Authentication and Key Establishment in Computer and Communication Networks: No Silver Bullet

Manish Mehta
Ph.D. Candidate, M.S., CISSP

September 15, 2006

http://research.manishmehta.com/

My Technical Background (1993-2006)

- Diploma in Computer Technology, 1996, India.
- Bachelors of Engineering in Computer Engineering 1999, Univ. of Mumbai, India.
- Masters of Science in Computer Science, 2002, Univ. of Missouri – Kansas City, USA.
- Certified Information Systems Security Professional (CISSP)
- Ph.D. Candidate in Computer Science, 2006 Univ. of Missouri – Kansas City, USA.
  - Research focus: Security Protocols, Digital Signature

Ph.D. Committee

- Dr. Lein Harn (Ph.D. Advisor)
- Dr. Yugyung Lee (Ph.D. Co-Advisor)
- Dr. Deep Medhi
- Dr. Xiaojun Shen
- Dr. Khosrow Sohraby

Outline

- The Problem and Current Research
- Motivation
- Dissertation Research – A Big Picture
- Selected Results from the Research
- Publications
- Ph.D. *beyond* Dissertation Research

The Problem

- Enable (two or more) entities to communicate securely.

Who is "entity"?

- Person
- Computer
- PDA
- Software module
- Sensor
- Router
- Access point
- Base station
- Telephone/Cell
- Refrigerator
- Printer
- ……..Anyone

Intuitive Solutions and Challenges

- Pure Symmetric-key solutions
  - Key distribution ($O(n^2)$)
  - It is not advisable to use the shared key over and over for multiple sessions.
- Pure Asymmetric-key solutions
  - The operations are computationally too heavy to use for all messages during the communication.

Better Approach is AKE

- Authentication and Key Establishment (AKE) provides a mechanism to establish a key, called *session key*, between/among participants for each session in an authenticated manner. This session key can be used for securing further communication.

Goals of AKE

- Mutual belief in each other and the key.
  - User-oriented Goals
    - Participants have fresh assurance that the other participant is a valid peer entity.
  - Key-oriented Goals
    - Key is fresh.
    - Key is known only to the intended parties.

Current State of Research in AKE

- The problem has been around for many years
- Symmetric key systems have been used for several years (e.g. Kerberos)
- Public key systems make more sense in modern distributed communications environment (IPSec, SSL)
  - Off-line trusted third party
  - Non-repudiation service
- New environments like Wireless networks and Sensor networks bring renewed interest in AKE.

Motivation

- Research in AKE field has been "spotty".
  No known attempt to consider the problem as a whole or at a higher level.
- New attacks on existing solutions
- Security assumptions can be proven wrong in future. (MD5 – one of the recent ones !!)
- New set of constraints in new environments (Wireless and Sensor Networks)

Dissertation Research -A Big Picture

1. Identifying Constraints in AKE Design
   - NINE major constraints identified
2. Categorizing Environments and Proposing Solutions
   - THREE categories of environments
   - SEVEN solutions in total

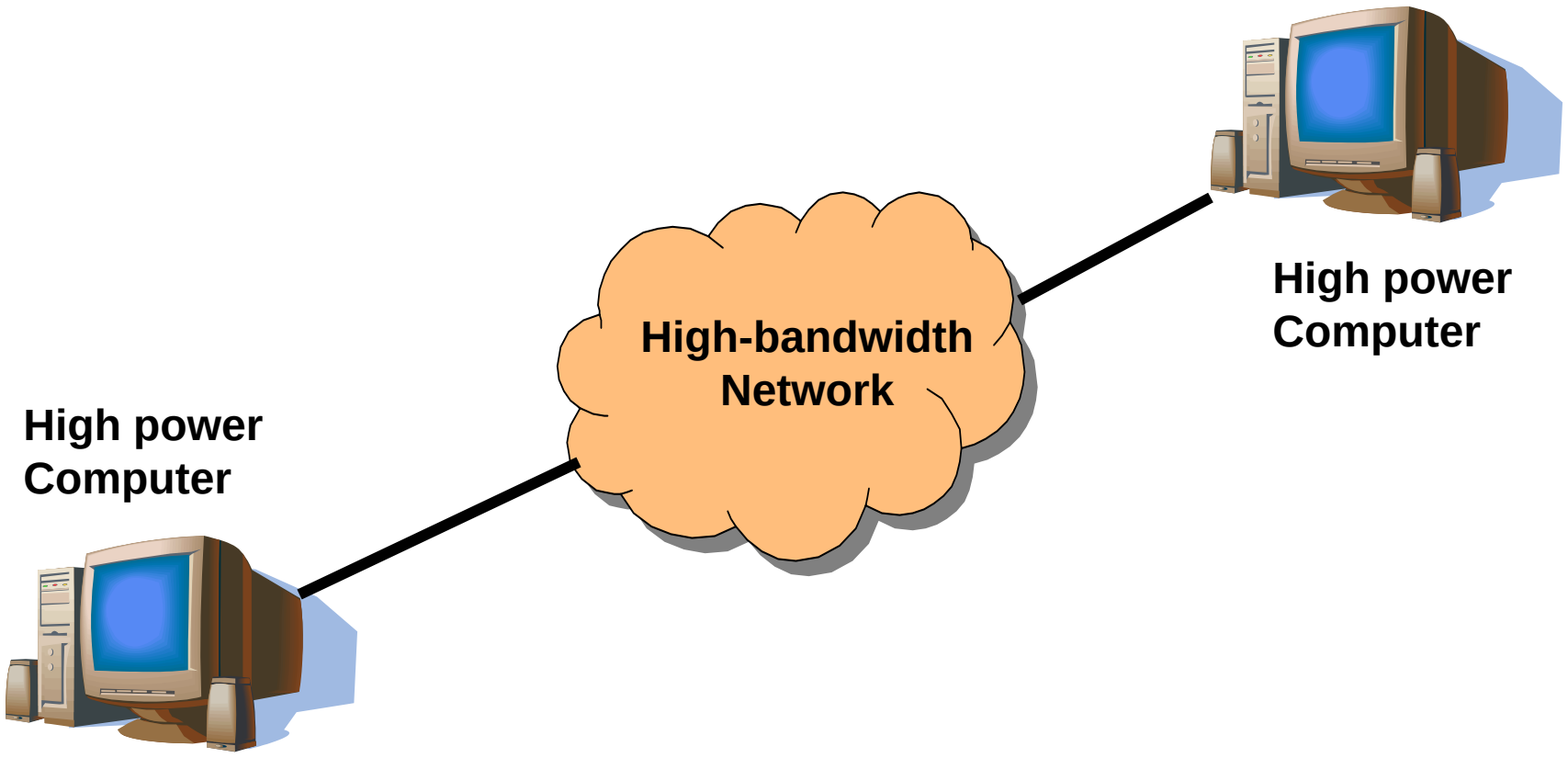
Identified Constraints

1. Cryptographic Algorithm Strength
2. Number of Cryptographic Assumptions
3. Anonymity/Privacy
4. Computational Capacity
5. Communication Bandwidth
6. Communication Media
7. Energy Consumption
8. Storage Capacity
9. Storage Security

Categorization – No Silver Bullet !

Categories	Example environment	Constraints	Solutions
1. High-end to High-end	Computer to Computer	1, 2, 3	3 solutions proposed
2. High-end to Low-end	Computer to PDA or Base station to Mobile	1, 2, 3, 4, 5, 6, 8	1 solution proposed
3. Low-end to Low-end	Sensor to Sensor	4, 5, 6, 7, 8, 9	3 solutions proposed

