Secure Admission Control in P2P and MANETs

Jeong H. Yi

jhyi@ics.uci.edu

Computer Science Department

University of California at Irvine



1

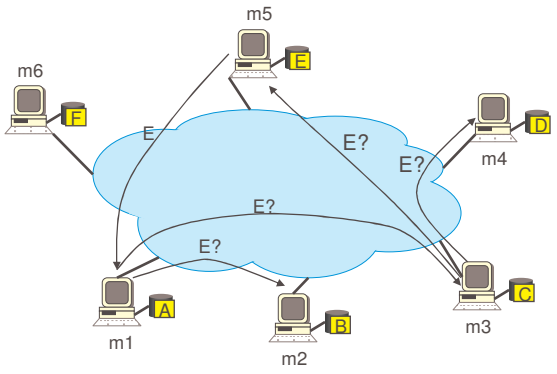
Outline

- Introduction
- Motivation
- Definition
- Work done
  - Analysis of existing schemes
  - Design and implementation
  - Performance evaluation
- Future work

2

Decentralized P2P: Gnutella

- Assume: m1's neighbors are m2 and m3; m3's neighbors are m4 and m5;…



3

(Potential) P2P Advantages

- Allows for scalable incremental growth
- Aggregate tremendous amount of computation and storage resources
- Tolerate faults or intentional attacks
- etc...

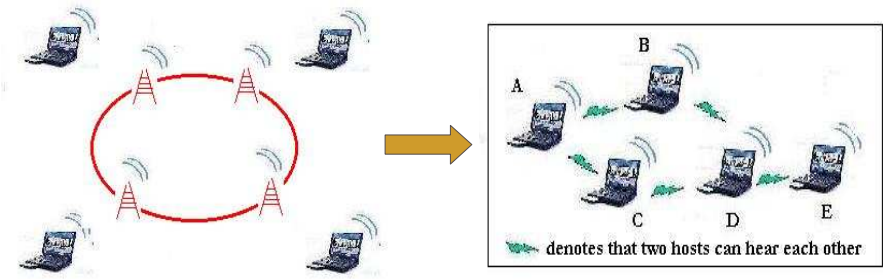
4

P2P Applications

- File sharing [CFS, OceanStore, PAST, ...]
- Web cache [Squirrel, ..]
- Backup store [HiveNet]
- Web archive [Herodotus]
- Censor-resistant stores [Eternity,FreeNet,..]
- Event notification [Scribe]
- Naming systems [ChordDNS, INS, ..]
- Query and indexing [Kademlia, ...]
- Communication primitives [I3, ...]

5

Ad Hoc Networks



- Infrastructure-less
- Distributed architecture
- Self-organizing
- Wireless connections

6

Classification

- **Mobile Ad-Hoc Networks (MANETs)**
  - human or vehicle based nodes
  - high mobility
- **Personal Area Networks**
  - communication between personal devices: PDAs, earphones...
  - medium mobility
- **Sensor Networks**
  - tiny nodes for collecting data /w border gateway
  - low relative mobility

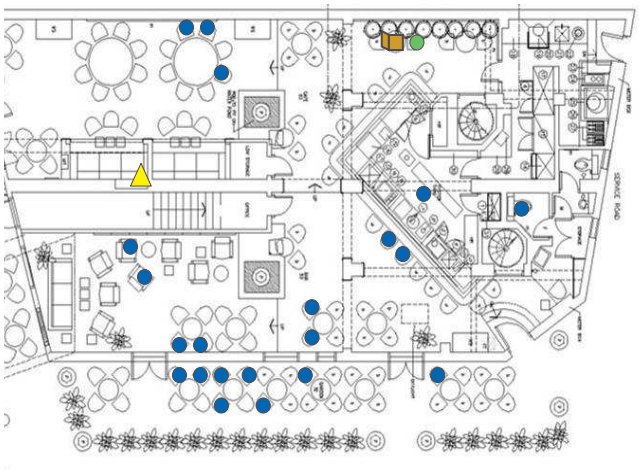
7

MANET Example

- It is a nice warm day in the University Cafe...
- All the students

  are happy, drinking some coffee, discussing which project group to attend.
- When suddenly somebody switches on his laptop

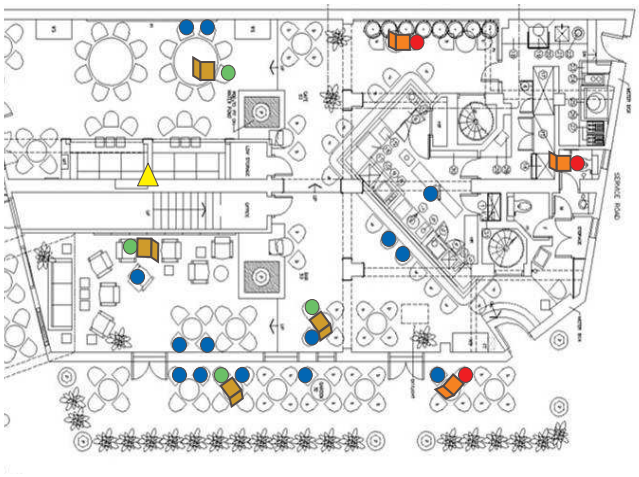
  ... and surfs the Internet while having a cup of coffee
- ... obviously somewhere there is a W-LAN base station



8

MANET Example

- The next day …
  - … more students
  - … connect to W-LAN
- But some of them
  - … have severe connection problems
- … **Something must be done**

