Security in Mobile Ad-Hoc Networks

Robert Kennedy,
George Constantinides,
Zhi Qu,
Zhichao Wu

GOAL

- Try to describe a solution that supports ubiquitous security service for mobile hosts, scales to network size, and is robust against break-ins

Challenge

- Security breach
- Mobility and service ubiquity
- Network dynamics
- Network scale

Why the conventional approaches fail to solve it

- In the centralized approach
- In the hierarchical approach
  - High mobility
  - Multi-hop communication
  - Every local CA is exposed

New Approach

- Distribute the certification authority (CA) functions through a threshold secret sharing mechanism, in which each entity holds a secret share and multiple entities in a local neighborhood jointly provide complete services.
- Update the secret shares to further enhance robustness against break-ins.

New Approach

- We consider a dynamic wireless ad-hoc network with *N* networking hosts/entities.
- Each entity i has a global unique nonzero ID v_i
- *N* may change over time because mobile hosts may join, leave, or crash
- *N* is no limited, may be a large number of communicating entities.

Architecture

- RSA

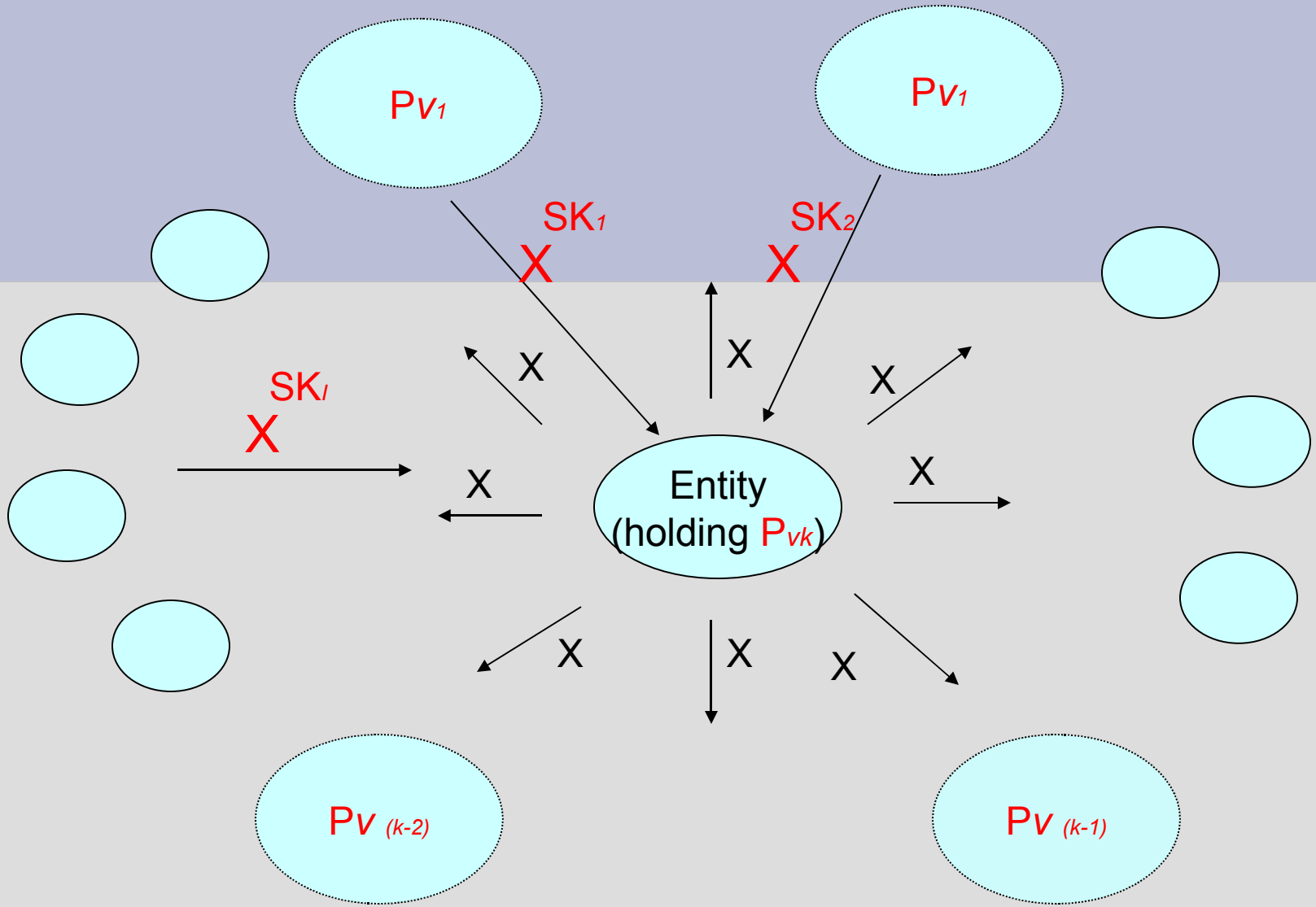
  CA's key pair: {SK,PK}

  each entity *vi* : {sk*i*,pk*i*}

- Secret sharing

  Each entity holds a secret share P*vi*, any K (1<K<N) of such secret share holders can collectively function as CA

  certificate <v*i*, pk*i*, T*sign*, T*expire*>,only valid if signed by SK



$$X^{SK_1} * X^{SK_2} \cdots X^{SK_K} = X^{SK_1 + SK_2 + \cdots + SK_K}$$

$$d = \sum_{j=1}^{K} SK_j \bmod n \longrightarrow SK = \langle d, n \rangle$$

Protocols

- **Certification service**

  Certificate issuing: need centralized trusted management