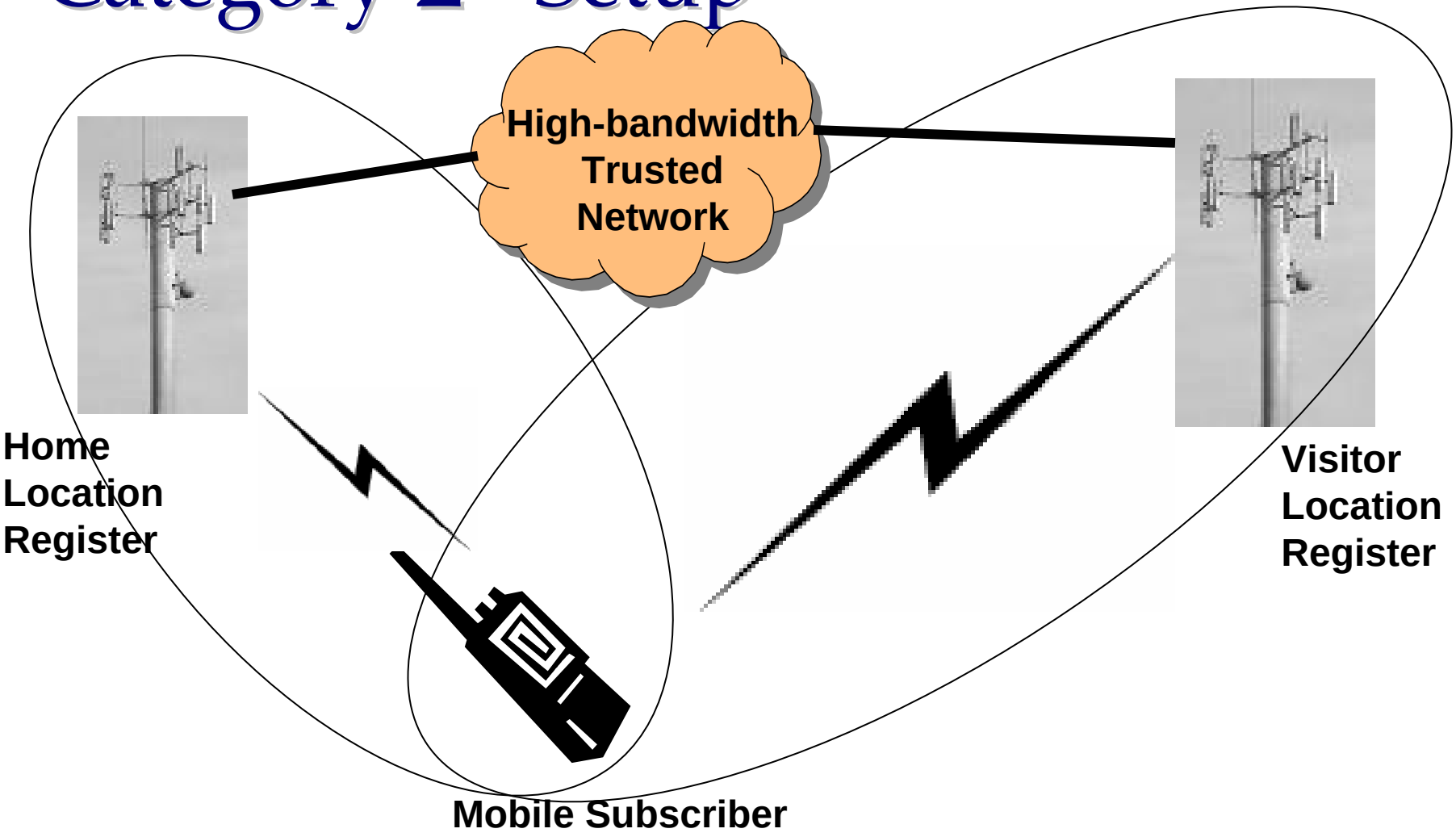
Category 1 - Setup



Category 1 (High-end to High-end)

- Relatively matured research area
- Yet, new attacks are introduced regularly on existing solutions
- Many combinations of cryptographic algorithms are used.
- New services demand more
  - IPSec, SSL, SSH, Web services

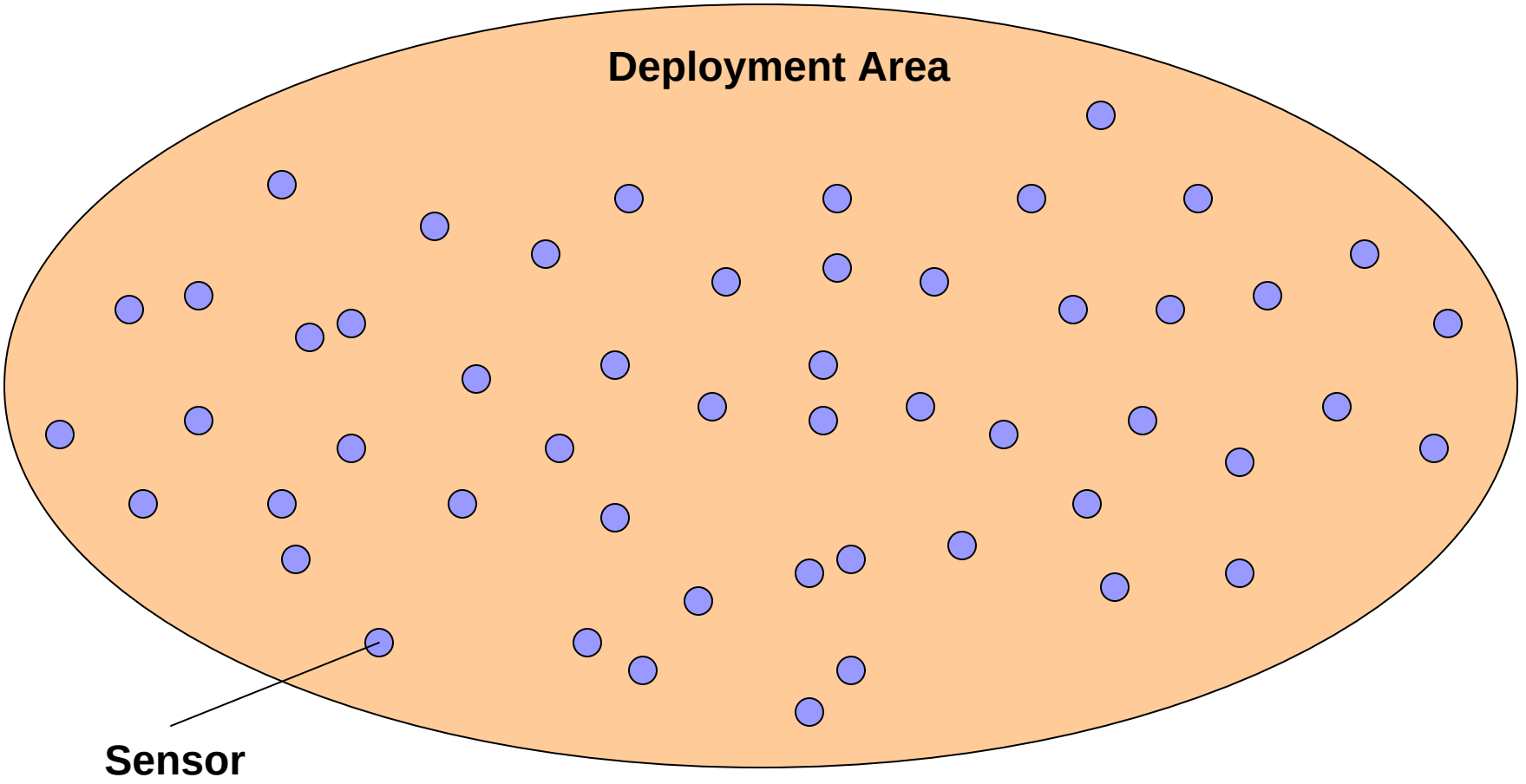
Category 2 - Setup



Category 2 (High-end to Low-end)

- Relatively new research area
  - Wireless Communication (e.g. WLAN, GSM)
- Many standards and frameworks in only few years (IEEE 802.1x, EAP, and more)
- Constraints demand affordable and secure solution
- Public-key solutions for category 1 are considered difficult and/or expensive to implement if not impractical

Category 3 - Setup



Category 3 (Low-end to Low-end)

- Relatively very young research area
  - Ad hoc and sensor networks
- Limited number of solutions proposed
- Stringent constraints
- Lack of post-deployment configuration information
- Public-key solutions are considered impractical at this time
- Sensor devices are not considered tamper-proof (your private key is no more private)

Solutions

- Category 1 (High-end to High-end)
  1. **Integrating DH into DSA (DH+DSA)**
  2. **AKE using Single Assumption**
  3. **Non-public key Digital Certificate**
- Category 2 (High-end to Low-end)
  1. **Wireless AKE protocol using only RSA**
- Category 3 (Low-end to Low-end)
  1. **Modeling Pairwise Key Establishment**
  2. **RINK-RKP**
  3. **Source-Routing-based Pairwise Key Establishment**

Overview of DH + DSA

- Integrating well-known Diffie-Hellman and Digital Signature Algorithm
- Three AKE protocols proposed for different applications
- Provides resistance to the *known-key attack*, the *unknown key-share attack*, and the *key replay attack*
- *Published in IEEE Communication Letters, 2004*

Overview of DH + DSA

Three round protocol with key confirmation

Step	User A	User B
1	Select random integer v $m_A = g^v \bmod p$ $\xrightarrow{m_A}$	
2		Select random integer w $K_{BA} = (y_A)^w \bmod p$ $K_{AB} = (m_A)^{x_B} \bmod p$ $m_B = g^w \bmod p$ $r_B = m_B \bmod q$ $s_B = ((w)^{-1}(H(m_B\|\|K_{BA}\|\|K_{AB}) + x_B r_B)) \bmod q$ $\xleftarrow{(m_B, s_B)}$
3	$K_{AB} = (y_B)^v \bmod p$ $K_{BA} = (m_B)^{x_A} \bmod p$ $r_B = m_B \bmod q$ Verify DSA signature (r_B, s_B) of message m_B $r_A = m_A \bmod q$ $s_A = ((v)^{-1}(H(m_A\|\|K_{AB}\|\|K_{BA}) + x_A r_A)) \bmod q$ $\xrightarrow{s_A}$	
4		$r_A = m_A \bmod q$ Verify DSA signature (r_A, s_A) of message m_A

Overview of AKE on Single Assumption

- Proposed an argument for building AKE protocols with minimum number of cryptographic assumptions
- Presented three one-assumption protocols based on DL, RSA, and ECC
- Presented security analysis for all three protocols
- *Published in IEE Proceedings of Communication Letters, 2005*

