



Distributed Software-based Attestation for Node Compromise Detection in Sensor Networks

Sencun Zhu

Joint work with Yi Yang, Xinran Wang,
and Guohong Cao

Dept. of Computer Science & Engineering
The Pennsylvania State University

Presented at SRDS 2007, Beijing, China

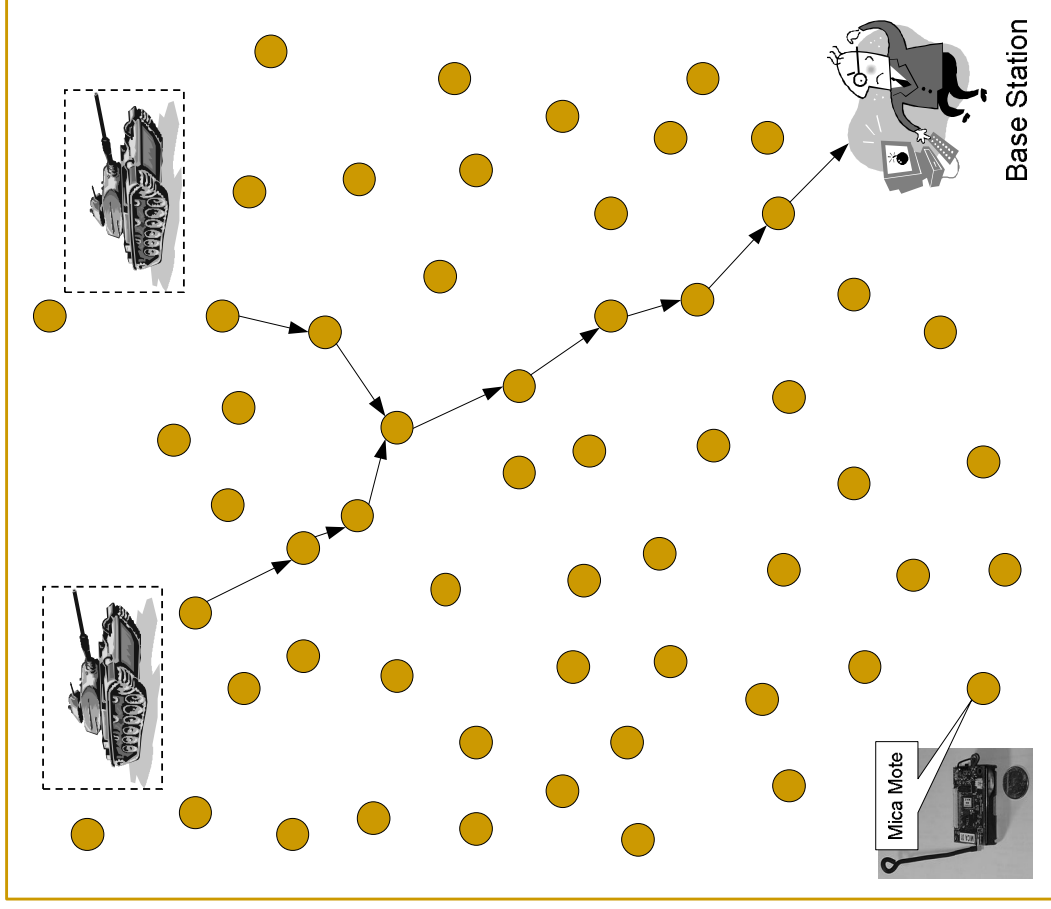


Sensor networks

- **Functions**
 - Sensing
 - In-network processing
 - Ad-hoc communication
 - Multi-hop routing
- **Applications**
 - Real-time traffic monitor
 - Military surveillance
 - Homeland security