- **Localized self-initialization**

  Any certificate-holding entity can obtain a secret share from a local coalition of K share holders

- **Secret share update**

  New version periodically updated is propagated by the self-initialization protocol.

- Bootstrapping phase: valid certificate for each entity
- Initialize (at least) K local coalition

  (centralized dealer not needed any more)
- Self-initialization: distributing secret share
- Renew certificate or renew secret share through self-initialization

Evaluation of implementation (1)

• Design is realized in both Unix and NS-2, a popular network simulator

UNIX:

Seeks to quantitatively characterize the computational cost of the work

NS-2:

Evaluate aspects of mobility, ability to handle ubiquitous service, channel and node dynamics, in a large network setting

Evaluation of implementation (2) Unix

key (bit)	RSA-PK (msec)	RSA-SK (sec)	PCC (sec)	Combine (sec)
512	0.093	0.0056	0.0466	0.0928
768	0.124	0.0173	0.1198	0.2416
1024	0.142	0.0386	0.2610	0.5280
1280	0.136	0.0669	0.4590	0.9742
1536	0.133	0.1089	0.7944	1.5598
2048	0.208	0.2462	1.7058	3.4410

Table 1. RSA and certification performance ($K = 5, SPEC = 20.5$)

key (bit)	RSA-PK (msec)	RSA-SK (sec)	PCC (sec)	Combine (sec)
512	0.884	0.0678	0.1835	0.1982
768	1.276	0.2165	0.5973	1.3430
1024	1.324	0.4672	1.1637	1.1978
1280	1.356	0.8734	2.2912	2.4109
1536	1.416	1.4863	3.5820	3.6952
2048	1.036	3.1883	7.7855	8.0324

Table 2. RSA and certification performance ($K = 5, SPEC = 12.1$)

key (bit)	RSA-PK (msec)	RSA-SK (sec)	PCC (sec)	Combine (sec)
512	2.782	0.2347	0.5499	0.6144
768	3.382	0.6403	1.4818	1.6478
1024	4.036	1.2953	3.1738	3.3283
1280	4.065	2.4607	5.5492	5.9019
1536	3.941	3.8543	10.1253	10.4301
2048	3.954	8.3826	20.6606	21.7095