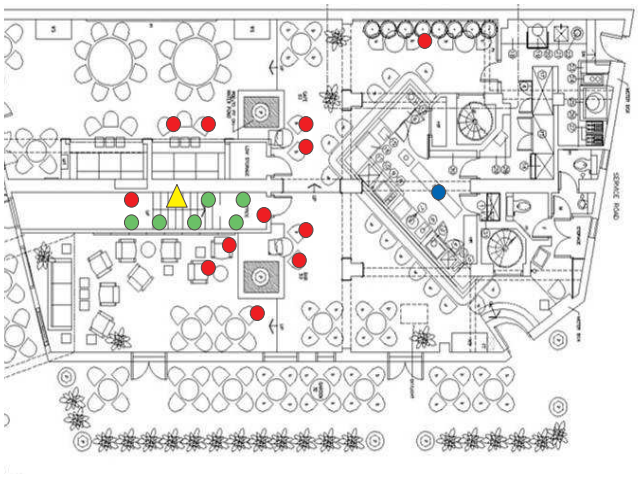


9

MANET Example

- So they move closer to the base station
  - … and closer
- The lucky ones establish a connection on the stairway
- The others

  can not connect and waste some time …



10

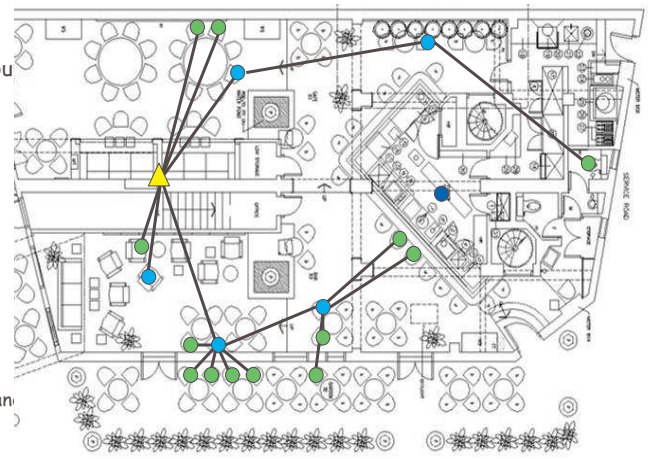
MANET Example

- … and so they live unhappily together until
- … some day a project grou implements a

Mobile Ad-Hoc Network based on Wireless LAN

- … and saves the day.
- Now some

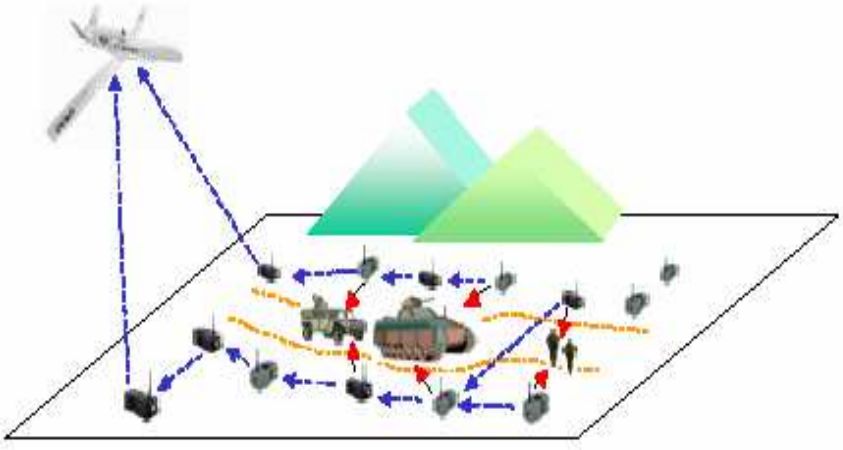
… act as client and router an shared their bandwidth to connect others



11

Sensor Networks

- Military Surveillance



12

Ubiquitous Computing

- many devices per person, becoming interconnected



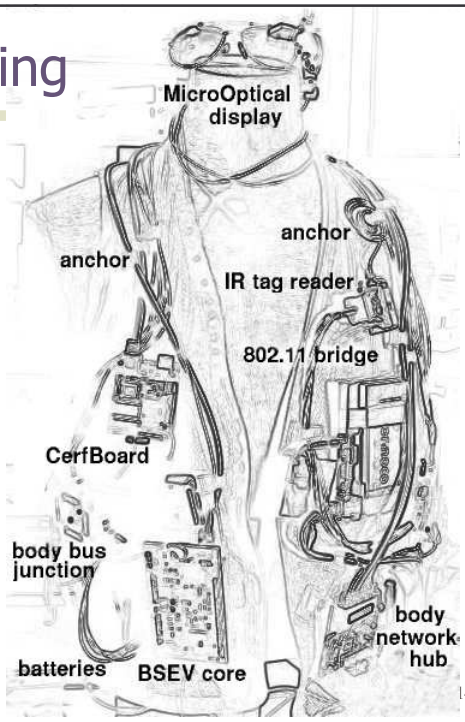
13

Wearable Computing

MIThril is a wearable computing project. Its goal is to build a computer system that you can wear, instead of a computer system that you carry

Source :
http://www.media.mit.edu/wearables/mithril/





14

State of The Art

- IBM ThinkPad 560X
  - 233 MHz Pentium
  - 64 MB DRAM
  - 340 MB HD





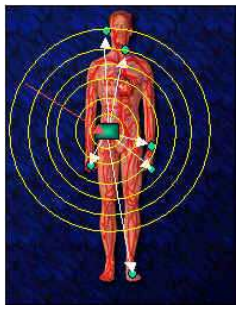
15

Bio-Digital Convergence

Wearable Computing



Computer/sensors embedded in the environment



- **Always ON**
- **Portable while Operation**
- **Hands-free Use**
- **Sensors**
- **Proactive**

Computers/sensors on **people**
Running all the time
Hands Free

16

Many Applications

- Civilian environments
  - taxi cab network
  - meeting rooms
  - sports stadiums
  - boats, small aircraft
- Military environments
  - soldiers, tanks, planes
- Personal area networking
  - cell phone, laptop, ear phone, wrist watch
- Emergency operations
  - search-and-rescue
  - policing and fire fighting

