  
time, public key



# Erasure-Coded Messages from KA

- Identical reports (assertions, revocations) are generated simultaneously by all KAs, and hashes are computed for the generated reports.
- At each KA, the report is erasure-coded:  $Q + k$  code blocks ( $Q$  primary blocks plus  $k$  parity blocks) are generated such that the reception of **any**  $Q$  different blocks will enable acquisition of the original report.
- At each KA, only a subset of the generated code blocks are transmitted. The distribution of key information relies on the acquisition of code blocks from multiple collaborating key authorities.
- Suppose  $Q=7$  and  $k=1$ , for a total of 8 blocks. Each KA then only multicasts 3 blocks:

KA	0	1	2	3	4	5	6	7
1	x	x	x					
2		x	x	x				
3			x	x	x			
4				x	x	x		
5					x	x	x	
6						x	x	x
7	x						x	x
8	x	x						x

So 24 blocks are multicast. Receiving any 7 distinct blocks will deliver the report (because  $Q = 7$ ).



# Bulletin from KAx

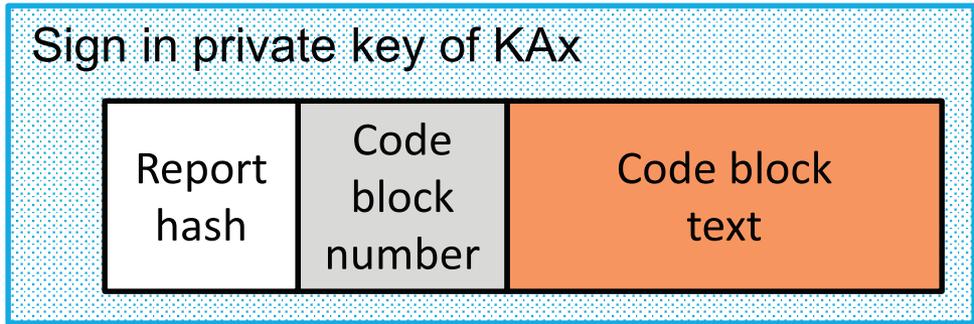
---





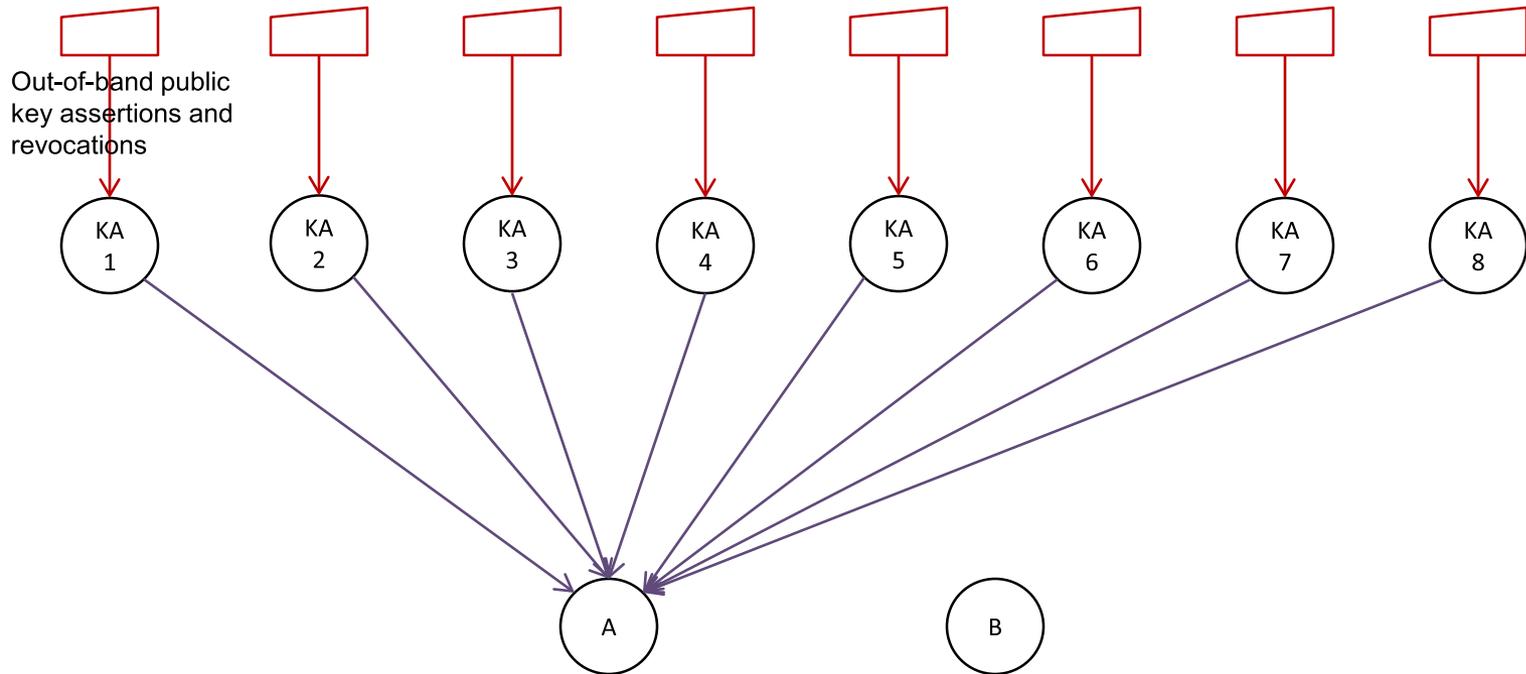
# Code Block of Bulletin from KAx

---





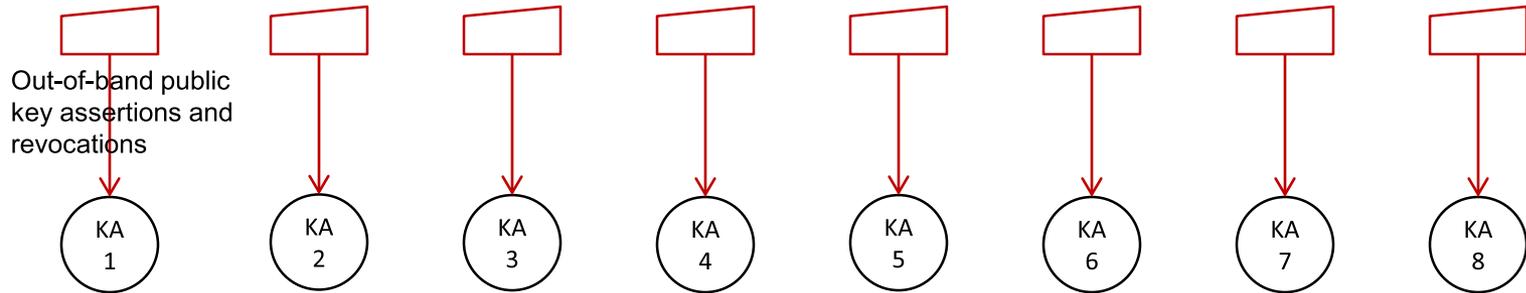
# Publication of Code Blocks from KAs



Only blocks with the same hash will be reassembled into the report, so if any KA is inadvertently out of agreement its report will be ignored.



# Use of New Public Key by Node A



Asserted key information is used for all bundles whose creation time is greater than the asserted effective time.



# Notes

---

- All keys required for Bundle Security Protocol are distributed securely.
- Human intervention in key management (e.g., revocation) is supported, but routine key management is automated.
- No key negotiation/querying over long-latency links.
- No session keys; no management of shared secrets over long-latency links.
- No single point of failure, yet no single point of authority that can be compromised.



# Status

---

- A prototype of this key distribution mechanism has been developed by JPL, is being adopted by the NASA AES program and is under study at Boeing.
  - Uses polarssl for cryptography, zfec for erasure coding.
  - Open-source release is pending.



# Questions?

---





**BACKUP**



# Key Management in DTN (1 of 5)

---

- All nodes can generate their own public/private key pairs.
- A single-use symmetric key – generated by the sender, encrypted in the receiver’s public key, and used to encrypt the payload – can be attached to each bundle.
  - Receiver uses its own private key to decrypt the symmetric key.
  - Receiver uses the symmetric key to decrypt the payload.
- Ephemeral symmetric keys can also be used for computing and verifying payload integrity signatures:
  - The symmetric key – in plain text – is attached to the bundle.
  - The symmetric key is itself signed in the sender’s private key.
  - The signature on the symmetric key is verified using the sender’s public key, proving authenticity.



# Key Management in DTN (2 of 5)

---

- Public keys can be asynchronously asserted by their owners, e.g., by multicast, with associated effective times. No need to query for them.
- **But** nodes can't simply assert public keys to one another: there would be no assurance that a public key asserted for a given node was authentic.