Overview of AKE on Single Assumption

DL-based one-assumption AKE protocol

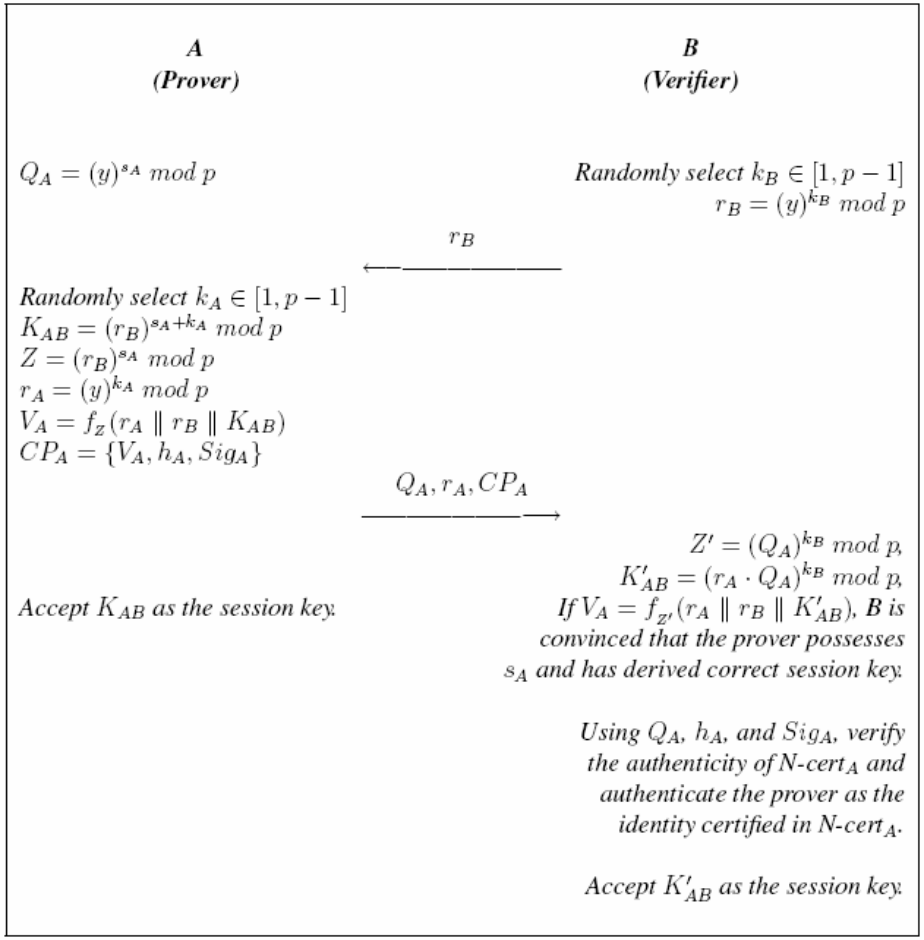
Run	Step	Computation	
1	1	C_A^1	A selects k_A
	2		A computes r_A
1	3	C_B^1	B selects k_B
	4		B computes r_B
	5		B computes $k_{AB} = (r_A)^{K_B} mod\, p$
	6		B computes $s_B = k_{AB}^{-1}(K_B - r_B k_B)\ mod\, q$
2	7	C_A^2	A verifies R_B by checking $cert(R_B)$
	8		A computes $k'_{AB} = (R_B)^{k_A} mod\, p$
	9		A verifies s_B and k'_{AB} by checking $R_B \stackrel{?}{=} (r_B)^{r_B}\ (\alpha)^{(s_B k'_{AB})}\ mod\, p$
	10		A computes $k_{BA} = (r_B)^{K_A}\ mod\, p$
	11		A computes $s_A = k_{BA}^{-1}(K_A - r_A k_A)\ mod\, q$
3	12	C_B^3	B verifies R_A by checking $cert(R_A)$
	13		B computes $k'_{BA} = (R_A)^{k_B}\ mod\, p$
	14		B verifies s_A and k'_{BA} by checking $R_A \stackrel{?}{=} (r_A)^{r_A}\ (\alpha)^{(s_A k'_{BA})}\ mod\, p$

Overview of Non-public key digital certificates

- Innovative asymmetric-key solution for Authentication and Key Establishment
  - No need for generating public key pair for each user
- Solutions can directly convert real-world certificates into digital form (*N-cert*)
- The solution can be easily converted to an identity-based AKE system

Overview of Non-public key digital certificates

One-way AKE using *N-cert*



RINK-RKP

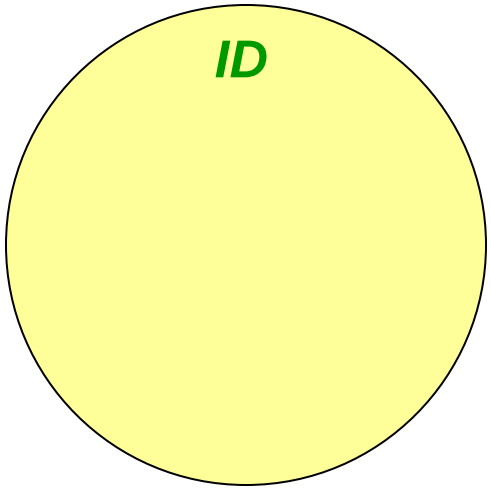
- Goal:
  - Secure communication among sensors
- Scheme:
  - Random Key Predistribution and Shared-key Discovery
- Main Constraints:
  - Computational power, storage, energy, bandwidth
- Main categories of existing solutions:
  - Purely Random Key Predistribution (P-RKP)
  - Structured Key-pool RKP (SK-RKP)
- Proposed solution (*IEEE IPCCC 2005*):
  - More secure
  - Power efficient

Phases in RKP Schemes

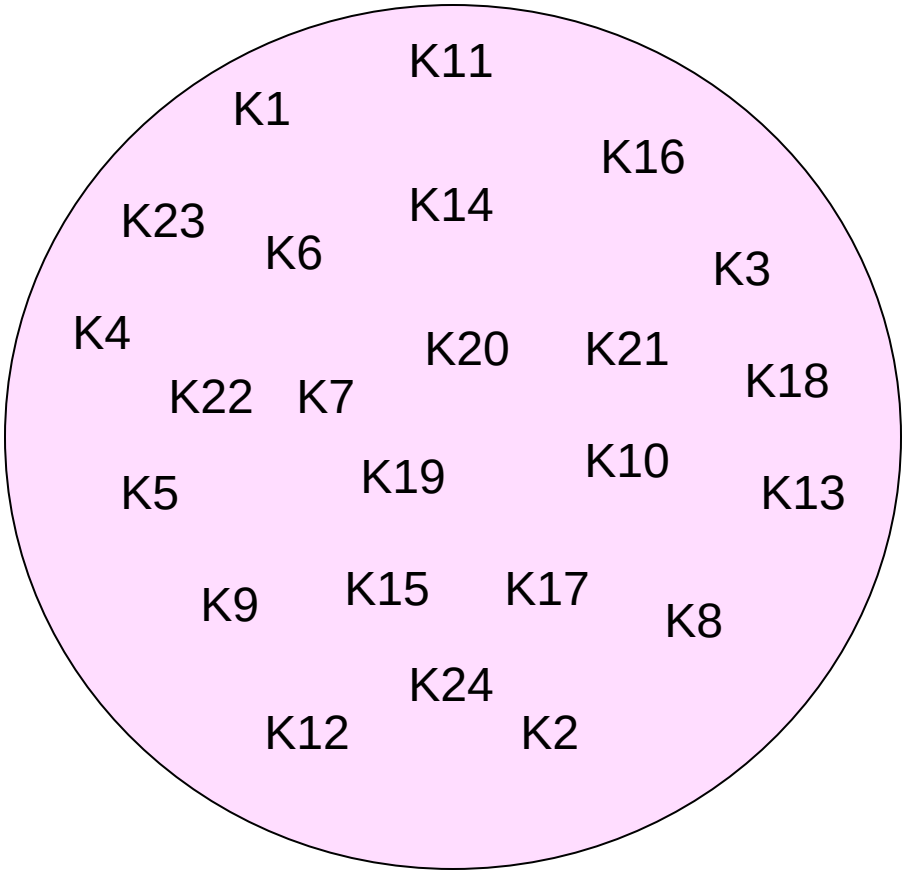
1. Key Predistribution
   - Select and install keys in sensors
2. Sensor Deployment
   - Place the sensors
3. Shared-key Discovery
   - Sensors find common (shared) key(s)
4. Pairwise Key Establishment
   - Those who don't find shared key(s), take help from others.

Existing RKP Schemes (Phase 1)

- P-RKP



Sensor – ***m*** keys

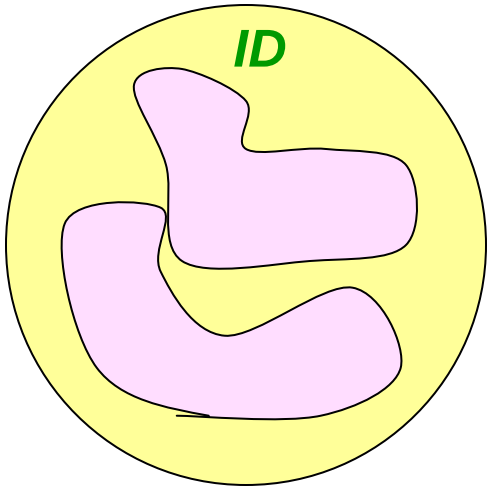


KEY POOL – Size ***n***

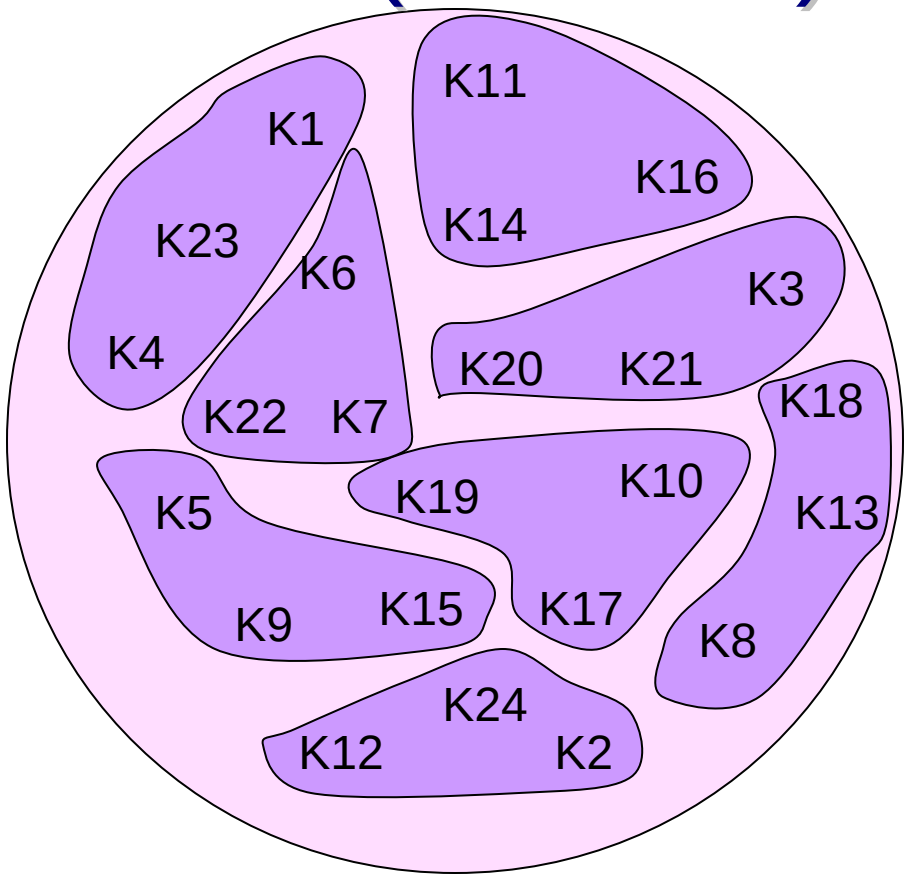
m << n

Existing RKP Schemes (Phase 1)