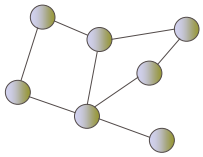
17

Outline

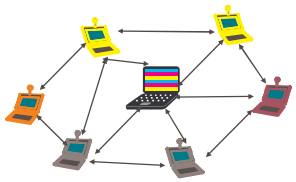
- Introduction
- **Motivation**
- Definition
- Work done
  - Analysis of existing schemes
  - Design and implementation
  - Performance evaluation
- Future work

18

Peer Group Settings



Decentralized P2P



MANETs

- No centralized control
- No hierarchy
- Fault-tolerant
- Dynamic membership

Distributed and scalable security services required

19

Node Misbehavior

- Selfish Nodes
  - Do not cooperate
  - Priority: battery saving
  - No intentional damage to other nodes.
- **Attacks:**
  - passive denial of service
  - black hole
  - idle status

- Malicious Nodes
  - Goal: damage to other nodes
  - Battery saving is not a priority
- **Attacks:**
  - active denial of service
  - traffic subversion
  - attacks exploiting the security mechanism

20

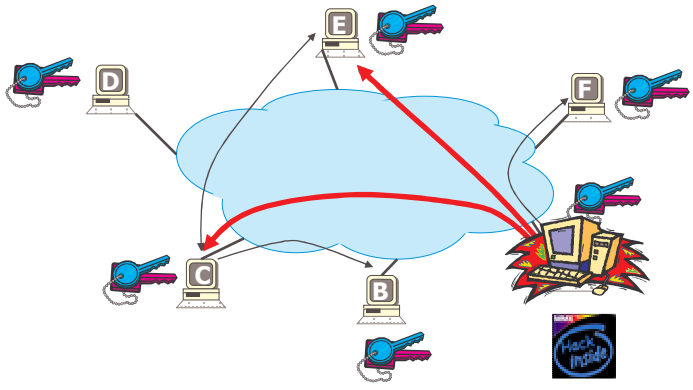
Peer Group Security: Prior Work

Secure communication:

- Key management
- Authentication
- Anonymity
- Secure routing

21

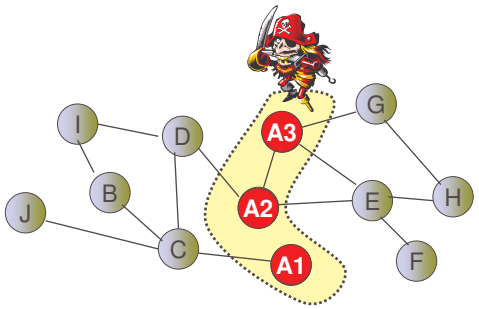
Key Management



22

Sybil attack

- Douceur [IPTPS'02]
- An adversary may create multiple identities



Lesson: Verify identities

23

Motivation

- Secure group communication does not address membership eligibility
- Without *secure admission control*, secure communication (e.g., key management) is useless



24

Outline

- Introduction
- Motivation
- **Definition**
- Work done
  - Analysis of existing schemes
  - Design and implementation
  - Performance evaluation
- Future work

25

Terminology

- Group Charter
  - defines admission policies



- Group Membership Certificate (GMC)
  - proves membership



- Group Authority (GAUTH)
  - Bootstrapping entity



26

Admission Control Models

- **Admission via Public ACL**
  - Not suitable for *dynamic* peer group



- **Admission by Centralized Authority**
  - Violates *peer* nature
  - Single point of failure



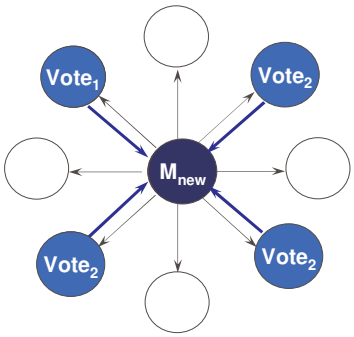
- **Our Focus: Admission by Members**



27

Admission Control

- New member (M_{new}) wants to join the group
- A quorum of t current members need to issue M_{new} a group membership certificate (GMC)
- If no quorum found, membership is denied



- Step 1: Join request
- Step 2: Join commit (Vote)
- Step 3: GMC issuance

28

Threshold Types

- Fixed Threshold
  - Expressed as minimum # of votes (e. g., 5)
  - What if *group size* < threshold?
- Dynamic Threshold
  - Expressed as percentage of # of current members (e.g., 30%)
  - Threshold = percentage * *group size*
  - Keeping an updated group size is important.



Ø *Group Authority (GAUTH), as bootstrapping node, is only trusted to keep account of group size.*

29

Relevant Cryptographic Techniques

- **Plain Signatures (PS)**
  - RSA, DSA, etc.
- **Accountable Subgroup Multi-signatures (ASM)**
  - e.g., Ohta, et al. [CCS'01]
  - Based on Schnorr signature scheme
- **Threshold RSA Signatures (TS-RSA)**
  - e.g., Kong, et al. [ICNP'01]
  - Dealer is required for initialization
- **Threshold DSA Signatures (TS-DSA)**
  - Extension of scheme by Gennaro, et al. [Eurocrypt'96]
  - # of group members (n) can be increased