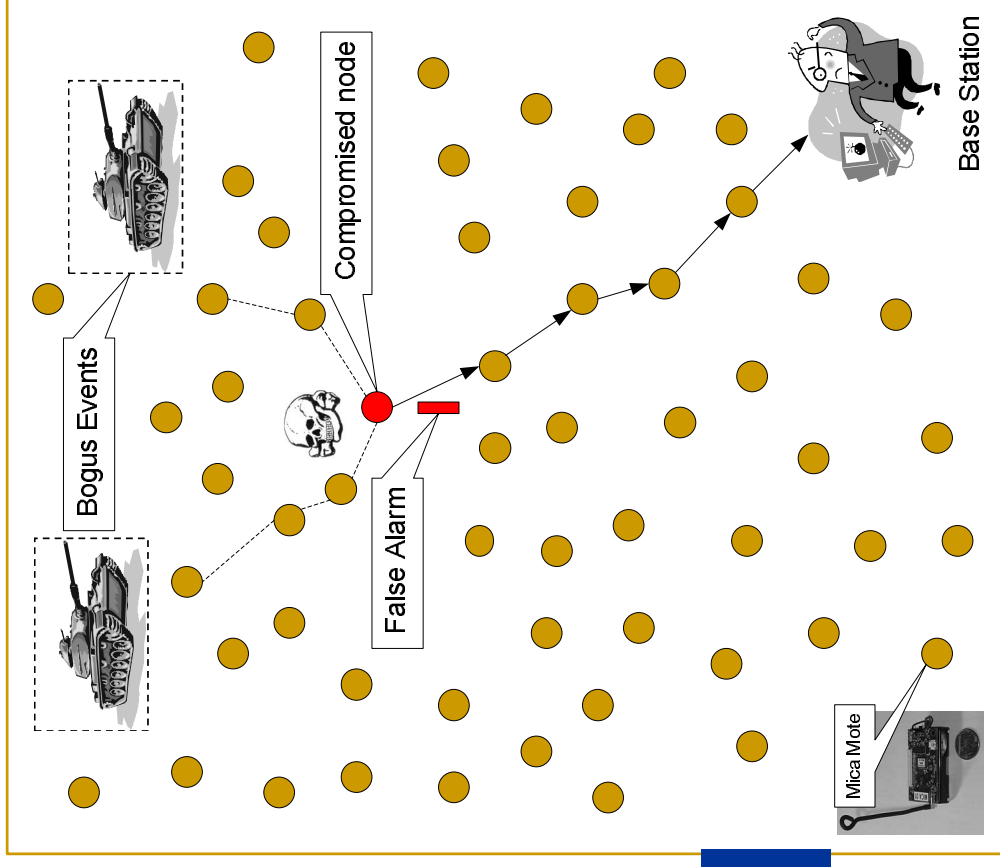
Berkeley Mica Motes





Node Compromises

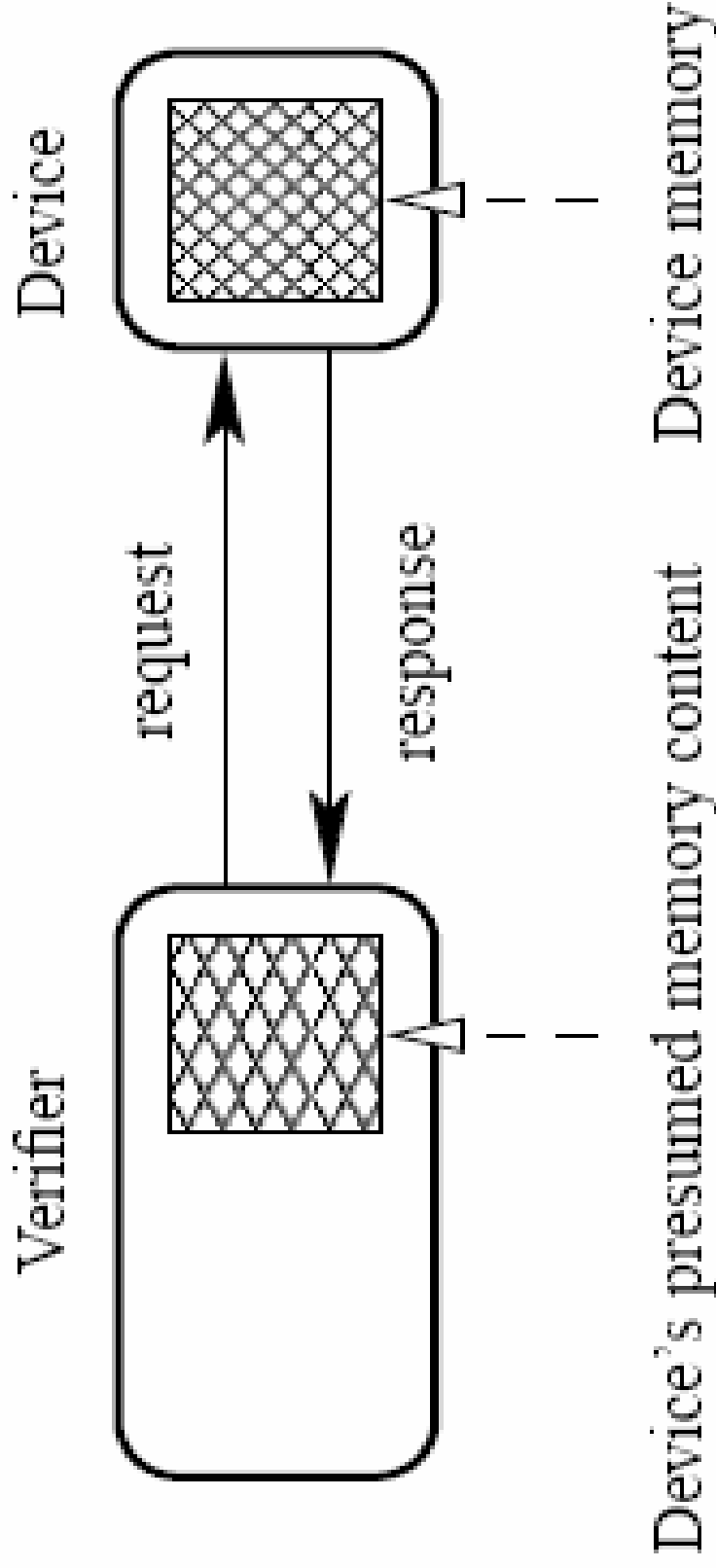
- Data Confidentiality
- Data Integrity
 - Bogus Events
 - False Alarms
 - Wrong Decision
- Denial of Services



Compromise Detection



Previous Work





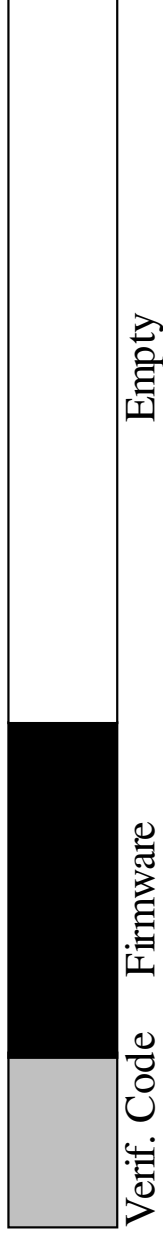
Our Work

- Contributions
 - A pseudo-random noise generation algorithm
 - A block-based memory traversal algorithm
 - Two distributed node attestation protocols
 - Neighbors of a suspicious node collaborate to make a joint attestation and decision
 - No dependence on response time measurement
- Assumptions
 - limited hardware modification (CPU, not memory)
 - non-persistent adversary