- SK-RKP



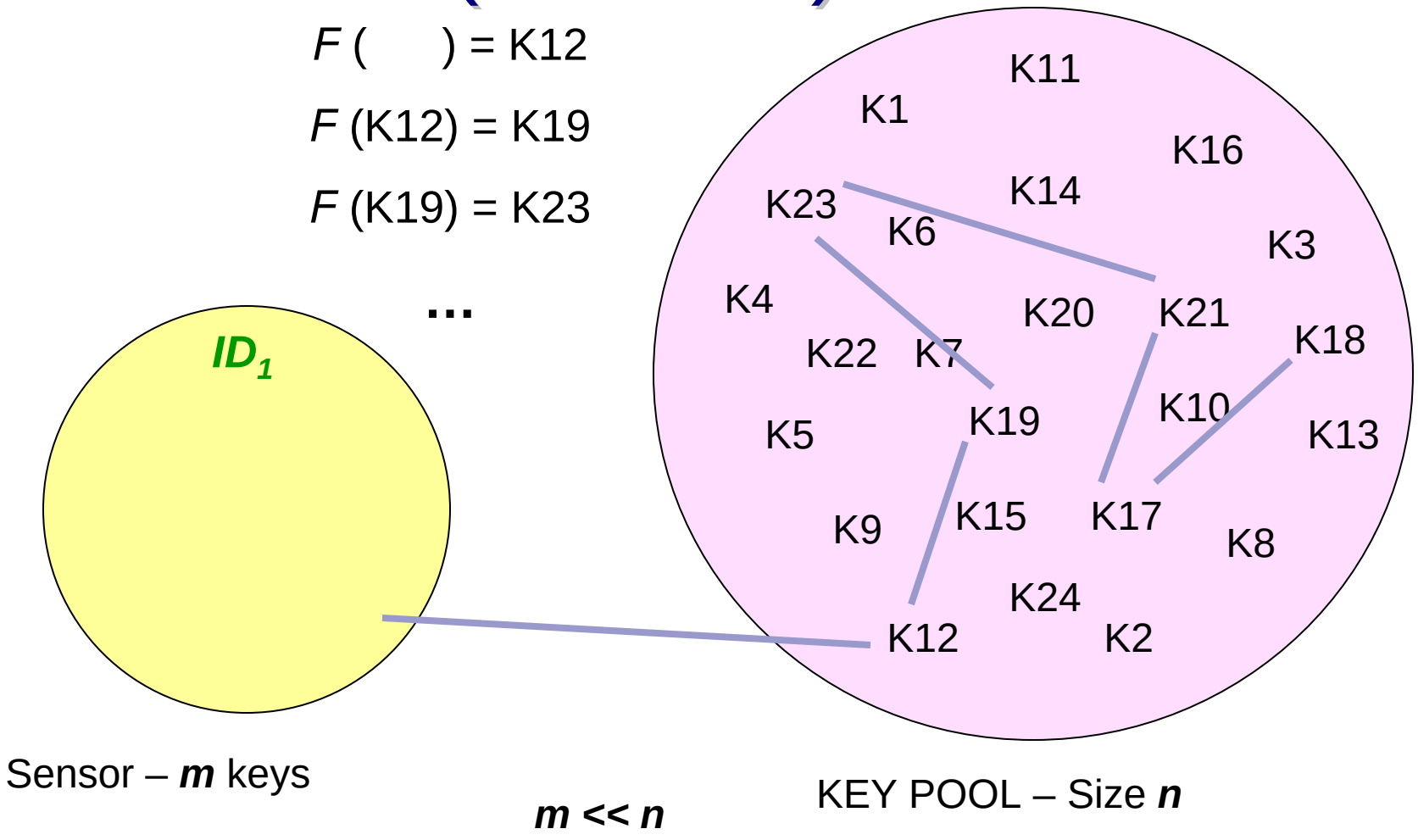
Sensor – ***m*** keys



KEY POOL – Size ***n***

m << n

RINK-RKP (Phase 1)

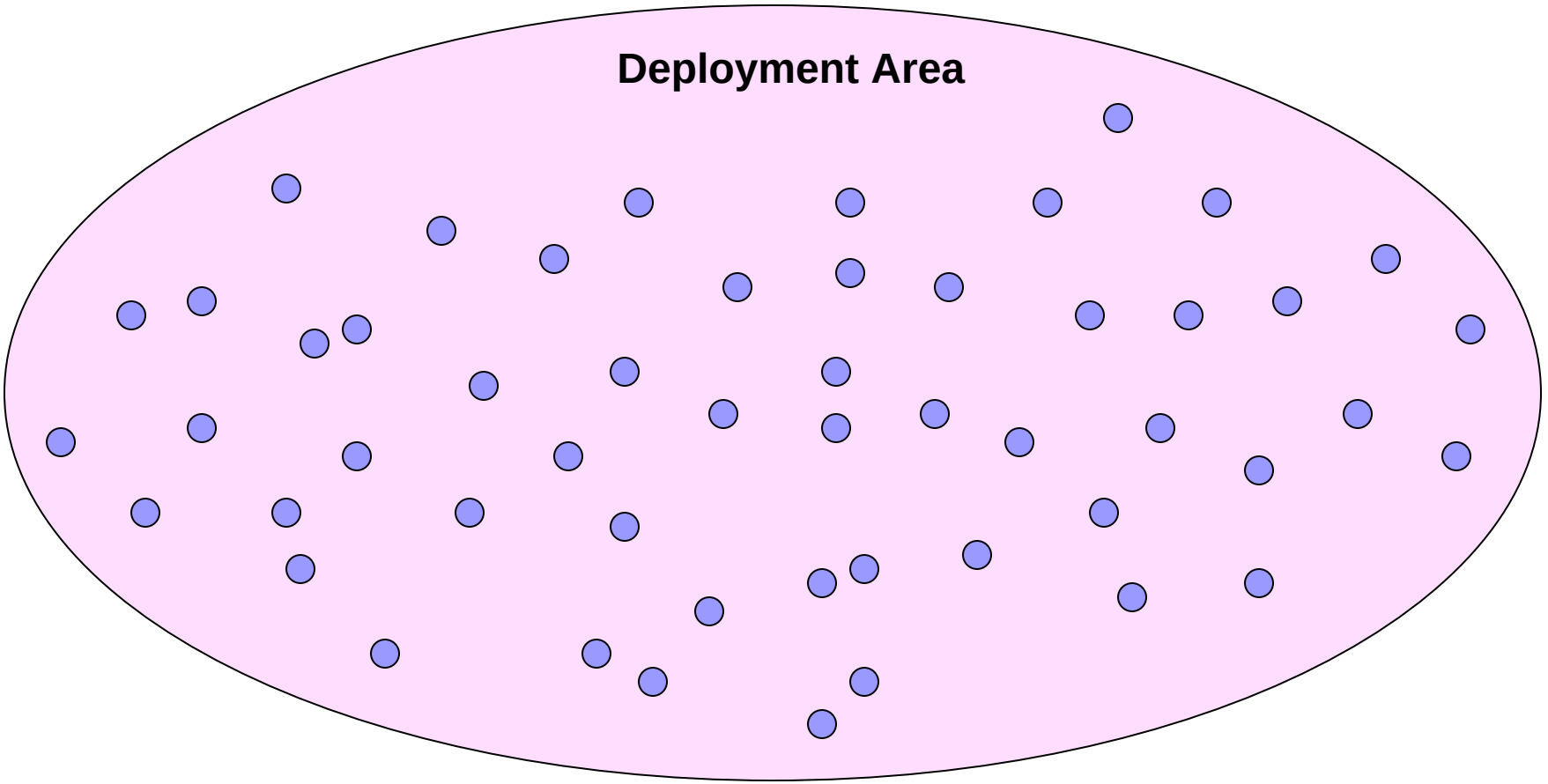


RINK-RKP (Phase 1 - Algorithm)