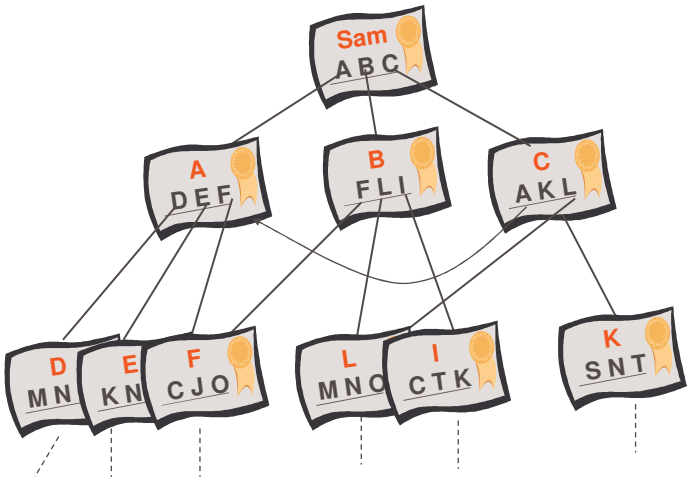
30

Feature Summary

	PS	ASM	TS-RSA	TS-DSA
Dealer involved			√	
Simultaneous on-line presence		√	√	√
Accountability	√	√		
Unlinkability			√	√
Verifiability of secret share	NA	NA		√
Lineage problem	√	√		

31

Membership Authentication: Lineage Problem



32

Outline

- Introduction
- Motivation
- Definition
- **Work done**
  - Analysis of existing schemes
  - Design and implementation
  - Performance evaluation
- Future work

33

Work Done: Existing Scheme Analysis

- *On the Utility of Distributed Cryptography in P2P and MANETs: The Case of Membership Control*
  - M. Narasimha, G. Tsudik, J. H. Yi
  - IEEE ICNP'03
  - Broke threshold RSA scheme
  - Proposed new threshold DSA scheme

34

Work Done: Implementation

- *Experimenting with Admission Control in P2P*
  - N. Saxena, G. Tsudik, J. H. Yi
  - WADIS'03
  - Designed and implemented "**Bouncer**" toolkit
- Related APIs
  - PS_APIs: implementation of plain RSA, May 2002
  - ASM_APIs: implementation of ASM, June 2002
  - TS-RSA_APIs: implementation of TS-RSA, July 2002
  - TS-DSA_APIs: implementation of TS-DSA, April 2003

Source code available at http://sconce.ics.uci.edu/gac

35

Work Done: Performance Evaluation

- *Admission Control in Peer-to-Peer: Design and Performance Evaluation*
  - N. Saxena, G. Tsudik, J. H. Yi
  - ACM SASN'03
  - Integrated "**Bouncer**" with Gnutella and Secure Spread
  - Measured performance of some existing schemes

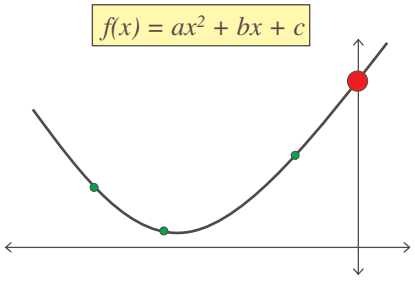
36

Outline

- Introduction
- Motivation
- Definition
- Work done
  - **Analysis of existing schemes**
    1. **Threshold Secret Sharing**
    2. Threshold RSA signatures (TS-RSA)
    3. Threshold DSA signatures (TS-DSA)
  - Design and implementation
  - Performance evaluation
- Future work

37

Lagrange Interpolation



- Given *t* points in the plane, there is only one polynomial of degree *t-1* that intersects all *t* points
- *t-1 points do not suffice for interpolation*

38

Shamir's Secret Sharing

- Dealing Secret Shares
  - Dealer randomly selects polynomial $f(x)$ of degree $t-1$

$$f(x) = S + a_1 x + a_2 x^2 + \ldots + a_{t-1} x^{t-1} \pmod q$$

$$ss_i = f(id_i) \pmod q$$

  - Note: $f(0) = S$
  - Dealer distributes secret shares to users
- Secret Recovery

$$S = \sum_{i=1}^{t} ss_i l_i(0) \pmod q \qquad l_i(x) = \prod_{j=1, j\neq i}^{t} \frac{x - id_j}{id_i - id_j} \pmod q$$

- Distributed Share Computation

$$ss_{new} = \sum_{i=1}^{t} ss_i l_i(id_{new}) \pmod q$$

What if users are malicious?

39

Verifiable Secret Sharing (VSS)

- P. Feldman [FOCS'87]
- Select f(x) over Z_q as in Shamir's

$$f(x) = a_0 + a_1 x + a_2 x^2 + \ldots + a_{t-1} x^{t-1} \pmod q$$

- Setup
- - Exponent is in **mod q**
  - q, p : large prime
  - q | p-1
- Secret share verification

$$g^{ss_i \pmod q} = \prod_{j=0}^{t-1} (W_j)^{(id_i)^j} \pmod p$$

$$\left(g^{f(id_i)} = g^{a_0 + a_1 (id_i) + \Lambda\, a_{t-1} (id_i)^{t-1}} \pmod p \right)$$

40

Outline

- Introduction
- Motivation
- Definition
- Work done
  - **Analysis of existing schemes**
    1. Threshold Secret Sharing
    2. **Threshold RSA signatures (TS-RSA)**
    3. Threshold DSA signatures (TS-DSA)
  - Design and implementation
  - Performance evaluation
- Future work

41

RSA Signature scheme

- Setup
  - Let $N=pq$, where p, q are large primes
  - $d = e^{-1} \bmod \Phi(N)$, where $\Phi(N)=(p-1)(q-1)$
  - d: secret key, (e, N): public keys
- Signature generation
  - $c = m^d \pmod N$
- Verification
  - $m \overset{?}{=} c^e \pmod N$

42

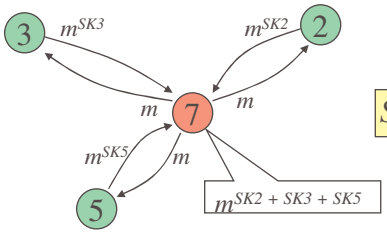
Threshold RSA (TS-RSA)