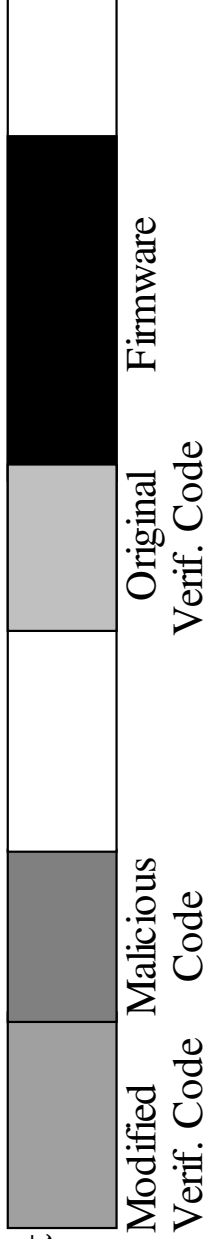


Pseudorandom Noise Generation

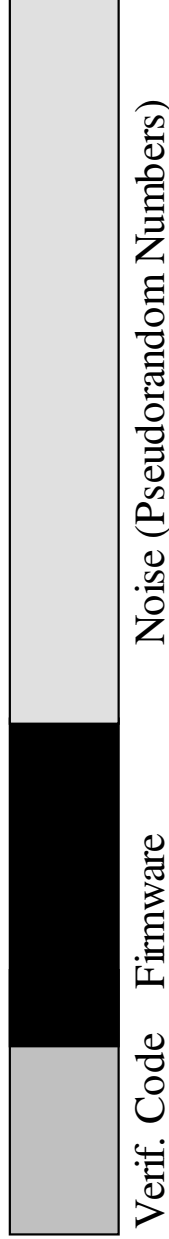
Original Memory Layout
(a)



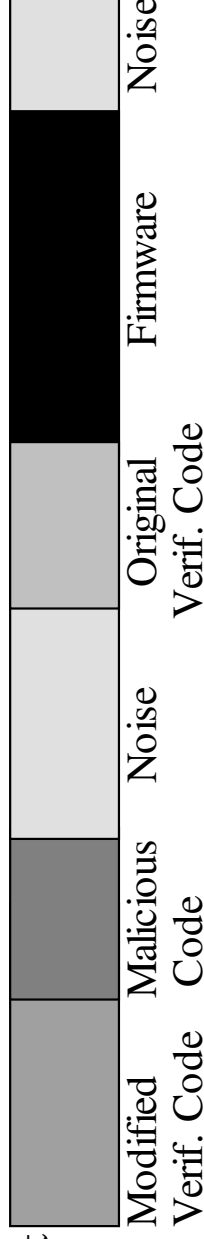
Attacker's Memory Layout
(b)



Expected Memory Layout after Adding Noise
(c)

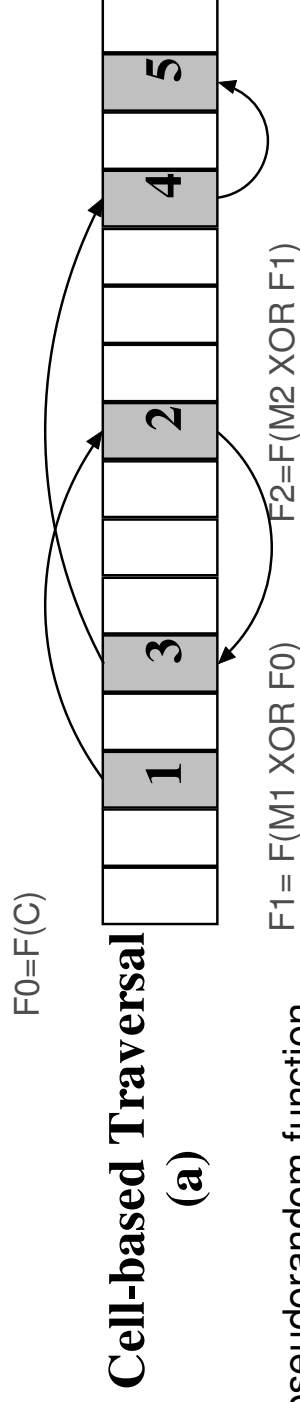


Attacker's Memory Layout after Adding Noise
(d)

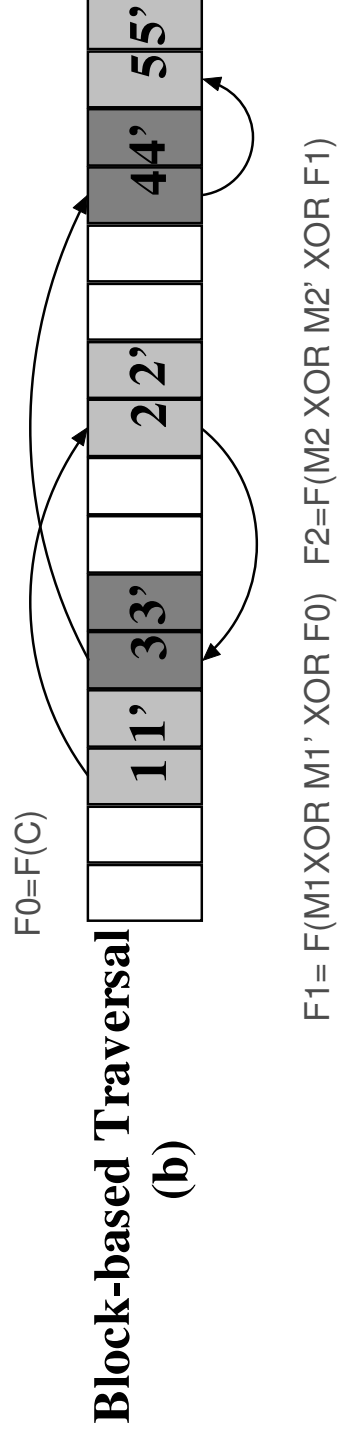




Block-based Pseudorandom Memory Traversal



F: a pseudorandom function
C: user challenge
 M_i : the content in location i





Algorithm 1 Block-based Memory Traversal Algorithm

Input: i_t -number of iterations, j -current byte of checksum, b -block size, m -memory size, A -traversed memory address;

Output: a checksum $C(C_0, \dots, C_7)$;

Procedure:

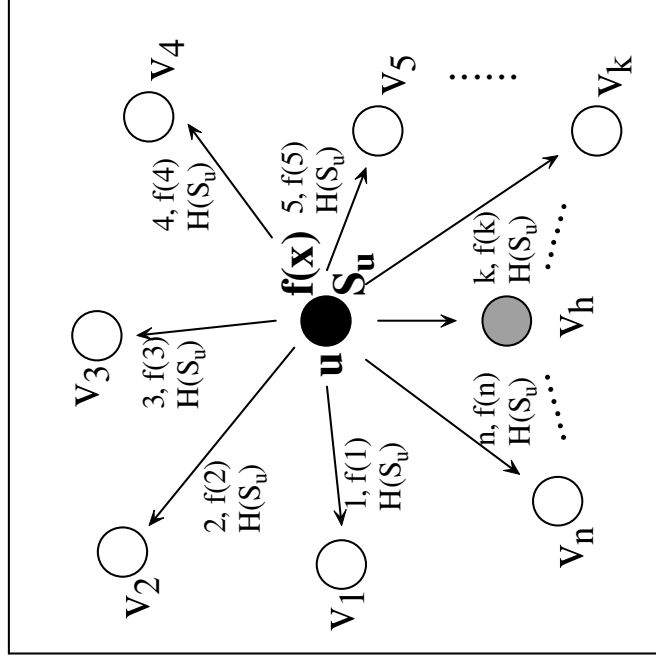
- 1: C is initialized as a 64-bit random number;
- 2: j is initialized to point to C_0 , the first byte of C ;
- 3: **for** $i = 1$ to $i_t/4$ **do**
- 4: $(A_0, A_1, A_2, A_3) = RC5_i$;
- 5: **for** $k = 0$ to 3 **do**
- 6: $C_j = C_j + (Mem[A_k \pmod m]) \oplus \dots \oplus Mem[A_k + b - 1 \pmod m])$;
- 7: $j = (j + 1) \pmod 8$;
- 8: **end for**
- 9: **end for**
- 10: return C ;

Theorem 4.1 *In block-based pseudorandom memory traversal, suppose b is block size, m is memory size, and random variable Y represents the number of traversal iterations needed to cover each memory cell at least once, then $E(Y) = O(\frac{m \ln m}{b})$ and $Pr[Y > \frac{cm \ln m}{b}] \leq m^{1-c}$, where c is a constant factor.*



Scheme I – Based on Threshold Secret Sharing

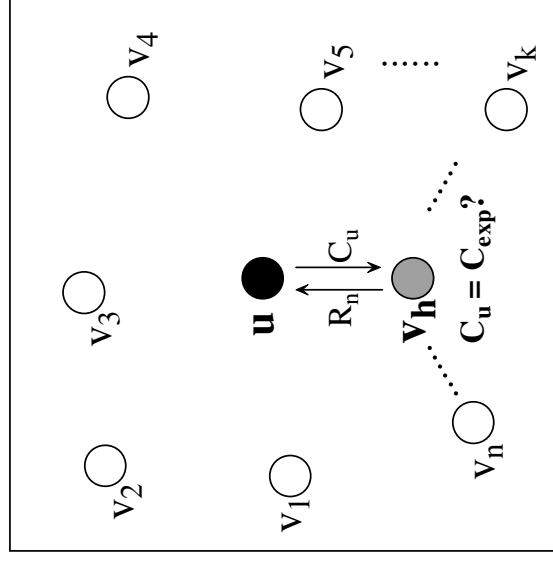
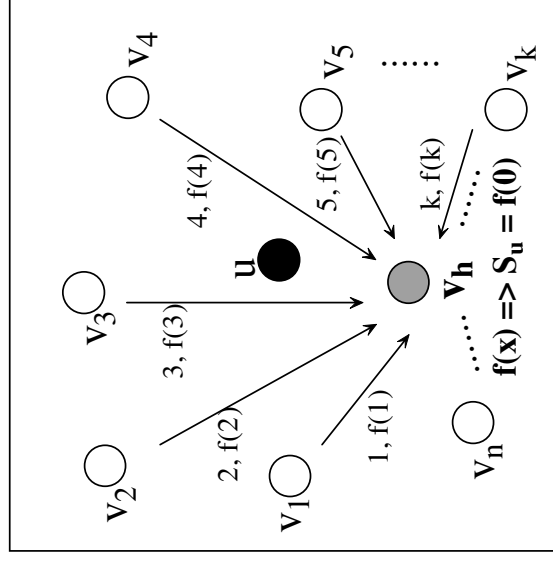
- Phase I – offline phase
 - After deployment, discovers neighbors and meanwhile starts a timer that will expire after T_{min} .
 - splits S_u into multiple shares and sends a separate share to each neighbor; a hash value $H(S_u)$ computed is also included in the message.
 - When the timer expires, it removes S_u from the memory.





Phase II – Online Phase

- collect shares
- verify shares S_u
- send a challenge C_u
- compute a local R_u' based on S_u and C_u
- receive response R_u
- compare R_u and R_u'
- make decision





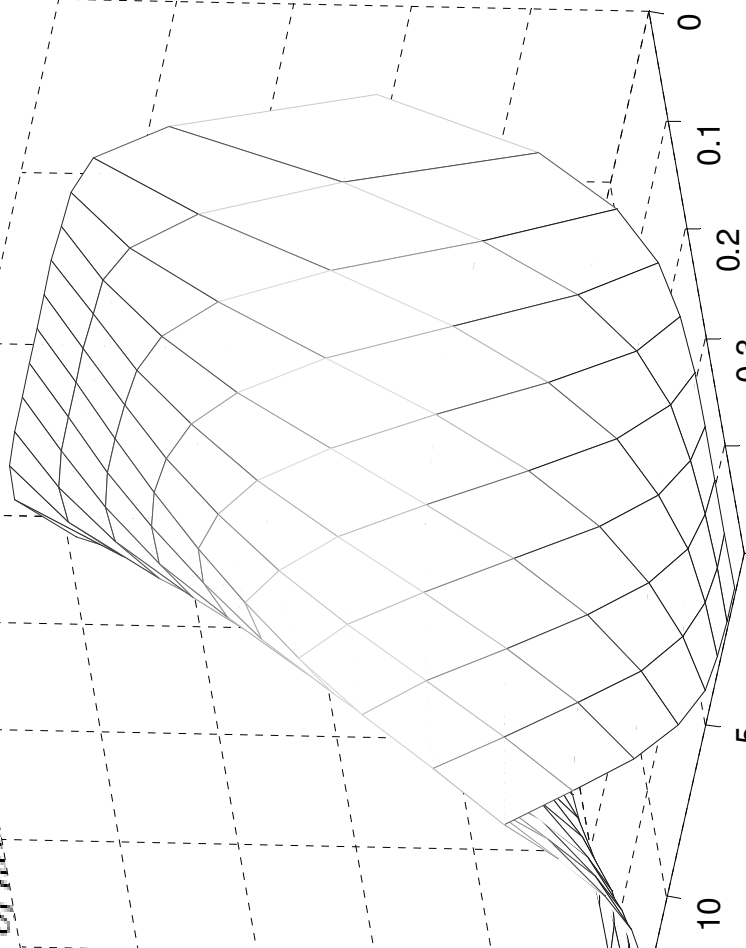
Detection Rate Analysis

Lemma 5.1 Scheme 1 is able to successfully detect a compromised node u , if: (1) cluster head is trustworthy; (2) cluster head obtains $\geq k$ correct shares of noise from neighbors of node u $\geq k$ shares to recover the