Table 3. RSA and certification performance ($K = 5, SPEC = 1.37$)

- Computation power is a critical factor for the efficiency of the RSA-based scheme

Evaluation of implementation (3) Unix

K	$SPEC = 20.5$		$SPEC = 12.1$		$SPEC = 1.37$	
	PCC	Combine	PCC	Combine	PCC	Combine
2	0.260	0.526	1.293	1.334	2.991	3.304
3	0.261	0.528	1.149	1.171	2.998	3.293
5	0.261	0.528	1.164	1.198	3.174	3.328
7	0.263	0.531	1.140	1.207	3.163	3.530
10	0.262	0.537	1.309	1.410	3.099	3.394
20	0.261	0.532	1.308	1.464	3.078	3.458
30	0.261	0.537	1.160	1.510	3.082	3.410

Table 4. Certification performance in terms of system parameter K (RSA key: 1024bit, time unit: sec)

key (bit)	$SPEC = 20.5$		$SPEC = 12.1$		$SPEC = 1.37$	
	PSS	Sum	PSS	Sum	PSS	Sum
512	0.413	0.288	1.145	0.378	3.861	1.196
768	0.459	0.382	2.588	0.443	5.163	1.497
1024	0.490	0.319	3.321	0.781	7.024	1.847
1280	0.561	0.411	4.926	0.840	8.215	1.996
1536	0.798	0.460	3.480	0.630	10.251	2.006
2048	1.420	0.473	5.245	0.754	24.414	2.528

Table 5. Self initialization service ($K = 5$, time unit: msec)

- Table 4: K does not affect performance
  - Partial certificates computed in parallel
  - Moderate operations at the requester's side
- Table 5: Operations used in self-initialization are inexpensive to compute

Evaluation of implementation (4) NS-2

- Metrics:
  - *Success ratio:* ratio of the number of successful certification services over the number of attempts during the simulation time.
  - *Average delay:* average latency for each node to perform a certification service
  - *Average number of failures:* number of times an entity fails on average before successfully completing its certification
- Network size: 30-100 nodes
- Node mobility: 1-20m/s
- Expiration time for certificate: 5 minutes

