A Framework for Secure End-to-End Delivery of Messages in Publish/Subscribe Systems

Shrideep Pallickara, Marlon Pierce, Harshawardhan Gadgil, Geoffrey Fox, Yan Yan, Yi Huang

spallick@indiana.edu

Community Grids Lab, Indiana University

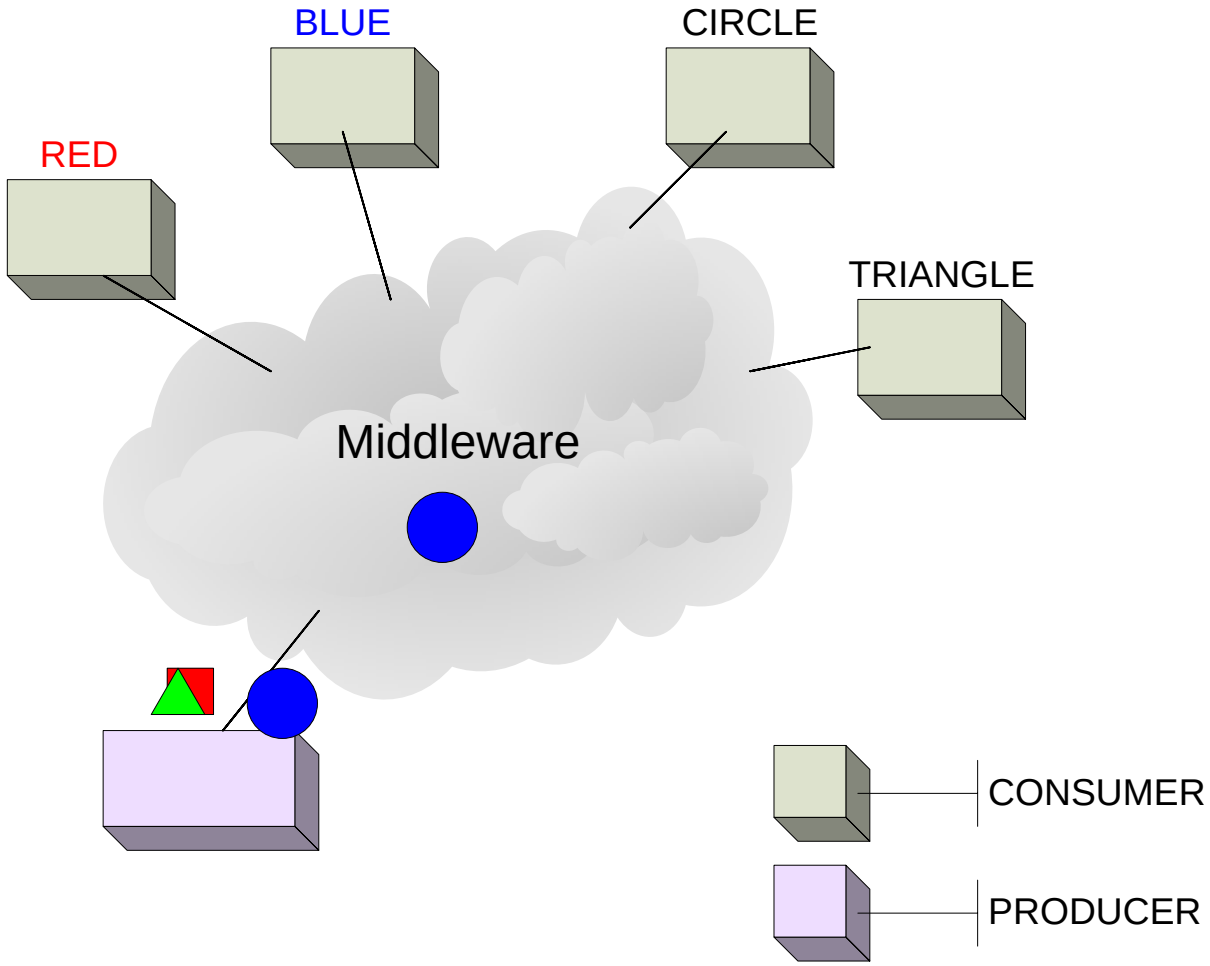
Motivation

As applications have gotten increasingly distributed there is a need for ensuring the secure and authorized distribution of data.