Theorem 5.2 Assuming in the network to be com $p_0(0 < p_0 < 1)$ when the

$$P_{bs}(k, n) = \left\{ \begin{array}{l} \sum_{i=k-1}^{n-1} \binom{n-1}{i} p_0^i (1-p_0)^{n-1-i} \\ \sum_{i=n-k}^{n-1} \binom{n-1}{i} p_0^i (1-p_0)^{n-1-i} \end{array} \right.$$

- False Positive $P_{fs} = p_0$

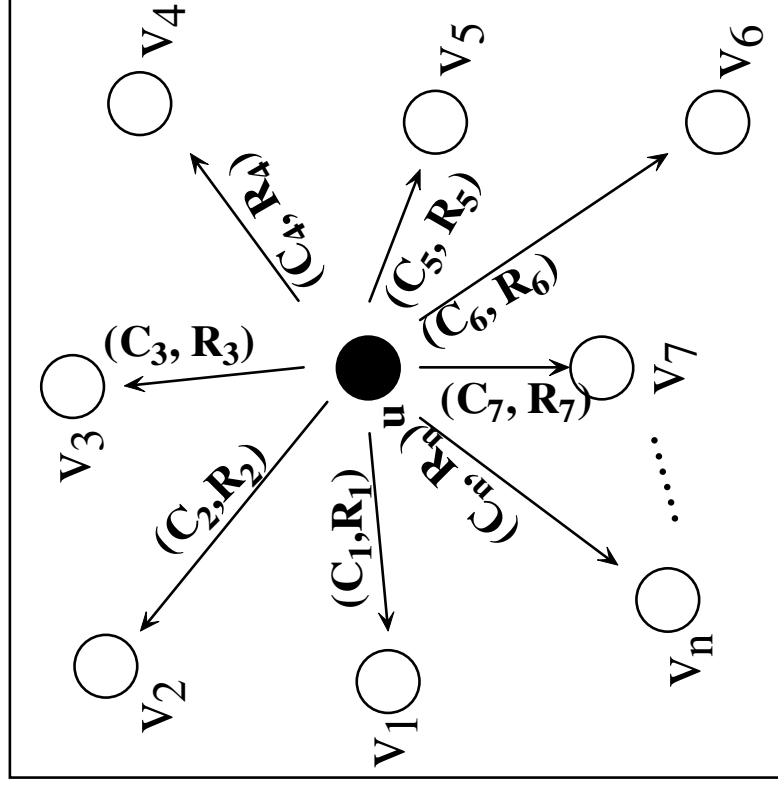


$k: 1 \sim 15$ $p_0: 0.05 \sim 0.5$



Scheme II -- A Majority Voting Based Attestation Scheme

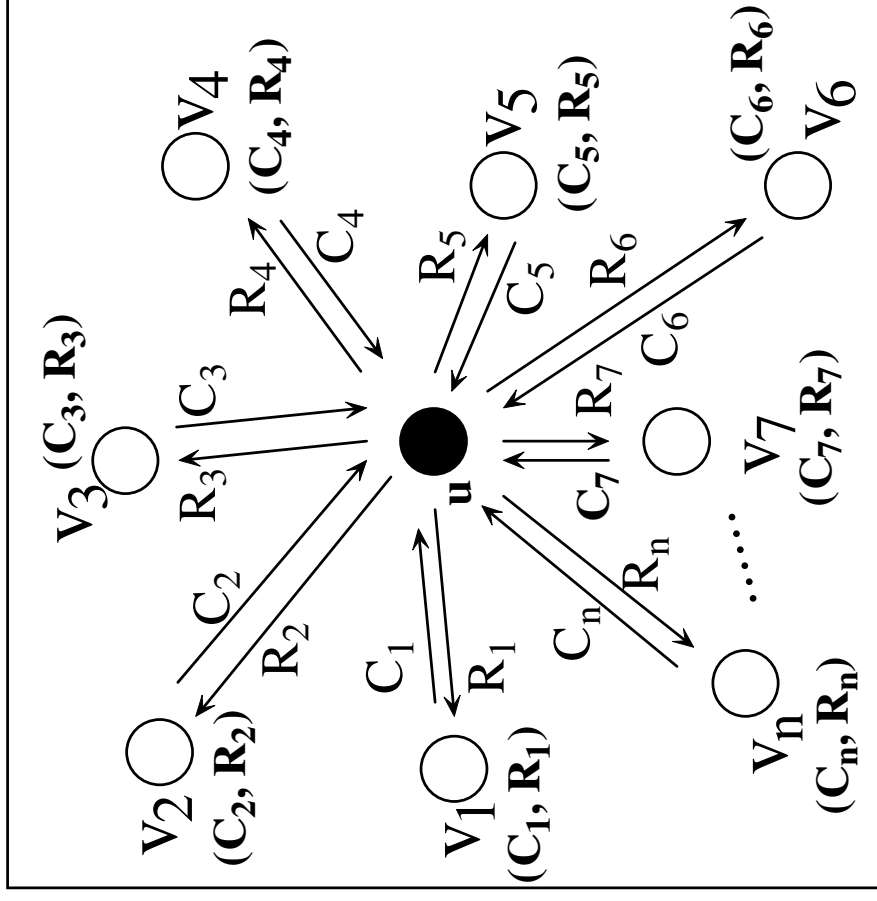
- no cluster head
- pre-generated challenge / responses pairs and every neighbor receives one pair
- for each challenge, #memory traversal is only 1/n of that in Scheme I