Key Predistribution Algorithm

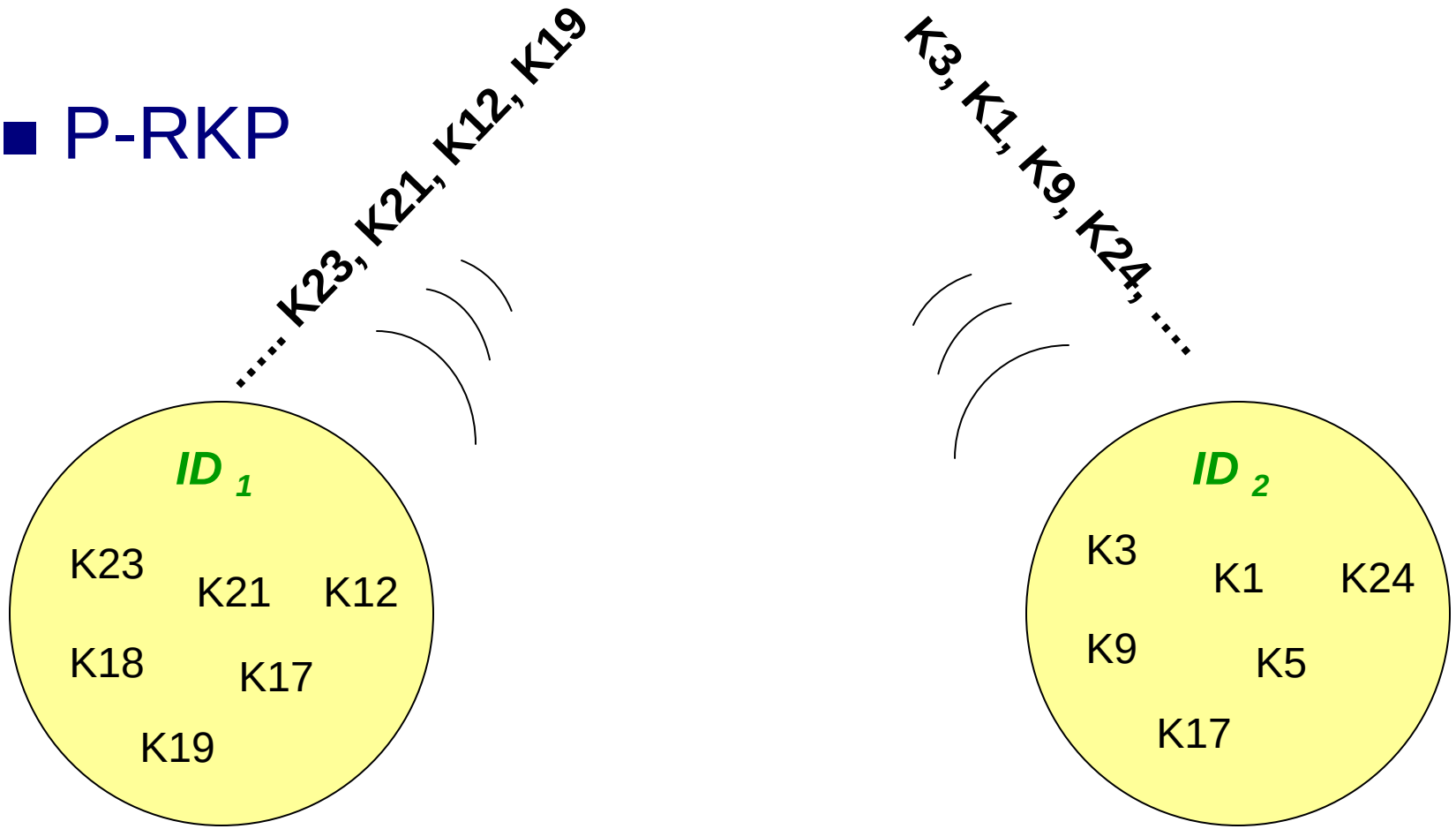
1.	Pick a unique identifier id for a sensor.
	Initialize $Output = \{1\}^z$
2.	for $i = 1$ to m
2.1.	if $(i \leq 2)$
2.1.1.	then $PrevKey = 0$
2.1.2.	else $PrevKey = k_{i-2}^{id}$
2.2.	$M_i^{id} = Output \mid\mid PrevKey \mid\mid id$
2.3.	$Output = H(M_i^{id})$
2.4.	if$(Output \geq 2^z - 1 - (2^z \bmod P))$
2.4.1.	then $i = i - 1$, goto step 2
2.5.	$Key = Output \bmod P$
2.6.	if (Key has already been generated for this sensor)
2.6.1.	then $i = i - 1$, goto step 2
2.7.	$k_i^{id} = Key$
2.8	add k_i^{id} to KC_{id}
3.	Store id, KC_{id}, and corresponding keys (Y's) in the sensor.

Sensor Deployment (Phase 2)



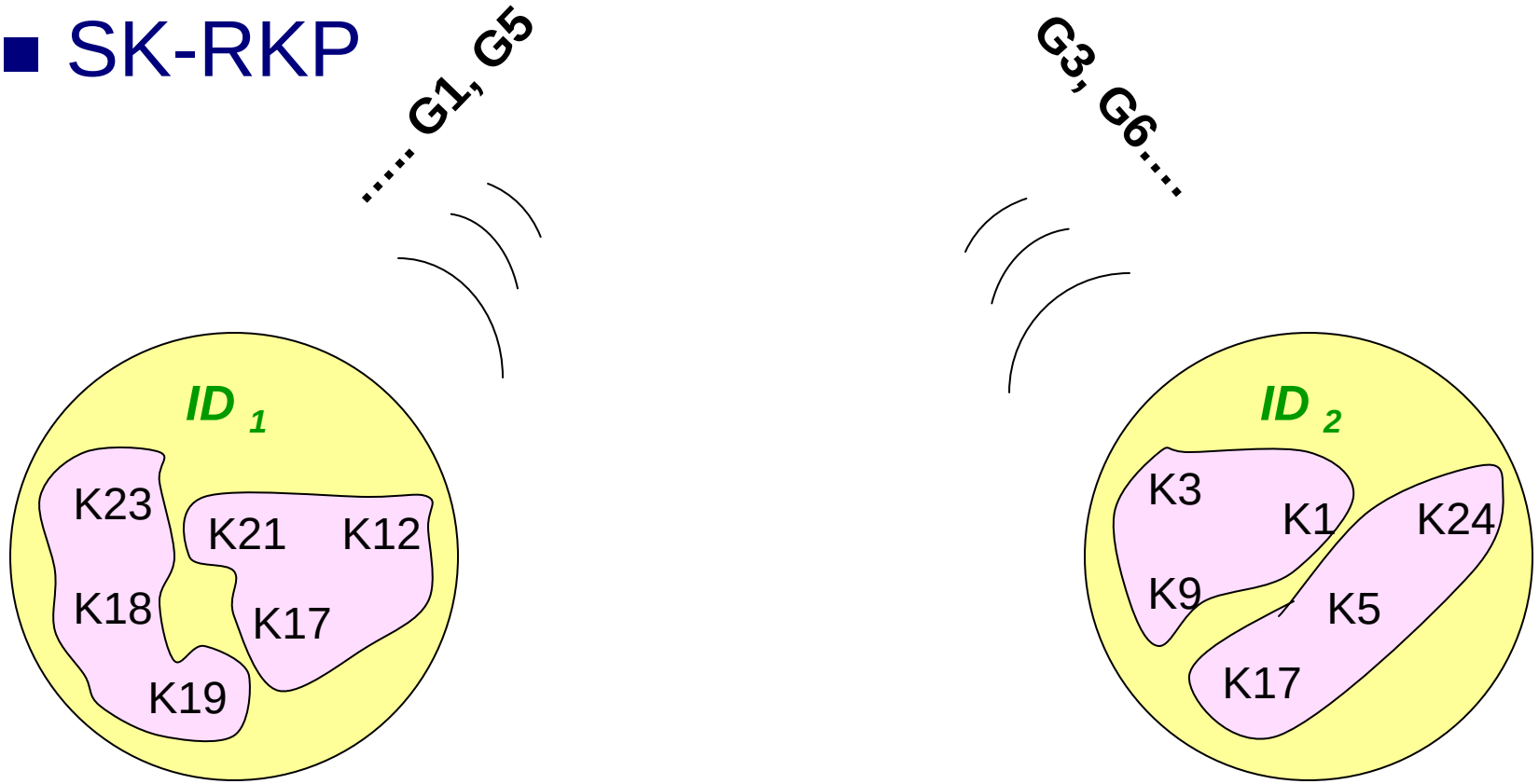
Shared-Key Discovery (Phase 3)

- P-RKP



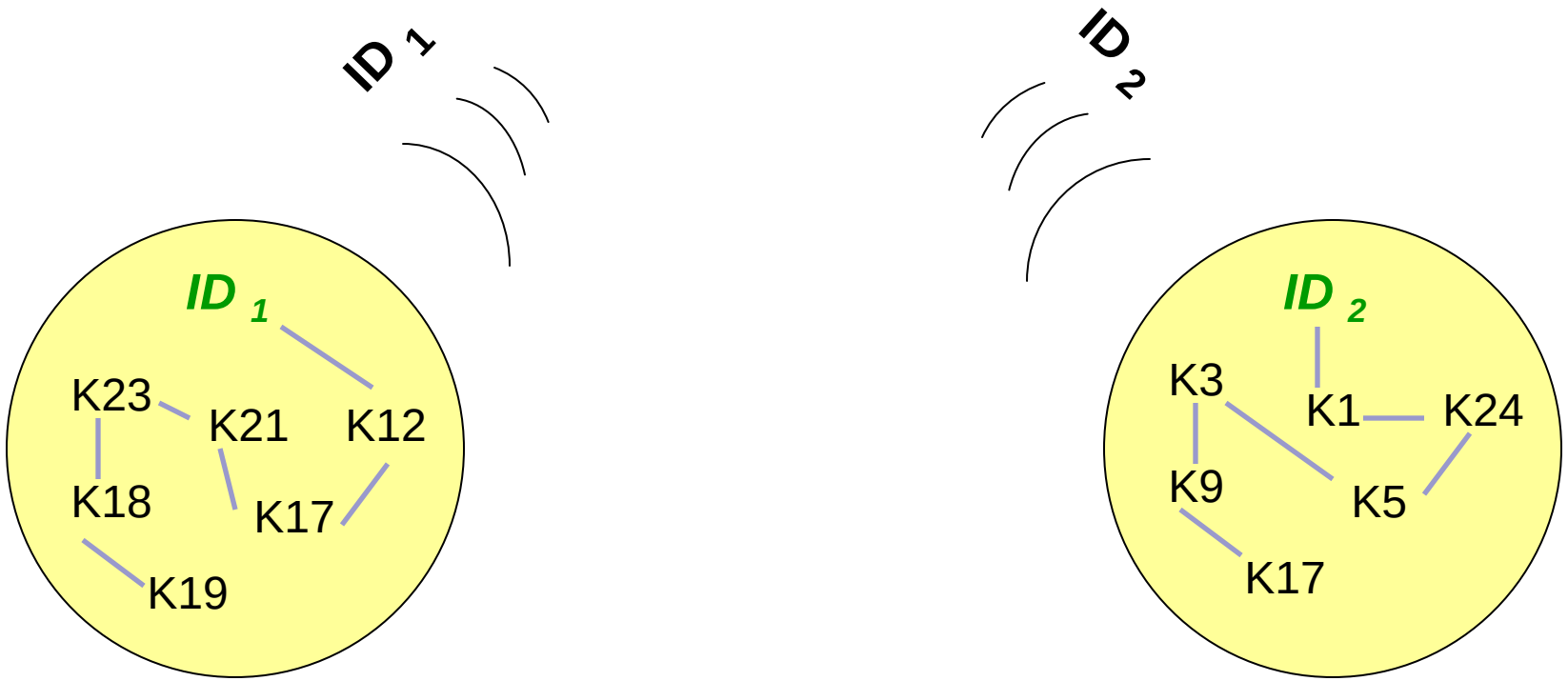
Shared-Key Discovery (Phase 3)