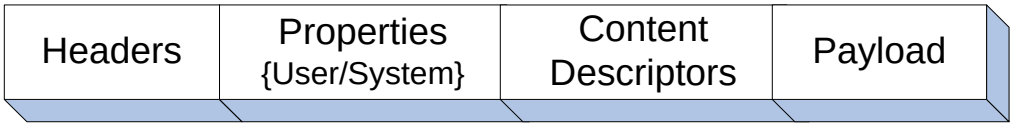
Messaging Systems

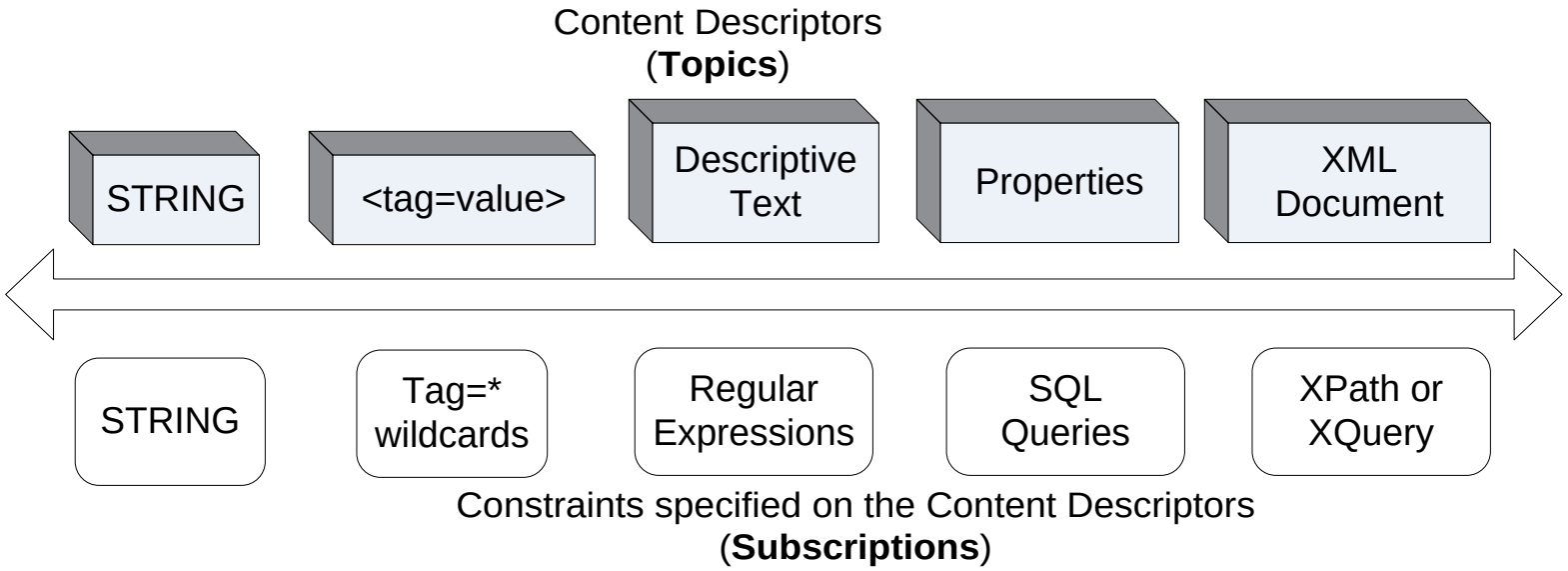
- Messaging is the routing of content from the producer to the consumer.
  - This can be point-to-point or many-to-many.
- Messaging Infrastructures underlie most complex systems
- Approaches to messaging include systems such as queuing, P2P systems and publish/subscribe.