Real-time Attestation

- doubting neighbors may challenge
- when the received responses do not match with local ones, negative vote
- when more than $n/2$ nodes vote negatively, considered compromised





Security Analysis

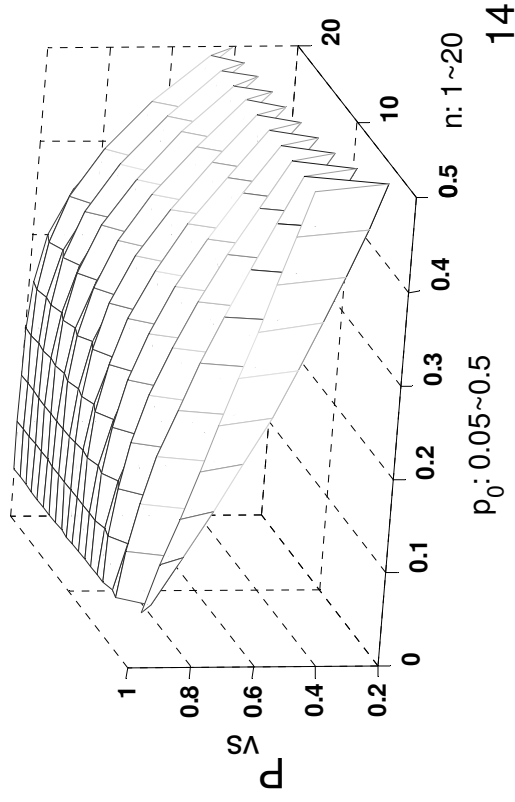
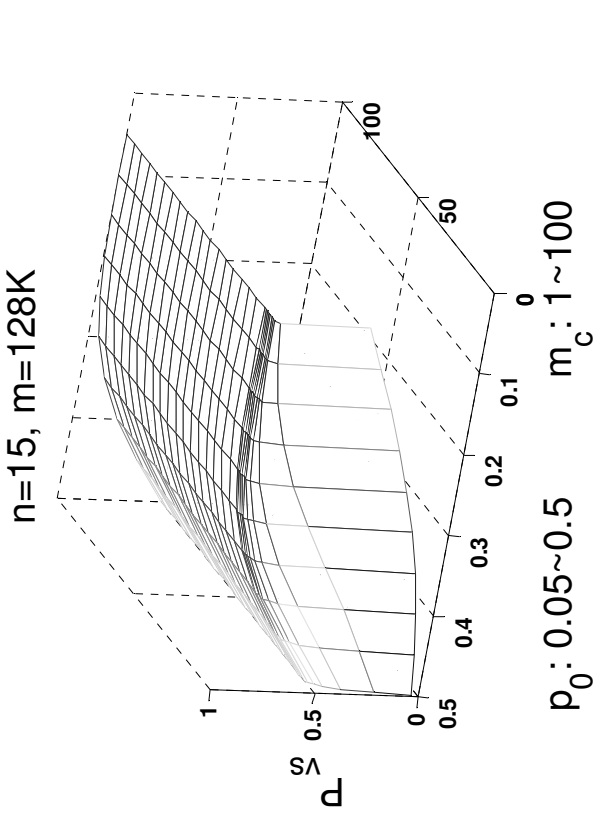
- Detection Rate

Theorem 1.3 Assuming that the probability for each node in the network to be compromised is the same and equals to $p_0(0 < p_0 < 1)$. For Scheme II with regard to node u , suppose m_c is the number of changed memory cells of node u , m is node u 's memory size, and we choose $i_4 = \frac{m \ln m}{bn}$, then the detection rate of Scheme II is $P_{vs}(n, m_c, m) = \sum_{i=\lceil \frac{n+1}{2} \rceil}^n \binom{n}{i} (1-p_0)^i p_0^{n-i} \sum_{j=\lceil \frac{n+1}{2} \rceil}^i \binom{i}{j} p_h^j (1-p_h)^{i-j}$, where $p_h = 1 - (\frac{m-m_c}{m})^{\frac{m \ln m}{n}}$ is the probability for an honest neighbor to detect the compromised node u .

- False Negative

Corollary 5.5: Assuming that the probability for each node in the network to be compromised is the same and equals to $p_0(0 < p_0 < 1)$, then the false positive rate of Scheme II is $P_{vf}(n) = \sum_{i=\lceil \frac{n+1}{2} \rceil}^n \binom{n}{i} p_0^i (1-p_0)^{n-i}$.