- J.Kong, et al. [
- Setup
  - Generate RSA
  - Dealer randomly selects polynomial *f(x)* of degre

- d is never reconstructed.
- mod N (composite)

$$f(x) = d + a_1 x + a_2 x^2 + \ldots + a_{t-1} x^{t-1} \pmod N$$

- Signature generation

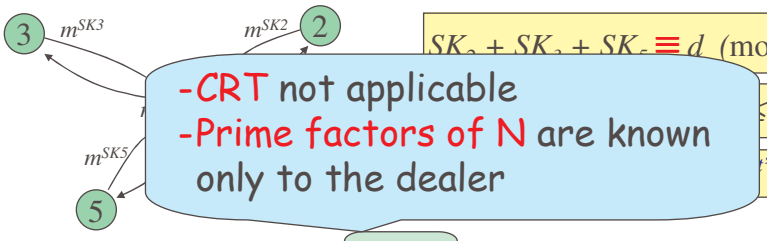


$$SK_i = ss_i l_i(0) \pmod N$$

$$SK_2 + SK_3 + SK_5 \equiv d \pmod N$$

$$m^d \equiv \prod_{i=1}^{t} m^{SK_i} \pmod N$$

TS-RSA: t-bounded offsetting



$SK_2 + SK_3 + SK_5 \equiv d \pmod N$

$\leq t' < t)$

$^{t'N}m^d$

- CRT not applicable
- Prime factors of N are known only to the dealer

22 msec

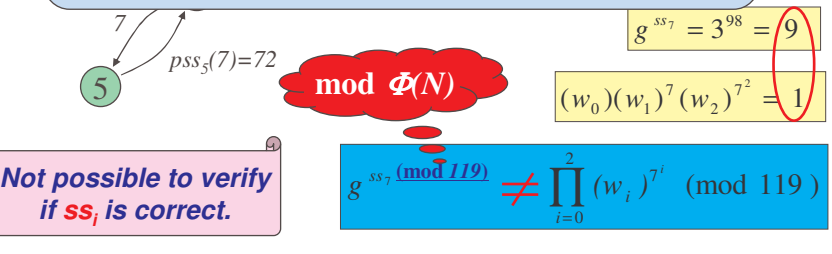
```
Y = m^(t'N+d);
offset= m^(-N) mod N;
for (i=0; i <= t; i++)   {
  if (Y^e = m  mod N)
       break;
       Y = Y * offset mod N;
}
return Y   (= m^d  mod N)
```

TS-RSA: VSS failure

- Example: $f(x) = 77 + 2x + 5x^2 \pmod{119}$, $g=3$
- Witnesses: $w_1 = 3^{77} = 12$, $w_2 = 3^{2^2} = 9$, $w_3 = 3^{5^2} = 5 \pmod{119}$

Not possible to detect malicious members in the group

↳ No robustness provided.



Not possible to verify if ss_i is correct.

TS-RSA: Summary

- No verifiability of secret shares
  - Gennaro [Crypto'96] and Shoup [Eurocrypt'00]: schemes to provide verifiability
  - Boneh & Franklin [Crypto'97]: distributed RSA key generation
    - very high communication and/or computation overhead
    - impractical in group setting such as MANETs
- Trusted dealer involved at initialization

Outline

- Introduction
- Motivation
- Definition
- Work done
  - **Analysis of existing schemes**
    1. Threshold Secret Sharing
    2. Threshold RSA signatures (TS-RSA)
    3. **Threshold DSA signatures (TS-DSA)**
  - Design and implementation
  - Performance evaluation
- Future work

47

DSA Signature scheme

- Setup
  - Generate p, q, such that $q \mid p-1$
  - $g \in Z_p^*$ such that $g^q = 1 \bmod p$
  - $x \in Z_{q-1}^*$, $y = g^x \bmod p$
  - secret key: x, public keys: (p,q,g,y)
- Signature generation
  - generate *random* $k \in Z_{q-1}^*$
  - compute $r = (g^{k^{-1}} \bmod p) \bmod q$
  - compute $s = k(m+xr) \bmod q$
  - signature: (r, s)
- Verification
  - compute $v = (g^{ms^{-1}} y^{rs^{-1}} \bmod p) \bmod q$
  - compare (r, v)

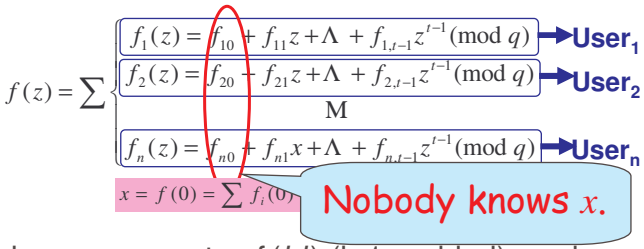
48

Threshold DSA (TS-DSA)

- We extended threshold DSS scheme proposed by Gennaro, et al. [Eurocrypt'96]
  - group size (n) can be increased.
  - threshold (t) can be changed.
- No dealer involved
- VSS holds

TS-DSA: Setup

- Self-initialization by founding members
  - uses Joint Secret Sharing (JSS), T.P.Pedersen [Eurocrypt'91]

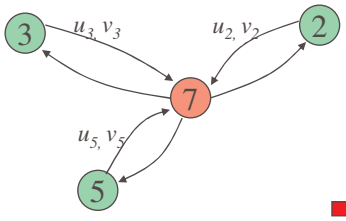


- Each user computes $f_i(id_j)$ (j=1..n, j != i), and sends it to others.
- Each user computes his own secret share

$$ss_i = f(id_i) = \sum_{j=1}^{n} f_j(id_i)$$

TS-DSA: Signature Generation

- DSA signature: (r, s)



O(t²) comm. O(t) exp.