Publish/Subscribe Systems



Messages & Selectivity





Topic Discovery Scheme

- Create topics that are unique in space and time in a decentralized fashion
- Establish topic provenance
  - Deterministic cryptographic verification of ownership
- Restrict discovery of topics to only those that possess valid credentials or within a defined set
- Establish topic life-cycle
- Manage topic collections & organization

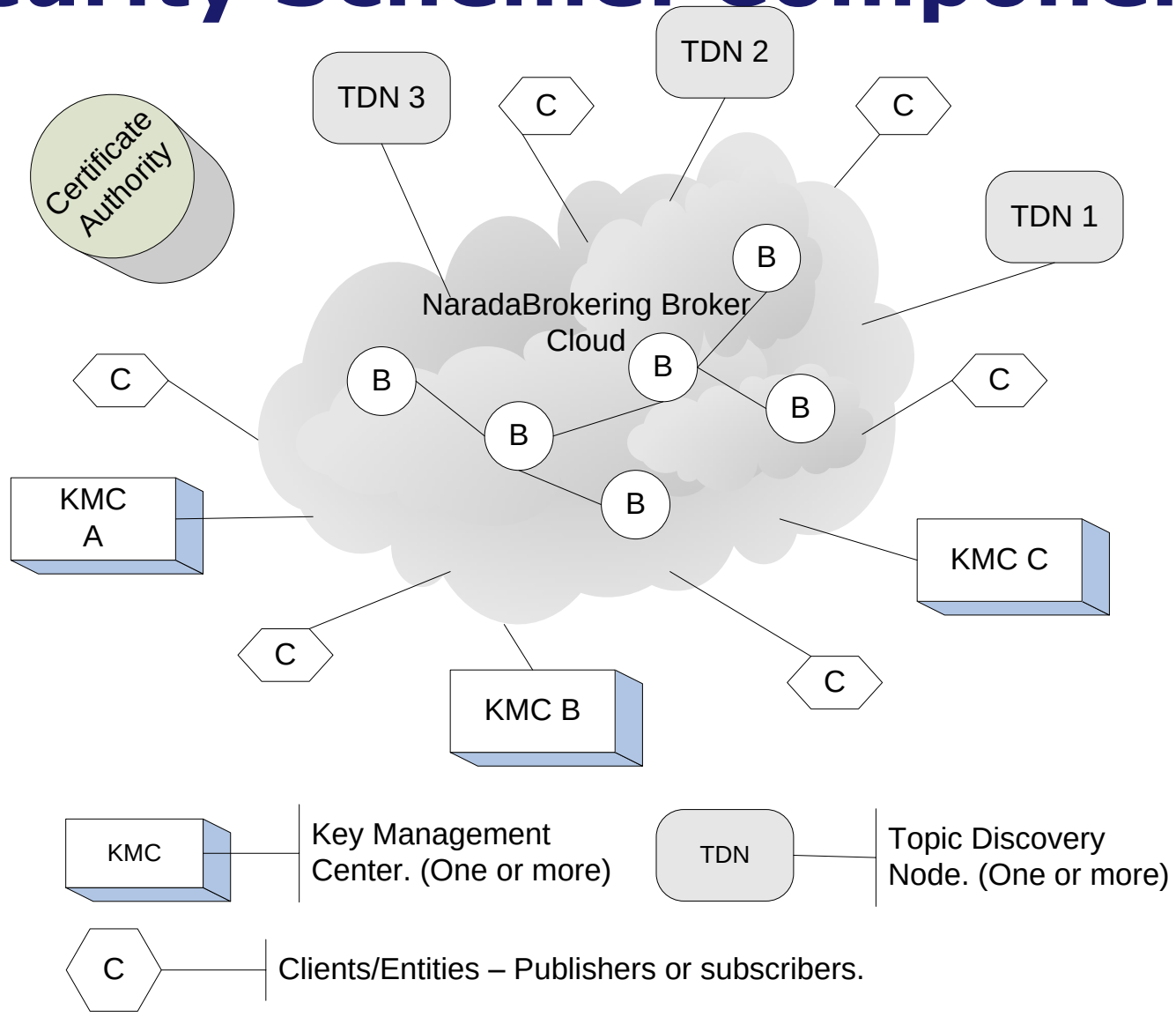
Security Scheme: Desiderata

- Thwart eavesdropping
- Tamper-evidence
- Authorized data generation/consumption
  - Specify allowed actions
  - Duration of rights
- Identity assertion & Non-repudiation
- Transport-independent
- Cope with attack scenarios