Evaluation of implementation (5) NS-2

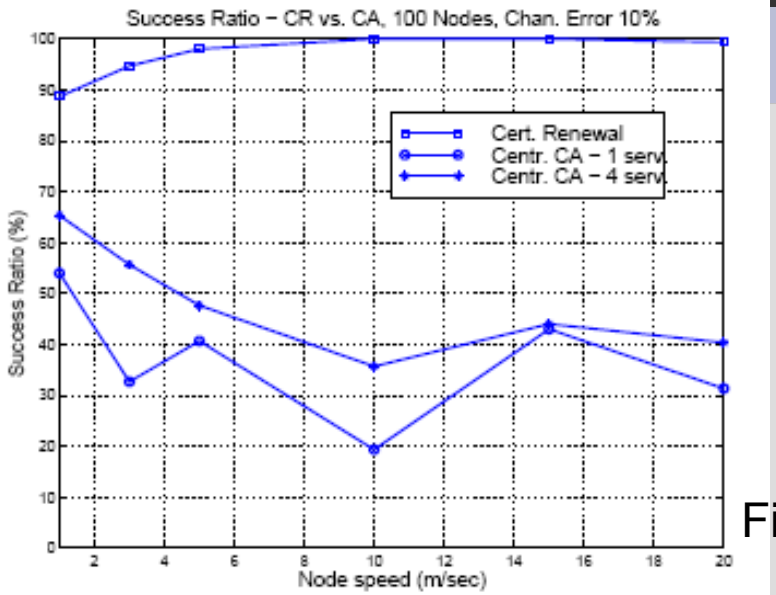


Fig. 1

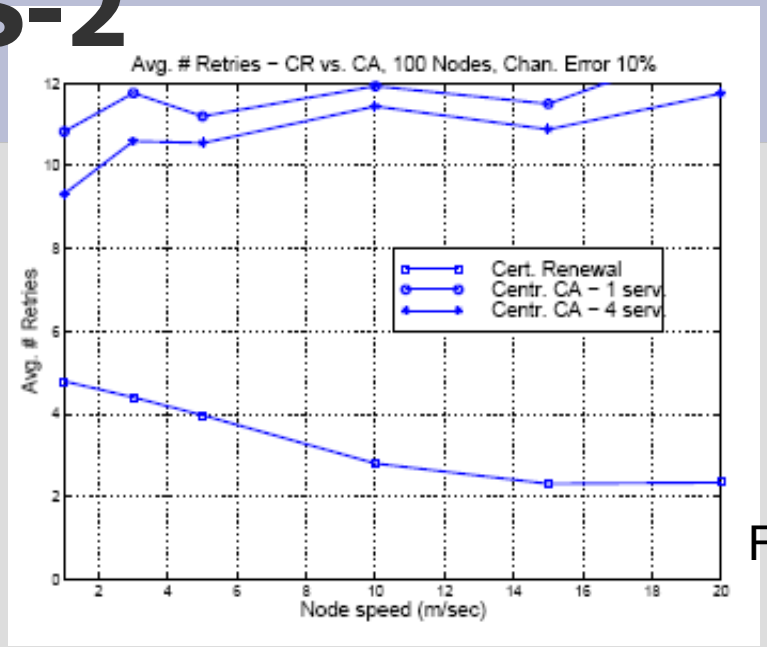


Fig. 2

- Error rate 10%
- Mobility helps the protocol
  - ➢ Requesting entity which is roaming may not have K one-hop neighbors at one time
  - ➢ Moving to new location means finding at least K share holders to server it.