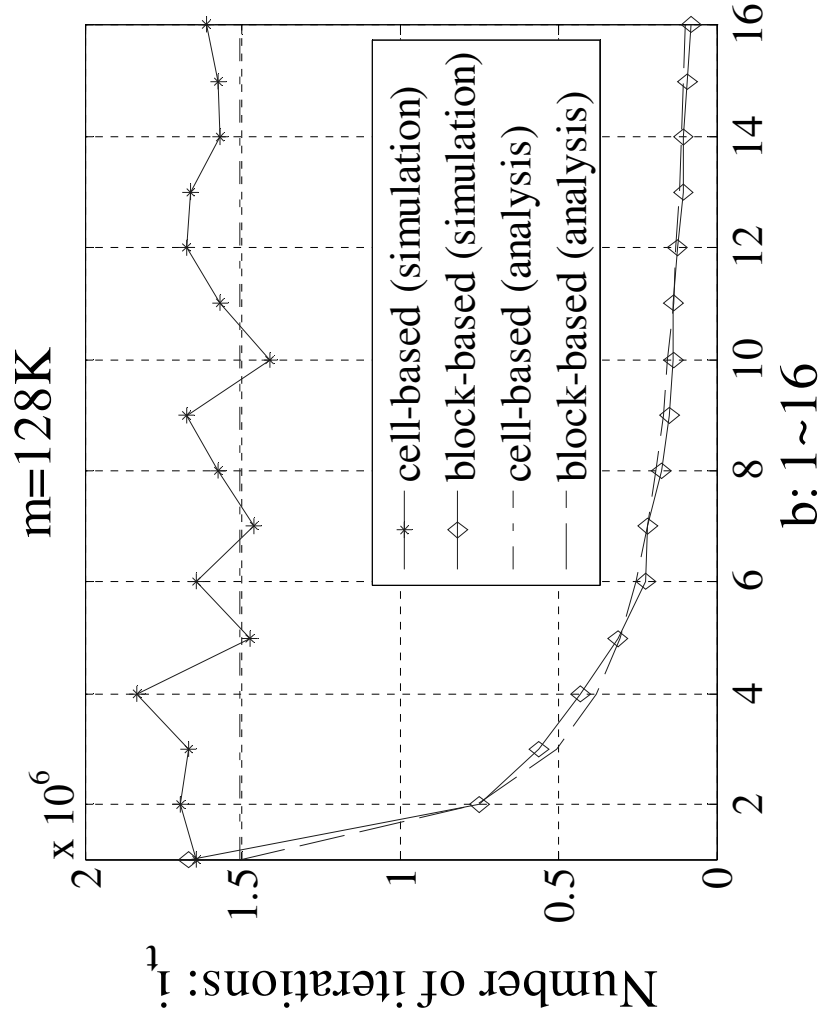
For example, when $n = 15$ and $p_0 = 0.2$, the false positive rate of Scheme II equals to 0.42%.





Performance Evaluation (1)

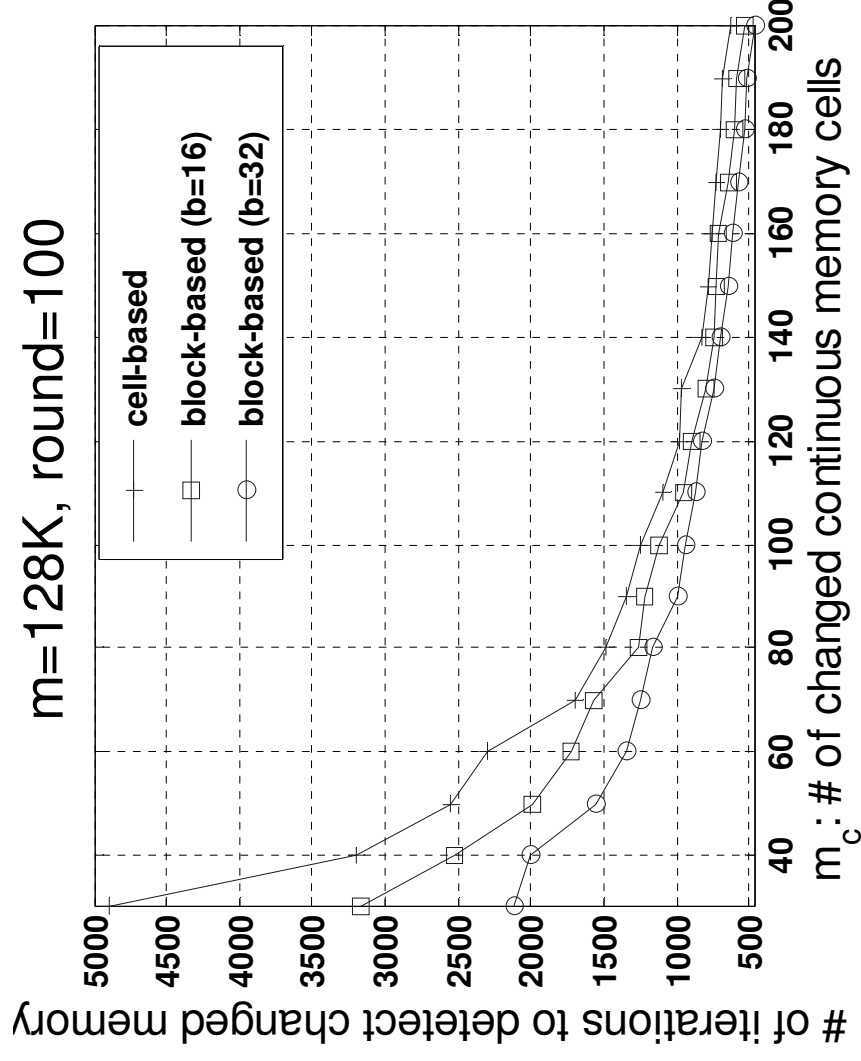
- Number of iterations for detecting a single-cell change





Performance Evaluation (2)