Leveraged cryptographic tools

- Symmetric keys for payload encryptions and decryptions
- Message digests for tamper evidence
- PKI for signing and verifications
- For secure "dialogue" between two entities use combination of symmetric and asymmetric keys

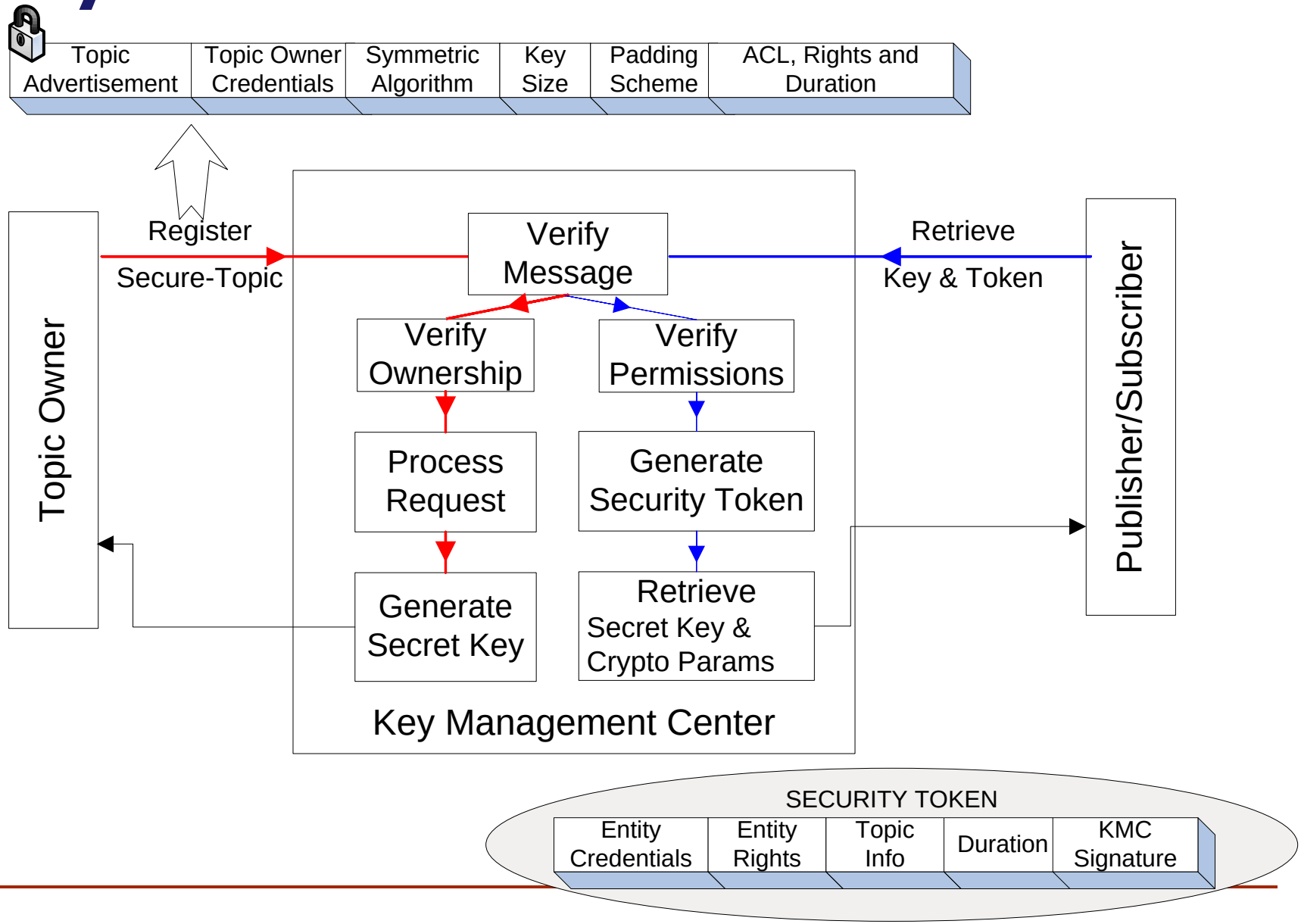
Security Scheme: Components



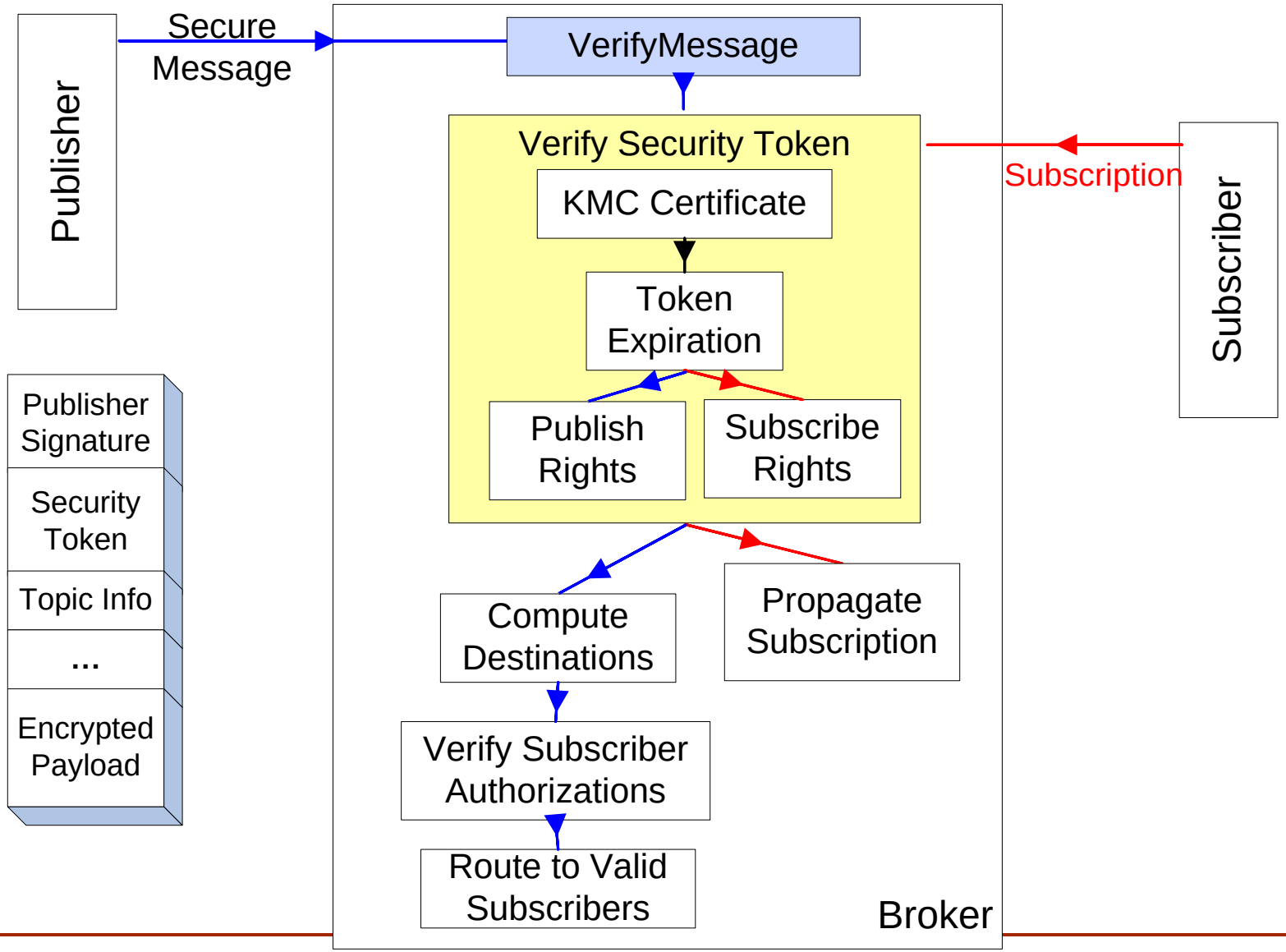
Communications/Interactions

- All interactions are through the exchange of discrete messages
  - Need to know the communication topic
- Entities are selective about who can discover its communication topic(s)
- Topic owner needs to first discover the KMC willing to host the secure topic
  - Based on credentials supplied during discovery
  - Willing KMCs will respond with their communication topics (secured) in the responses