- Suppose a node is physically compromised, to the point at which its current private key is exposed to the attacker.
  - The attacker can now decrypt all confidential information transmitted to this node.
  - The attacker can also induce the node to announce a new public key – paired with a private key that is known to the attacker, and certified in the node's current private key – to the network at any time.



# Key Management in DTN (3 of 5)

---

- The network's only defense against such an assault is to “revoke” the current public key of the compromised node.  
But:
  - Which node can be trusted to issue such a revocation? (Inauthentic key revocations could seriously damage the network.)
  - In order to reinstate the compromised node following its recovery, it would be necessary to once again issue a new public key for that node. But which node can be trusted to issue that reinstatement key? (If the node itself were authorized to issue such a key then the node could reinstate itself while still under the control of the attacker.)



# Key Management in DTN (4 of 5)

---

- Moreover, when new nodes are instantiated, if their initial public keys are self-generated and self-certified then the nodes receiving those keys have no way of knowing whether or not they are authentic and whether the nodes themselves are trustworthy.
- In short, the missing component to DTN security key administration is a central key authority that can be trusted to issue and revoke the public keys of all nodes in the network.
- But the design of such a central key authority must be approached with care...



# Key Management in DTN (5 of 5)

---

- The key authority must not be a single point of failure: loss of the trusted authority would cripple the network.
- But redundancy is not enough, because a key authority node might itself be compromised by an attacker: the distribution of bogus keys to all nodes, or the revocation of all keys, would likewise cripple the network.
  - No single key authority node can be granted unilateral key distribution authority. Key administration proclamations must be issued by agreement among multiple key authority nodes.
- Nor may any single key authority (potentially compromised) be permitted to sabotage key distribution by simply refusing to agree. What's required is consensus, not unanimity.



# Key Effective Time

---

- The start time of a key's validity is a "DTN time" (seconds since 1/1/2000, count within second).
- The selection of a key for operation on a given bundle is based on *bundle creation time*: the system selects the most recently effective key whose start time is before the bundle's creation time.
- Solves the problem of synchronizing transmission and reception key selection.
  - No matter how long the bundle takes to arrive, its creation time is the same as when it was transmitted.



# It's All Public Information

---

- Nothing managed by the key authorities or communicated to or from the key authorities is secret. None of the key agreement interchange needs to be encrypted – just authenticated.
  - This distinguishes DTKA from, for example, threshold cryptography and other distributed shared secret technologies: rather than combining encrypted fragments of a secret data item, DTKA relies on combining signed fragments of a clear-text data item.