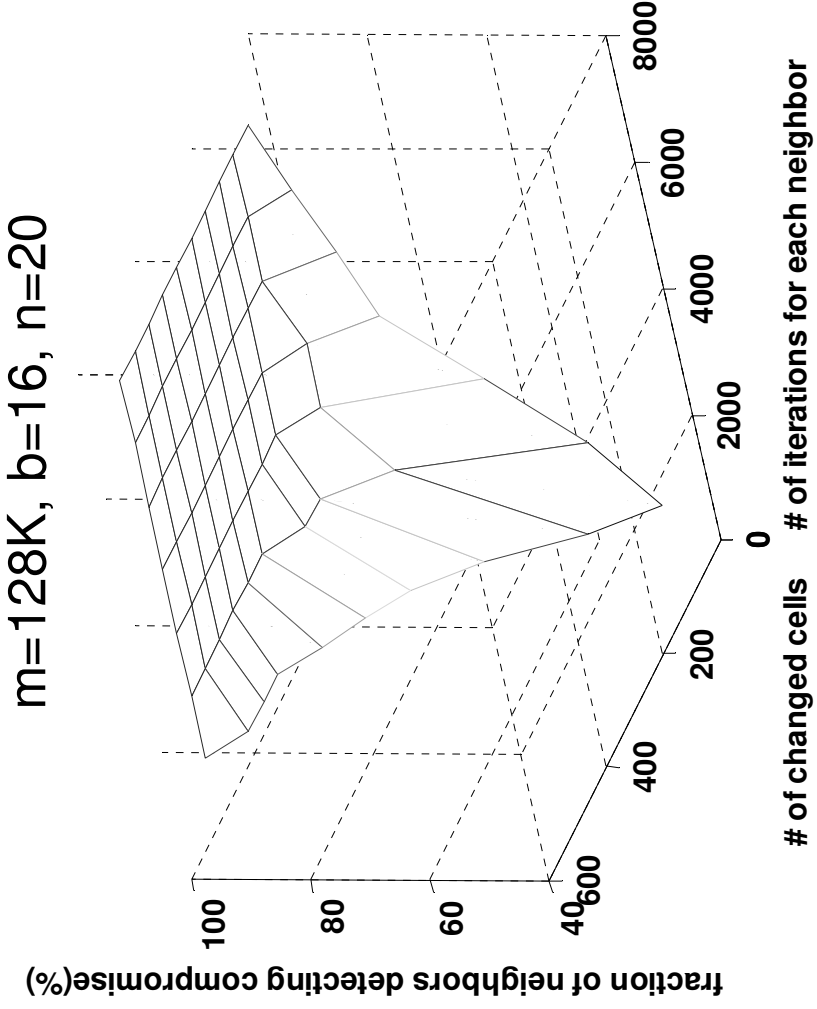
- Performance of Scheme I





Performance Evaluation (3)

- Performance of Scheme II



- When 50 cells are changed, every neighbor can detect the change with 6272 memory traversals



Implementation

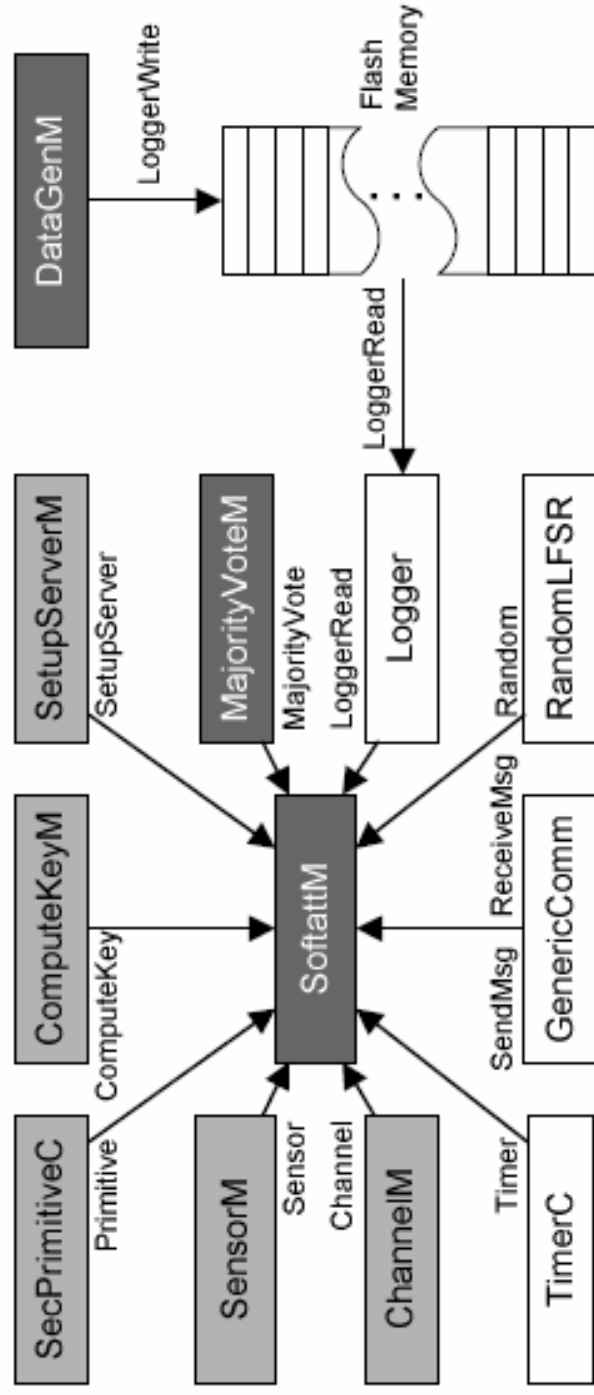


Fig. 11. The relationship among components in the implementation: Dark gray components are developed by us; light gray components are provided by TinyOS; white components are directly adopted from TinyOS.

n	1	3	5	7	9
RAM(bytes)	963	1059	1155	1251	1347

Code memory: 21KB

data memory consumption



Comparison of Two Schemes

TABLE I
SECURITY COMPARISON OF OUR SCHEMES

	Detection Rate	False Positive	Attacker Reward	Eavesdrop	Replay	Message Dropping	Compromised Neighbors
Scheme I	High	Low	Low	Yes	Yes	No	Yes, except to compromised cluster head
Scheme II	Higher	Lower	Lower	Yes	Yes	Yes	Yes

Notations: Yes - the scheme can defend against this attack;
No - the scheme is vulnerable to this attack.

TABLE II
PERFORMANCE COMPARISON OF OUR SCHEMES

	Computation		Neighbors	Communication	Storage
	Attested Node				
Scheme I	$n(k-1)$ -degree poly. eval., 1 hash comp., $O(\frac{m \ln m}{b})$ trav.		1 $(k-1)$ -degree poly. interp., 1 $(k-1)$ -degree poly. eval., 1 hash comp., $O(\frac{m \ln m}{b})$ trav.	$(n+k)L_s + nL_h + L_c$	$n(L_s + L_h)$
Scheme II	$O(\frac{m \ln m}{b})$ trav.		$O(\frac{m \ln m}{b})$ trav.	$2n(L_s + L_c)$	$n(L_s + L_c)$



Conclusion and Future Work

- Two distributed schemes to make software-based attestation more practical for sensor networks
 - Neighbors of a suspicious node collaborate to make a joint decision
 - No dependence on response time measurement by mobile verifier or base station
- Future Work
 - Handle Network topology change
 - Minimize transmissions collision
 - Apply to different sensor memory architectures



Thank you!

- Questions?