Entity-KMC Interactions



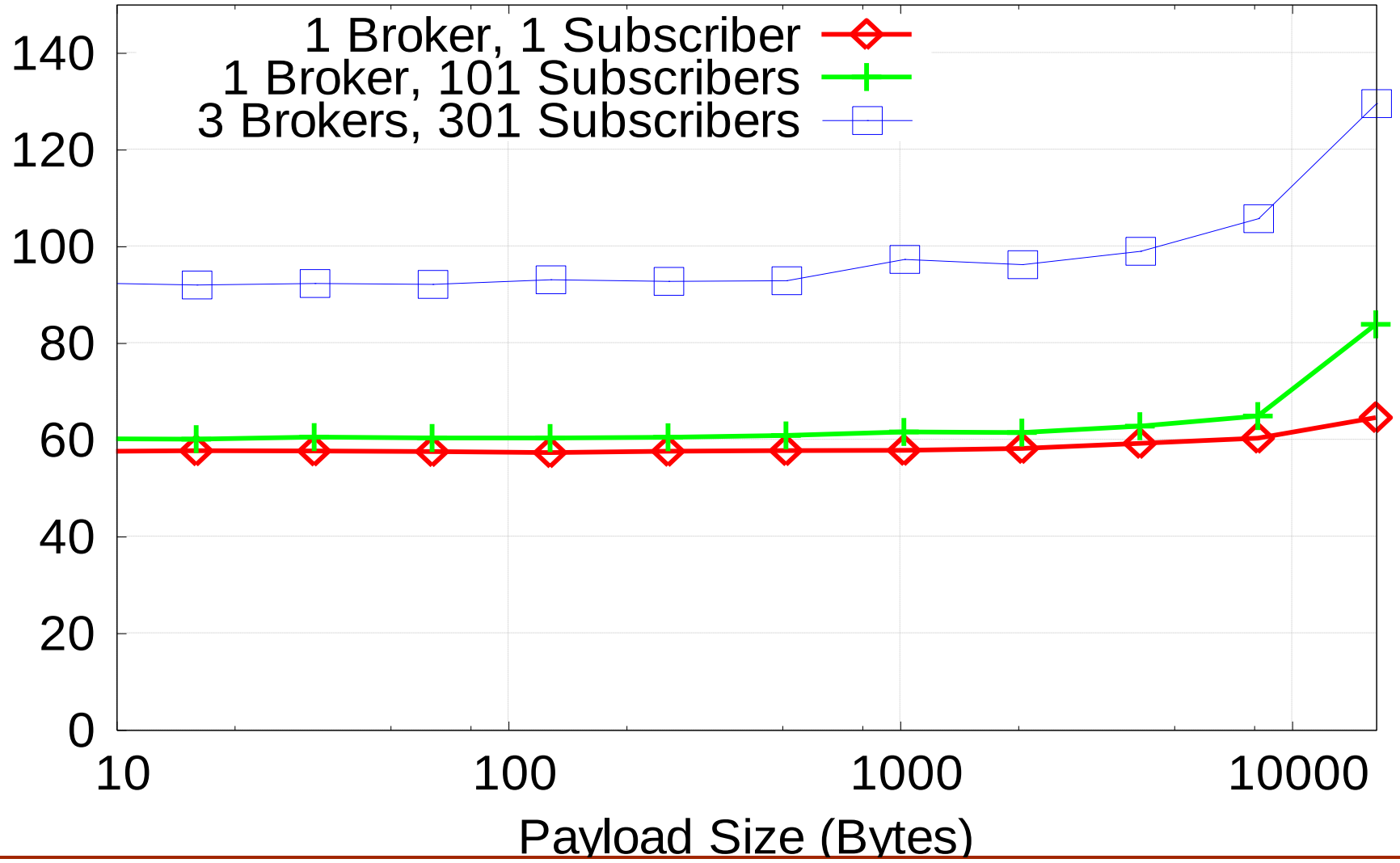
Security Scheme: Broker Processing



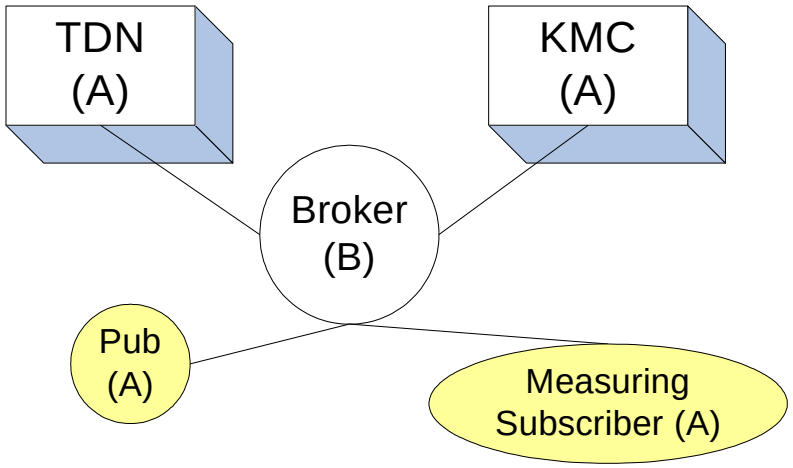
Coping with a couple of attacks

- Denial of Service attacks
  - Unauthorized generation of data is not allowed
  - Deluging KMC is difficult
    - Quite hard to "guess" the 128-bit UUID
    - Network location know only to hosting broker
- Replay attacks