- SK-RKP



Shared-Key Discovery (Phase 3)

- RINK-RKP



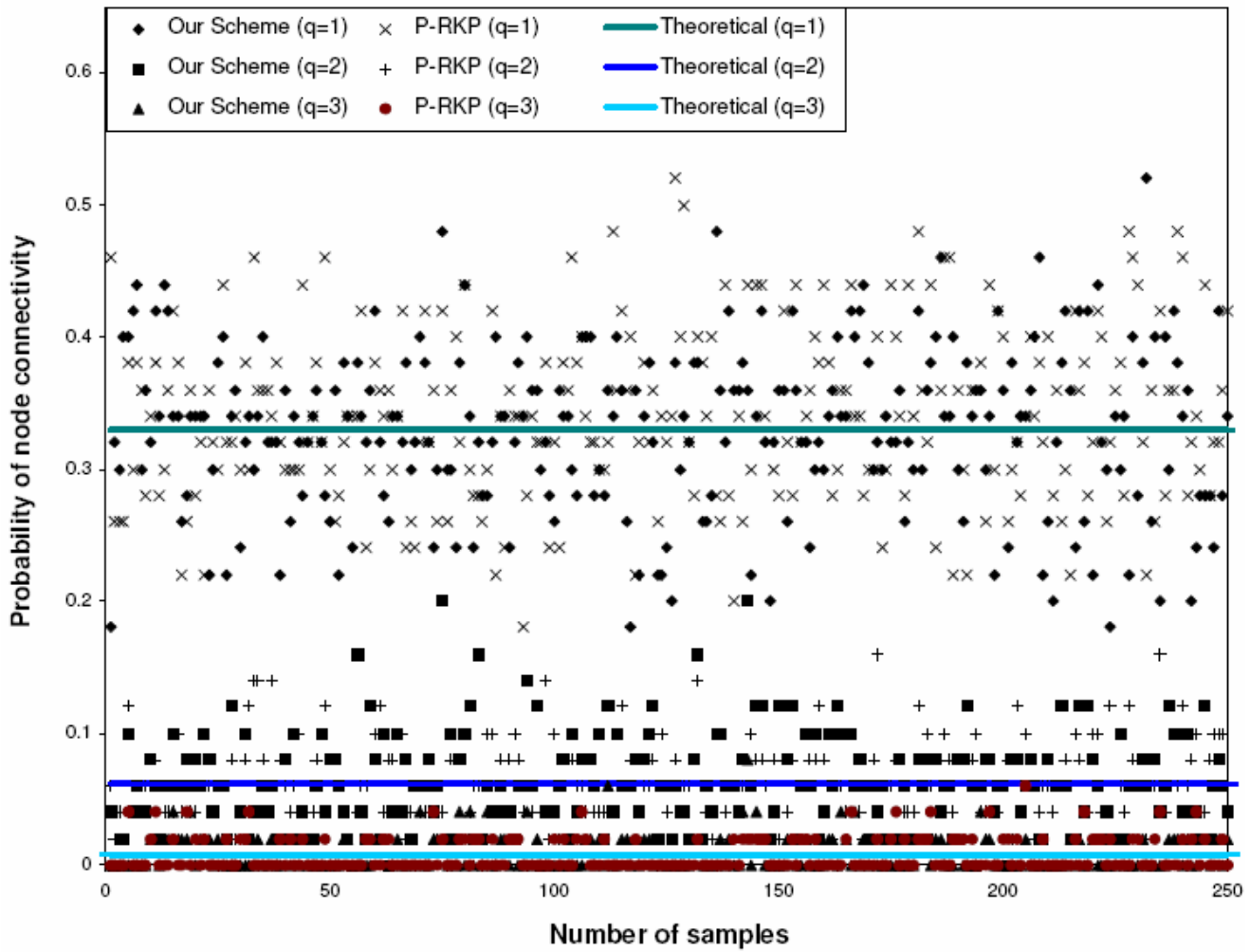
RINK-RKP (Phase 3 - Algorithm)

Shared Key Discovery Algorithm

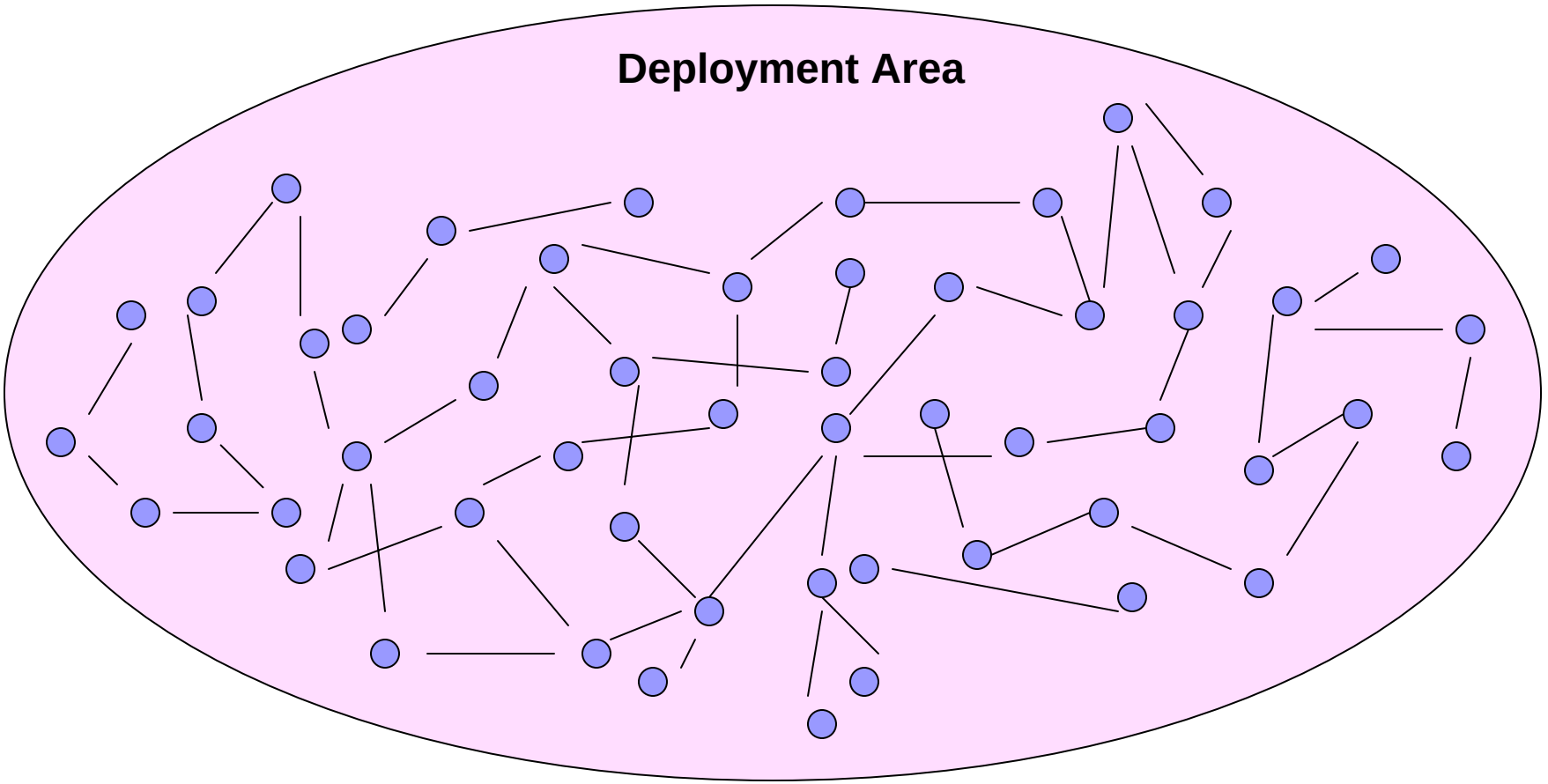
1. Broadcast node identifier, i
2. Receive node ids transmitted by neighbors and generate set of neighbors, W. Generate *key graph* with elements of W as vertices.
3. for $(\forall\ j \in W)$ do
   - 3.1. Generate KC_j using the node identifier, j, as the input to Procedure 1
   - 3.2. Generate Q, a set of key values (Ys) corresponding to common key identifiers in KC_i and KC_j.
   - 3.3. if($|Q| \geq q$) then
     - 3.3.1. Compute the shared-key K_{ij} as $\bigoplus\{Q\}$
     - 3.3.2. Add a link between node i and node j in the *key graph*
4. Stop.

Simulation Results

P = 10000
m = 200



After Phase 3



So what is different in RINK-RKP ?