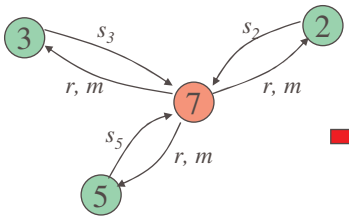
$$u_j = k_j a_j \bmod q \qquad v_j = g^{a_j} \bmod p$$

$$U = \sum_{j=1}^{t} u_j l_j(0) \bmod q \quad [= ka \bmod q]$$

$$V = \prod_{j=1}^{t} (v_j)^{l_j(0)} \bmod p \quad [= g^a \bmod p]$$

extra exp.

$$r = (V^{U^{-1}} \bmod p) \bmod q \quad [= g^{k^{-1}} \bmod p \bmod q]$$



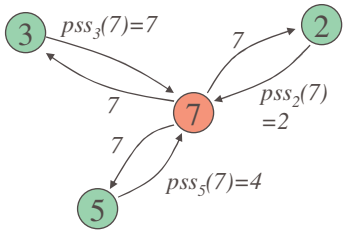
(2t-1)-degree

$$s_j = k_j(m + x_j r) \bmod q$$

$$s = \sum_{j=1}^{t} s_j l_j(0) \bmod q$$

TS-DSA: VSS success

- Example: $f(x) = 7 + 2x + 5x^2 \pmod{11}$, $g=9$, $q=11$, $p=23$
- Witnesses: $w_0=9^7=4$, $w_1=9^2=12$, $w_2=9^5=8 \pmod{23}$



$$pss_i(j) = ss_i l_i(j) \bmod p$$

$$ss_7 = pss_2(7) + pss_3(7) + pss_5(7) = 2$$

$$g^{ss_7} = 9^2 = 12$$

$$(w_0)(w_1)^7 (w_2)^{7^2} = 12$$

$$g^{ss_7 \pmod{11}} = \prod_{i=0}^{2} (w_i)^{7^i} \pmod{23}$$

TS-DSA: Summary

- Pros:
  - VSS guaranteed
  - Key generation fully distributed
- Cons:
  - Robust only if fewer than $\lfloor (t+1)/2 \rfloor$ malicious users
  - Extra $O(t^2)$ communications between signers for random secret *k*

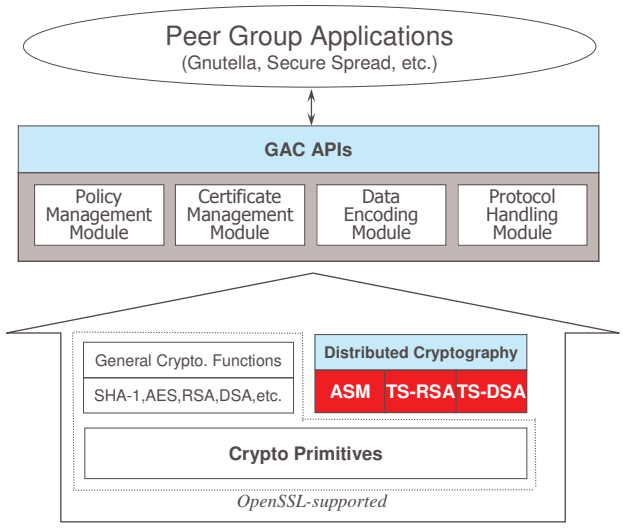
53

Outline

- Introduction
- Motivation
- Definition
- Work done
  - Analysis of existing schemes
  - **Design and implementation**
  - Performance evaluation
- Future work

54

System Design - "Bouncer" toolkit



55

Dynamic Threshold Update

Function **GAC_Dynamic_Threshold_Update();**

Input parameters:

```
    X509* GChart,          \* group charter *\
    int N_old,             \* old group size *\
    int N_cur,             \* current group size *\
    int T_cur              \* current threshold *\
```

Body:

```
    int diff;
    int offset;
    int Win;               \* Window buffer size *\
    int T_new              \* new threshold *\

     T_new=T_cur;
     Win=WIN_TIMES*GChart.threshold.fixed;
     diff = N_cur - N_old;
     if (diff >= Win) {
         offset = ⌊diff / Win⌋;
         N_old = N_old + (offset*Win);
         T_new = ⌊(GChart.threshold.dynamic /
                   100) * N_old⌋;
         if (T_new > T_cur)
              T_cur=T_new;
     }
     return T_new;
```