Evaluation of implementation (6) NS-2

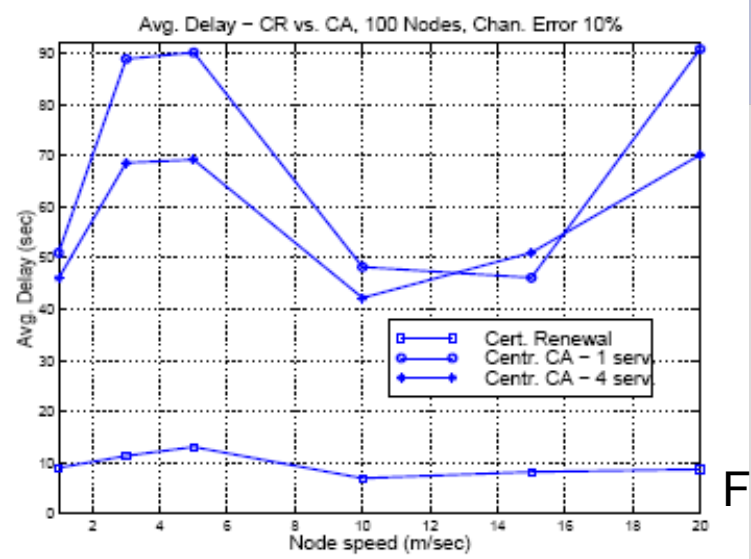


Fig. 3

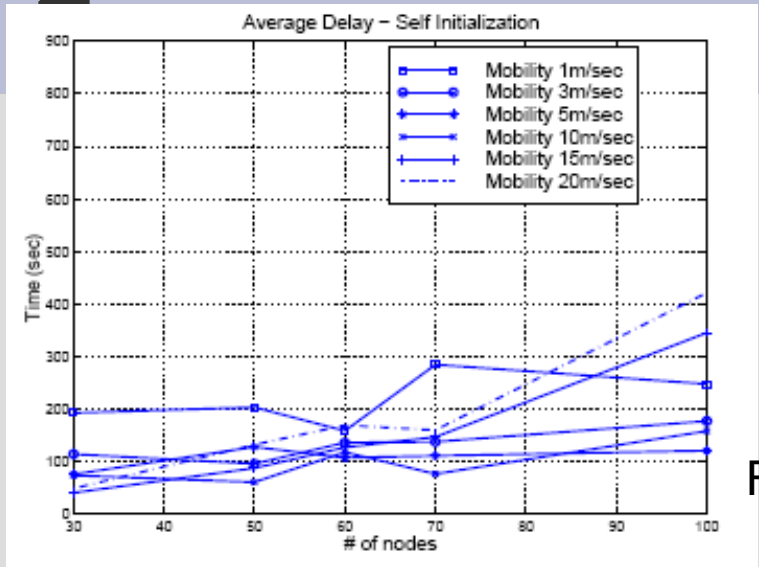


Fig. 4

- Fig. 3: Centralized and hierarchical solutions incur much higher delay which fluctuates
- Fig. 4: 2*K nodes initialized with an imaginary dealer; average latency for each node to self-initialize with a coalition of K neighbors

Critical Appraisal (1)

- Aims of paper for proposed solution:
  - Scalable
  - Intrusion-tolerant
  - Dynamically changing network topology
- Scalability: Certificate renewal is tolerable but self-initialization suffers
  - Asymmetric cryptography costly
  - What about channel errors / node failures?

Critical Appraisal (2)

- Intrusion-tolerant: As discussed in paper algorithm prevents passive attacks.
  - BUT what about active attacks?
- Dynamically changing network topology
  - Value of K: Trade off between availability and robustness. BUT not clear on optimal value of K based on nodes significantly increasing/decreasing
  - Error rate fixed at 10%
  - What about channel interference?

Summary : Objectives

Objective was to design a distributed security solution for dynamically changing wireless ad-hoc networks that would be:

- Robust to break-ins
- Scalable to network sizes
- Able to support ubiquitous service availability

Summary : Approach (1)

- Employed a certificate-based approach based on the public key infrastructure, to provide security.

- Distributed certificate-authority functionality amongst nodes in each local neighbourhood.
  - Allowing for service ubiquity to roaming users
  - Avoiding problems associated with centralised certificate-authorities

Summary : Approach (2)

- Used 'threshold secret sharing' to distribute CA authority amongst nodes.
  - Each of k nodes in a neighbourhood hold secret share.
  - K nodes form coalition and use their secret shares in combination, to provide CA functionality.
  - System security not compromised as long as there are less than K nodes compromised at any one time.
- Self-Initialisation of nodes.
  - Apart from initial bootstrap period, nodes that join the network can be initialised by k neighbours.
  - Once initialised, a node can serve as coalition member.
  - Provides scalability and allows for mobility.

Summary : Implementation

- Results of implementation generally positive.
  - Protocol performs favourably in their tests against centralised approaches.
  - Size of K does not significantly affect performance which is good news for scalability.
  - Mobility actually helps performance of protocol.
- Evaluation.
  - Their de-centralised approach was designed to be robust, scalable, and ubiquitously available to mobile users.
  - Their tests strongly suggest that the protocol performs efficiently and effectively with regard to these objectives.

Related Work

Self-Organized Public-Key Management for Mobile Ad Hoc Networks ~Capkun et al. 2003

- Fully self-organised public-key management with no centralised server needed, even on initialisation .
- Certificates Stored in local repositories on each node.
- Nodes compare certificate repositories to authenticate each other. Detection of false certificates through conflicts.

INSENS: Intrusion Tolerant Routing in Wireless Sensor Networks ~Deng et al. 2002

- Protocol designed to provide intrustion tolerant routing in sensor networks..
- Use of redundant multipath routing to bypass mailicious nodes.

Questions?