  - For every entity maintain information about last timestamp
    - Discard messages published in the past
    - For higher publish rates, maintain combination of NTP timestamps and message numbers
    - No need to keep track of message identifiers

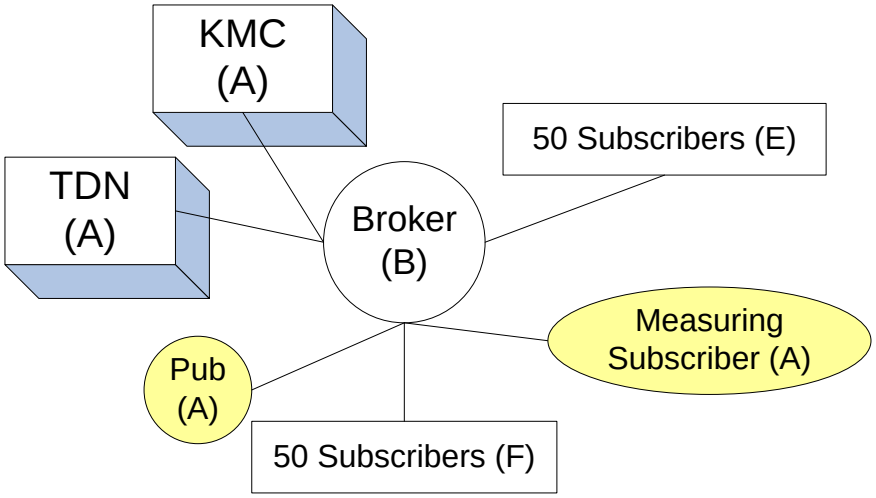
End-to-End Delivery of Messages for different topologies Cryptographic Profile: 256-bit AES, 7PKCS padding 1024-bit RSA keys and 160-bit SHA-1