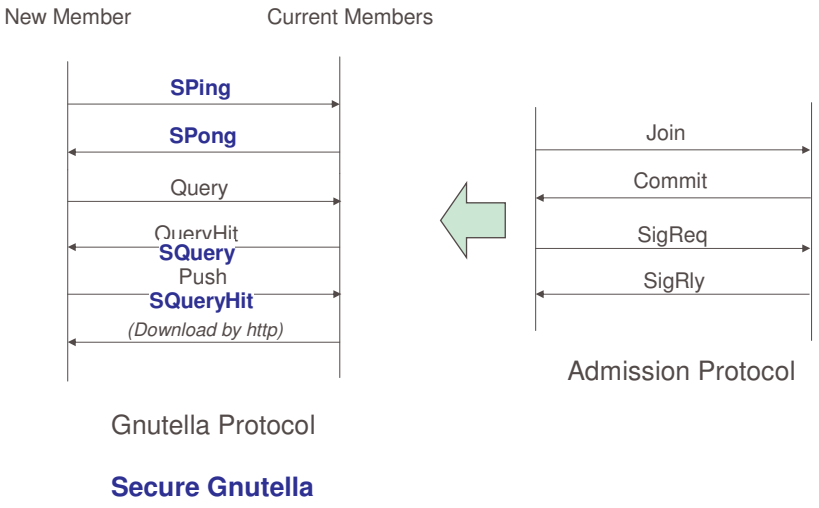
56

GAC APIs

- Plain RSA APIs
  - GAC_PACKET *PS_Join_Reqest();
  - GAC_PACKET *PS_Join_Commit();
  - GAC_PACKET *PS_GMC_Request(); /* optional */
  - GAC_PACKET *PS_GMC_Reply(); /* optional */
- TS-RSA APIs
  - GAC_PACKET *TSS_Join_Request();
  - GAC_PACKET *TSS_Join_Commit();
  - GAC_PACKET *TSS_Sign_Request();
  - GAC_PACKET *TSS_Part_Sign();
  - GAC_PACKET *TSS_GMC_Request(); /* optional */
  - GAC_PACKET *TSS_GMC_Reply(); /* optional */
- TS-DSA APIs
  - GAC_PACKET *TSD_Join_Request();
  - GAC_PACKET *TSD_Join_Commit();
  - GAC_PACKET *TSD_Chal_Req();
  - GAC_PACKET *TSD_Chal_Rly();
  - GAC_PACKET *TSD_Rnd_Req();
  - GAC_PACKET *TSD_Rnd_Rly();
  - GAC_PACKET *TSD_Sign_Request();
  - GAC_PACKET *TSD_Part_Sign();
  - GAC_PACKET *TSD_GMC_Request(); /* optional */
  - GAC_PACKET *TSD_GMC_Reply(); /* optional */
- ASM APIs
  - GAC_PACKET *ASM_Join_Request();
  - GAC_PACKET *ASM_Join_Commit();
  - GAC_PACKET *ASM_Sign_Request();
  - GAC_PACKET *ASM_Part_Sign();
  - GAC_PACKET *ASM_GMC_Request(); /* optional */
  - GAC_PACKET *ASM_GMC_Reply(); /* optional */

57

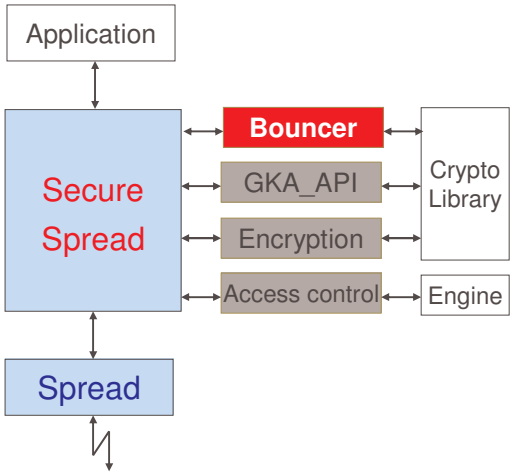
Integration with Gnutella



58

Integration with Secure Spread

- Spread: A wide area reliable group communication system
- Secure Spread: Integrates security services with Spread
  - Supports only static access control
    - daemon level ACL's
    - *flush* mechanism
  - *No notion of secure, dynamic, distributed admission.*
- Modified Spread APIs
  - SP_GAC_Join(); /* new */
  - SP_receive(); /* modified */



59

Outline

- Introduction
- Motivation
- Definition
- Work done
  - Analysis of existing schemes
  - Design and implementation
  - **Performance evaluation**
- Future work

60

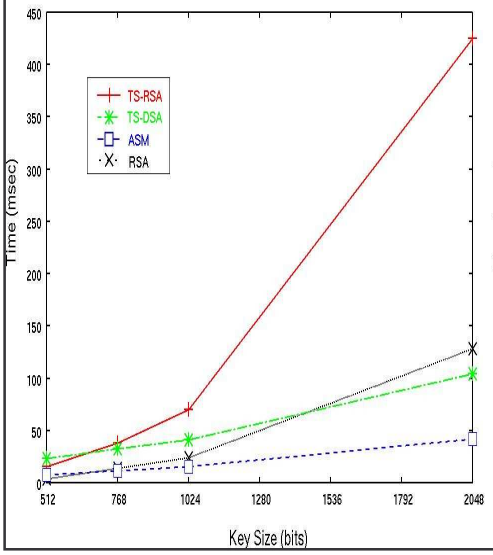
Experimental Setup

- **Gnutella Experiment:**
  - Integrated decentralized protocol with Gnut-0.4.21
  - Tested on a high-speed LAN
- **Secure Spread Experiment:**
  - Integrated centralized protocol with Secure SPREAD-2.1.0
  - Spread daemons on 10 machines at Johns Hopkins Univ.
  - A client at UCI
- Measurements
  - Fixed threshold
  - Dynamic threshold