- Previous approaches do not use node *ID* for key selection, we do !
- That is we define
  RINK = Relation between ID aNd Keys

Security Analysis – Problem 1 - Reasons

- Unattended deployment environment
  - Physically insecure
- No tamper-resistance due to low cost
  - Compromised sensor can reveal the stored keys.

Problem-1 (Capturing Nodes)

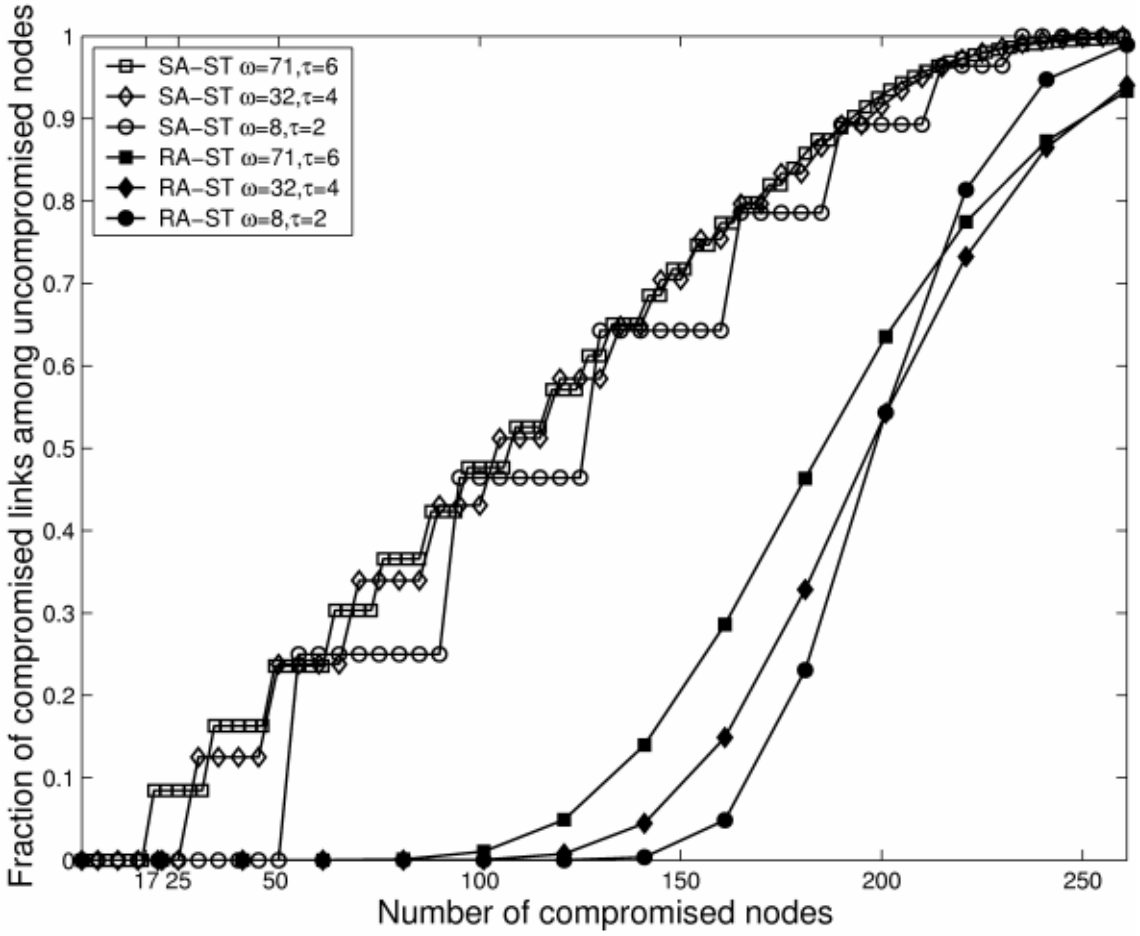
- Random Capture (*naïve approach*)
  - Randomly pick nodes and obtain keys
- Selective Capture (*proposed approach*)
  - Pick sensors that can give you keys that you do not already have

Random vs. Selective Capture

- SHK RKP affected the most
- P RKP and RINK- RKP not affected much

m=100

$p_{connect}$ = 0.432



Security Analysis – Problem 2 - Reasons

- Wireless environment
  - Passive listening is easy
- Unattended deployment environment
  - Fake sensors can be added to the system (proposed attack)

Problem – 2 (Deploying fake sensors)

- Learn keys from captured nodes and fabricate fake nodes
- Fake nodes have enough keys to appear legitimate to other sensors
- Fake nodes can
  - Inject / Absorb sensed data
  - Alter data in specific way

Problem – 2 (Deploying fake sensors)

Probability that a fake node can establish a key with a given neighbor when *x* nodes are captured

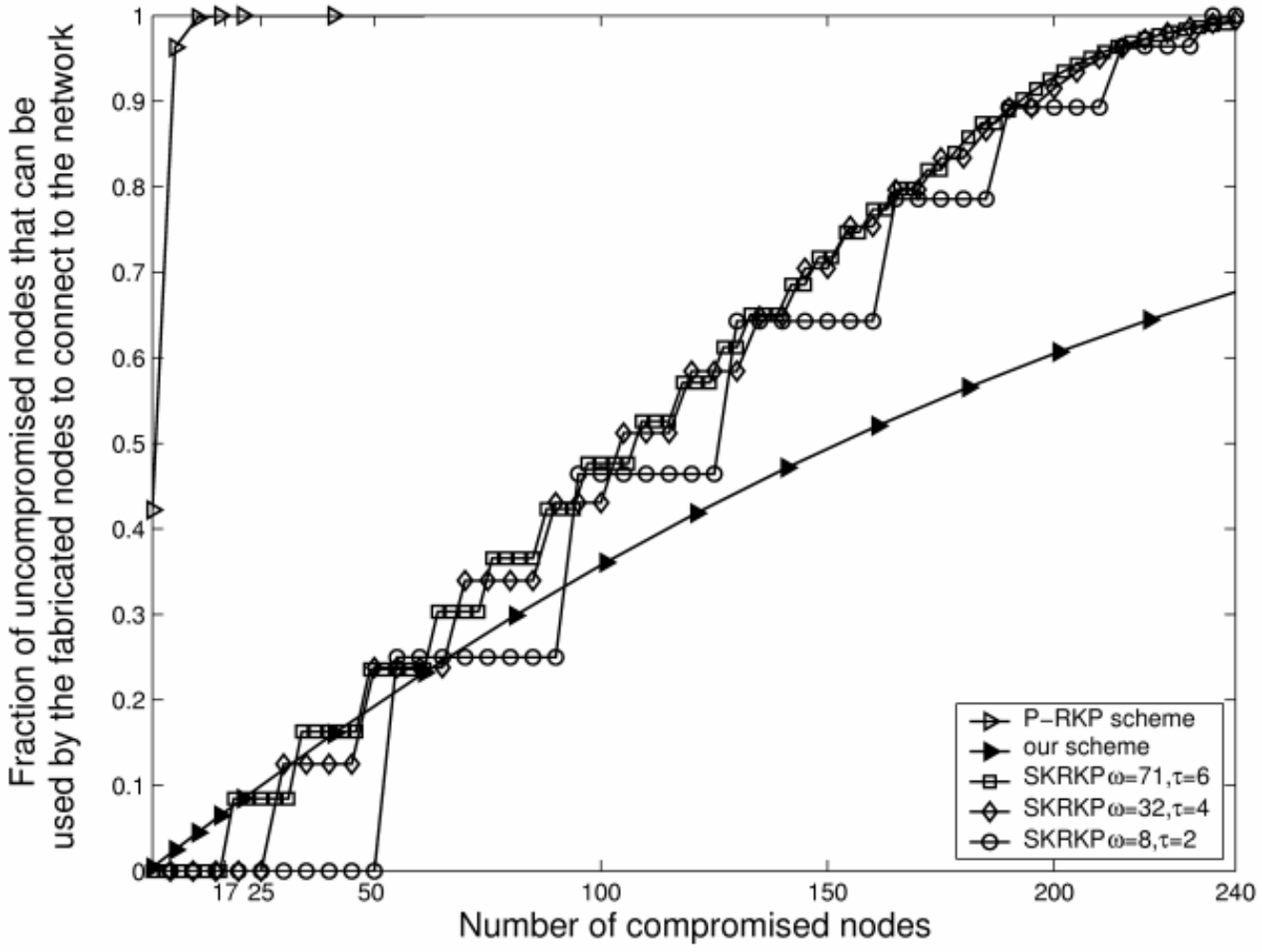
$$p_f(x) = \frac{1}{p_{connect}} \sum_{i=q}^{m} \frac{\binom{P}{m}\binom{m}{i}\binom{P-m}{m-i}}{\binom{P}{m}^2} \left(\frac{C_x}{P}\right)^i$$

Where,

$$C_x = \left[1 - \left(1 - \frac{m}{P}\right)^x\right] \cdot P$$

m = number of keys in each sensor
P = size of the keypool
q = required number of keys to be shared

Damage by fake sensors



Energy Consumption Analysis

Energy consumption in Sensoria's sensor node

Operation	Energy
Transmission at 10 kbps, 10 mW	0.021 mJ/bit
Reception at 10 kbps	0.014 mJ/bit
SHA-1 Hash	0.0000072 mJ/bit
128-bit modular multiplication	0.000115 mJ