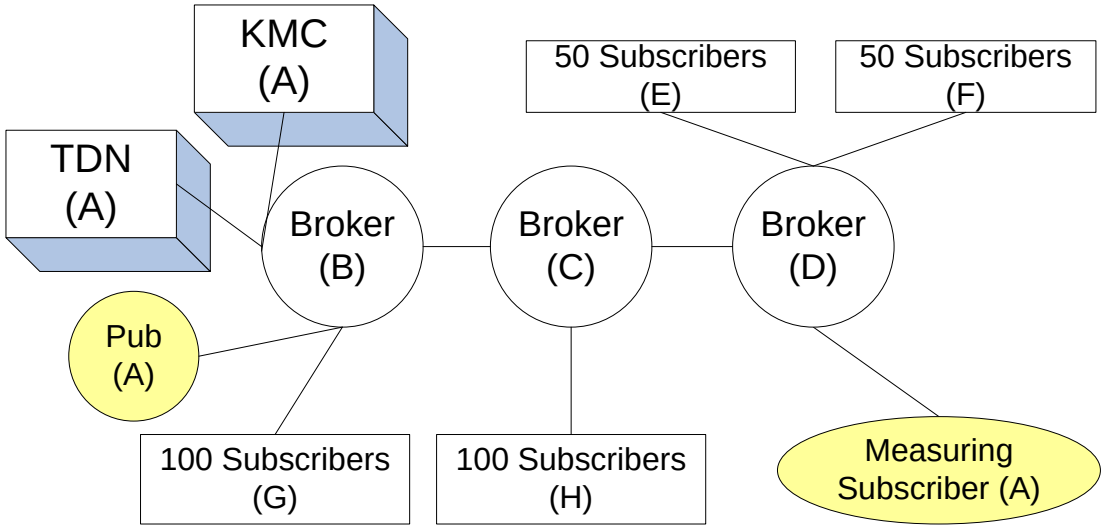
Benchmark Topologies



Topology I: 1 Broker, 1 Subscriber



Topology II: 1 Broker, 101 Subscribers



Topology III: 3 Brokers, 301 Subscribers

Operation		Mean	Standard Deviation	Error
Publisher Costs				
	Initialization Vector	1.108	0.025	0.003
	Encryption	1.421	0.055	0.005
	Signing			
	Payload	15.518	0.126	0.013
	Header	15.238	0.112	0.011
Broker Costs				
	Token and Message Validation	6.989	0.199	0.020
	Replay-attack check	0.031	0.005	0.0
	Subscription validity	0.027	0.004	0.0
Subscriber Costs				
	Verify Token + Header	3.74	0.13	0.013
	Verify Payload	1.64	0.032	0.003
	Decryption	1.41	0.021	0.002

Encryption {AES 256, PKCS7 padding CBC mode}
Signing {1024-bit RSA, 160-bit SHA-1}

Conclusions

- Topic provenance lays the groundwork for the security framework
- Since the scheme is transport independent, it is applicable for systems that can't use SSL
  - E.g. Audio/Video conferencing systems
- Overheads introduced by the security scheme relate to cryptographic operations.
  - No significant increase in message size
  - Jitter introduced by scheme is quite low
- Since the nature of processing is determined by the contents in autonomous messages the system can enforce secure and best-effort schemes equally well.

Future Work

- Detecting security compromises
  - Issue authentication challenges at regular intervals
  - Issue queries from a previously negotiated set of queries/responses during initialization
  - Shorter key lifetimes
- In case of a compromise
  - Compute new keys
  - Propagate compromise info to relevant nodes within the system

Related Work

- GKMP – For Multicast
- Groove – Secure shared spaces
- GSI
- WS-Security