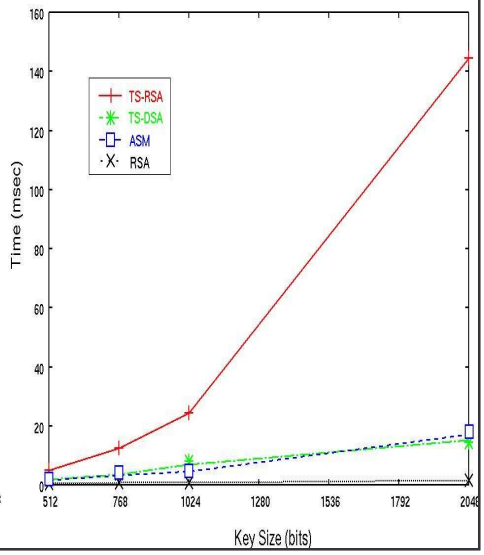
61

Computation Cost

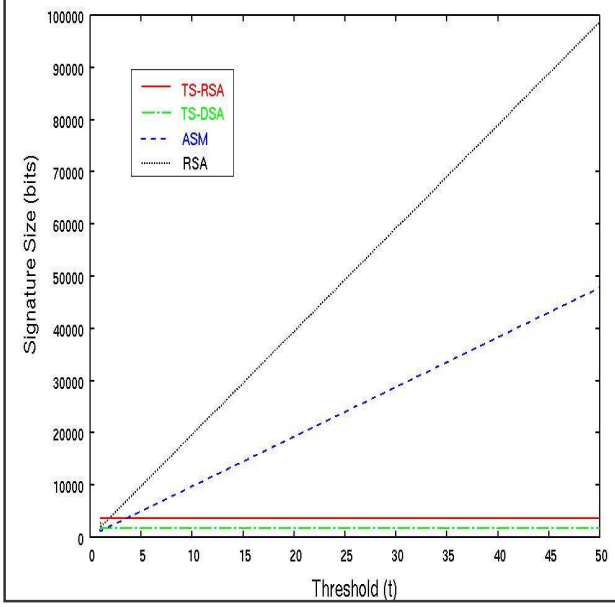
Signature generation



Signature verification



Signature Size

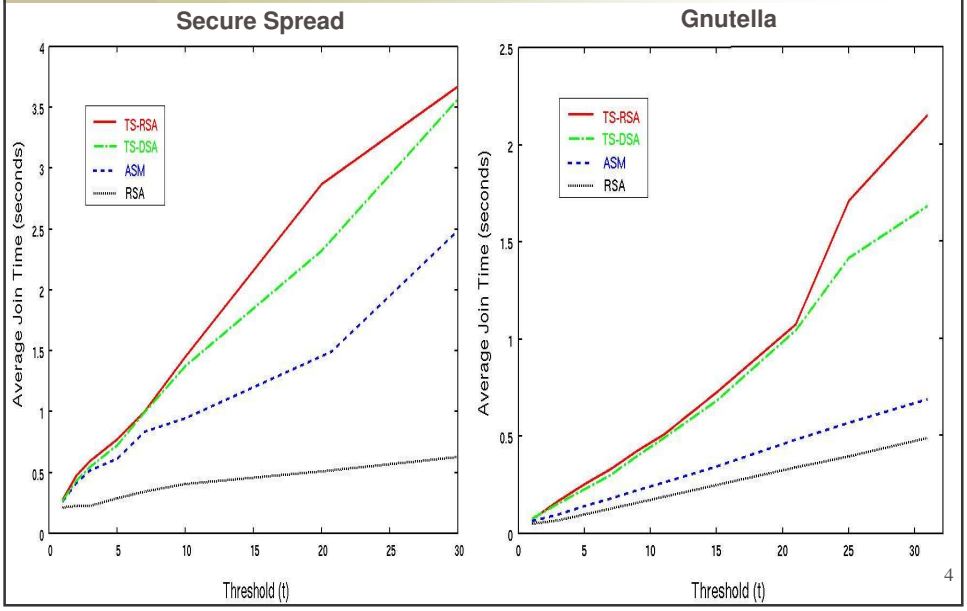


	Signature length
RSA	*t* * (|K| + |id|)
ASM	|q| + |E| + *t* * |id|
TS-RSA	|K|
TS-DSA	2*|q|

t: threshold
K: private key
id: signer's id
q: modulus q
E: challenge (hash size)

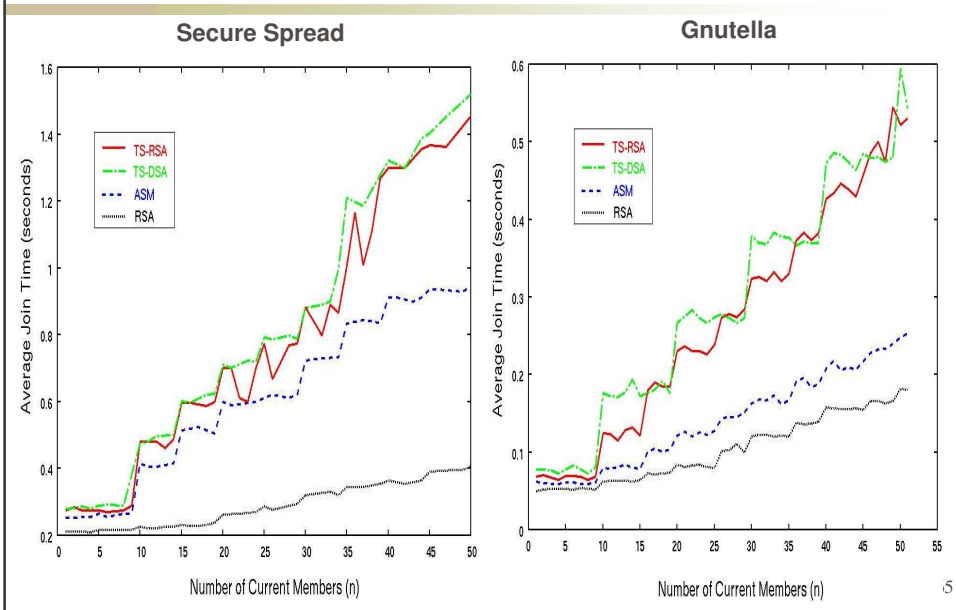
63

Fixed Threshold Experiments



4

Dynamic Threshold Experiments



65

Summary

- Designed several P2P admission control mechanisms
- Assessed practicality of distributed cryptography for dynamic peer groups.
  - Threshold signatures are currently ***NOT PRACTICAL*** in MANETs and sensor networks
  - Reasonable for Internet-based P2P systems that operate in (at least partially) synchronous mode
- Difficult to identify one scheme best-suited for all peer group admission scenarios.
- If admission is difficult, distributed membership revocation is even harder!

66

Future Work

- TS-RSA
  - Efficient RSA distributed modulus generation
  - VSS in Dynamic setting
- TS-DSA
  - Better communication efficiency?
- Aggregated Signatures?

- Bilinear mapping based approach
  - Security proof of the protocol
  - External membership proof process

- Other, more “systems” approaches?
- Revocation?

67

End



68