Energy Consumption Analysis

Computational overhead

Overhead	P-RKP scheme	SK-RKP		Our scheme
		One seed	λ seeds	
# of comparison operations	$n'(m \log m)$	$n'(\tau \log \tau)$	$n'(\tau \log \tau)$	$n'(m \log m)$
# of hash operations	0	0	0	$n'm$
# of modular multiplications	0	$(2\lambda)p_{connect}$	$(\lambda + 1)p_{connect}$	0

m = number of keys in each sensor
n' = number of neighbors

Generally λ, τ, n' << m

Energy Consumption Analysis

Communication overhead

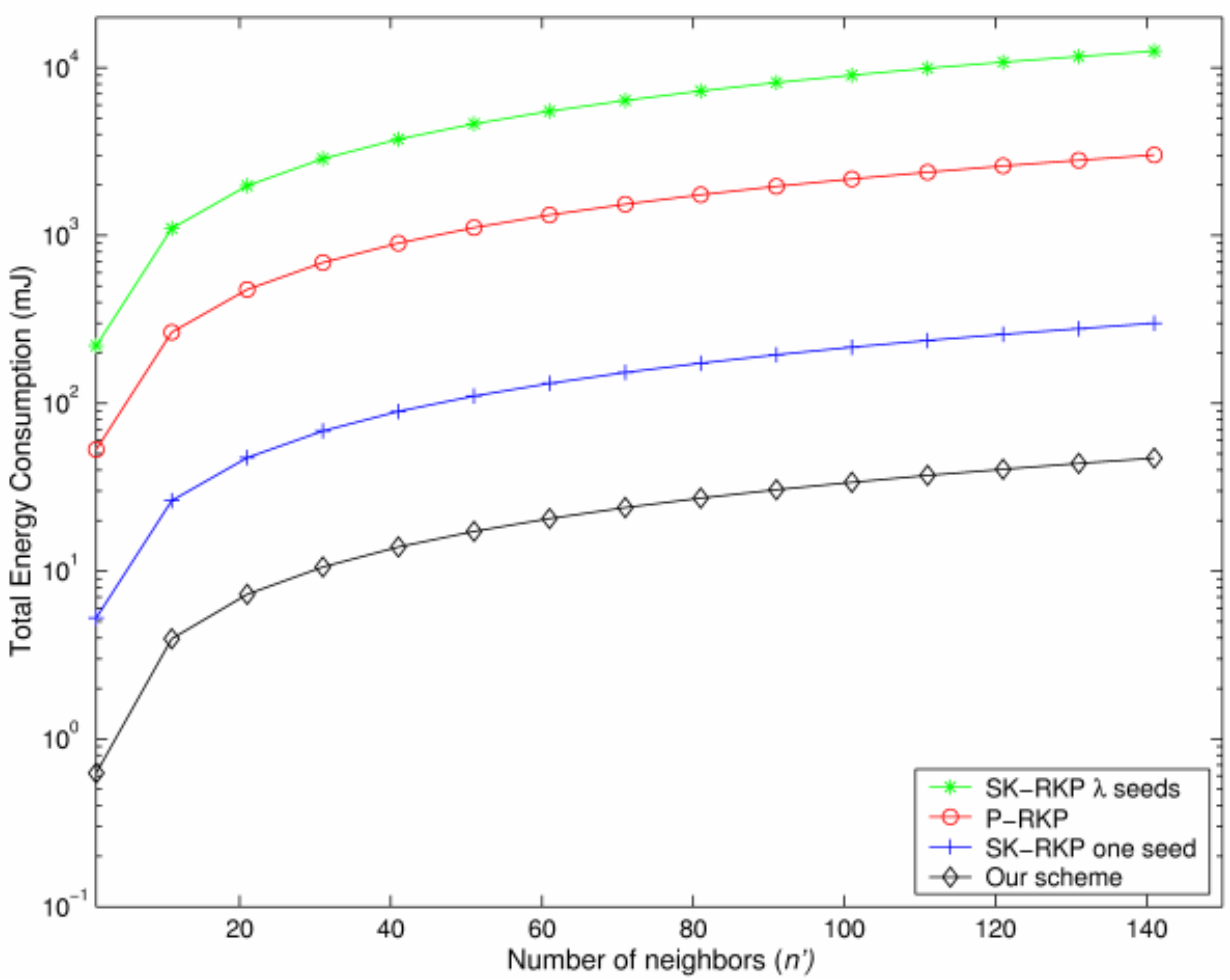
Overhead		P-RKP schemes	SK-RKP scheme		Our scheme
			One seed	λ seeds	
Transmission	Node id (bits)	$\log N$	$\log N$	$\log N$	$\log N$
	Seeds (s bits/seed)	0	s	λs	0
	Key ids (bits)	$m \log P$	$\tau \log \omega$	$\tau \log \omega$	0
	Total (T_{tr}) bits	$\log N + m \log P$	$\log N + s + \tau \log \omega$	$\log N + \lambda s + \tau \log \omega$	$\log N^{\natural}$
Reception	Node ids (bits)	$n' \log N$	$n' \log N$	$n' \log N$	$n' \log N$
	Seeds (s bits/seed)	0	$n's$	$n'\lambda s$	0
	Key ids (bits)	$n'm \log P$	$n'\tau \log \omega$	$n'\tau \log \omega$	0
	Total (T_{rev}) bits	$n'(\log N + m \log P)$	$n'(\log N + s + \tau \log \omega)$	$n'(\log N + \lambda s + \tau \log \omega)$	$n' \log N^{\natural}$

$^{\natural}$: In our scheme, the key ids are generated by the hash function.

N = Total number of sensors
P = Size of the keypool
s = Size of the keys (128 bit)

Energy Efficiency

$P = 23700$
$N = 10000$
$m = 100$
$\omega = 11$
$\lambda = 49$
$\tau = 2$
$s = 128$
$P_{connect} = 0.3455$



Contributions in RINK-RKP solution

- Introduced smart attacks
- Showed that the existing solutions were very vulnerable to the smart attacks
- Proposed a new Predistribution and Shared-key Discovery scheme
- Provided Security and Energy Consumption Analysis

Conclusion

- There is no silver bullet for AKE
- Identifying constraints first makes AKE design process more effective
- Changes in technology can quickly make AKE design obsolete
- Continuous efforts are required to make AKE process more secure and efficient

Future work

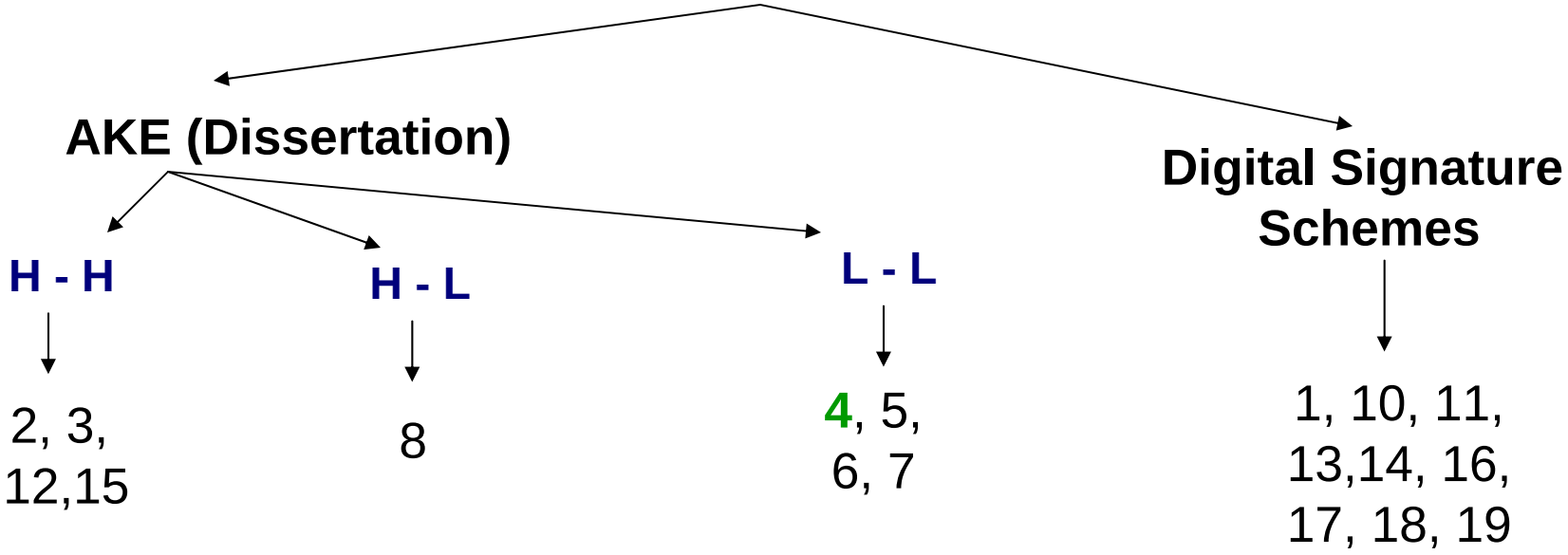
- Towards Perfect Security
  - Mathematical proof
- New Attacks
  - Attackers are getting smarter by the day
- Advances in Technology
  - Computational power

Publications

(Complete list in the handout)

- 4 Refereed Journals (IEE, IEEE, ACM)
- 3 Refereed Conferences (IEEE, ACM)
- 6 Technical Reports (UMKC)
- 1 Book Chapter (IEEE)

Ph.D. Research (Overall)



Statistics

Approx. **38.88 %** of Total Ph.D. research is included in Dissertation.
Approx. **71.42 %** of Dissertation material is published.
Approx. **60 %** of Total Ph.D. research is yet to be published.

Ph.D. beyond Dissertation

- Reviewer for IEEE journals, conferences, magazines, and books chapters.
- Grant writing
  - NSF
  - Univ. of Missouri Research Board
- Earned CISSP Certificate
- One year of industry experience as a Security Consultant

Special Thanks

- Dr. Appie van de Liefvoort
- Dr. Dijiang Huang
- Dr. Balaji Krithikaivasan
- Amit Sinha
- Gaurav Agarwal
- Dr. Jayesh Kumaran
- Chaitanya Garikiparthi
- Royall Hall coffee machine
- Taco bell
- Souper Salad
- …

Questions?

http://research.manishmehta.com