Volume 2, No. 3, May-June 2011

ISSN No. 0976-5697

International Journal of Advanced Research in Computer Science

RESEARCH PAPER

Available Online at www.ijarcs.info

An Improved Key Pre-distribution Scheme and Deployment model for Wireless SENSOR Networks

Sophiadas D.Y
Centre for IT & Engineering
Manonmaniam Sundarnar University
Tirunelveli, India
sophia_dy_2005@yahoo.co.in

Divya.C*
Assistant Professor, Centre for IT &Engineering
Manonmaniam Sundarnar University
Tirunelveli, India
cdivyame@gmail.com

Krishnan.N*
Prof.& Head,Centre for IT &Engineering,
Manonmaniam Sundarnar University,
Tirunelveli, India

Abstract— Key distribution plays an important role in wireless sensor networks. A wireless sensor network has a large number of tiny sensors with limited computational capability memory space and power resource. Many key pre-distribution schemes have been developed to establish pairwise keys for WSN. In WSN, node capture attack is the most series attack. To improve the resistance against the node capture attack, this paper proposes a hashed key pre-distribution scheme, which uses Hash function to stop an adversary to get information of non-compromised sensor nodes from the compromised sensor nodes and the deployment model, is based on hexagonal to improve the local connectivity. The proposed scheme can provide the best resilience against sensor nodes capture and the probability of links between any sensor nodes are compromised is zero after pairwise keys establishment. Our new scheme can be used in a large network and achieves good network connectivity.

Keywords— Wireless sensor networks; Key Management; Key pre-distribution scheme; Hash function

I. INTRODUCTION

Wireless sensor networks (WSN) are a special kind of ad-hoc network, but have many new application areas such as military target tracking, environment monitoring, patient monitoring and scientific exploration in dangerous environment etc. When sensor networks are deployed in a hostile environment, security becomes extremely important, as they are vulnerable to different types of malicious attacks. Public key such as Elliptic Curve Cryptography (ECC) can achieve good security performance [1]. But it requires high computation and communication ability, which is not suit for resource-limited WSNs since sensor nodes usually have only 8-bits CPU. Thus, the symmetric key is still a better choice for WSNs.

However, key distribution is a challenging problem for symmetric key cryptography. A primitive way of key distribution is to give all the nodes in the network a same key. Then they can communicate with each other using the key. This scheme is suitable for large scale networks, but as all the nodes use only one key, it is easy for the adversary to break the entire network if it gets the key. Many key pre-distribution schemes proposed in literature are trade-offs between security and complexity. The common drawback is that, a number of compromised nodes may cause either a fraction of the remaining network to become insecure, or the entire network broken [2,3,4,5].

schenauer and Gligor proposed the basic probabilistic key pre-distribution, in which each sensor node picks a random subset of keys from a large key pool before deployment of the network [5]. By doing this, two sensor nodes can have a certain probability to share at least one key. This common key can be used as a shared secret key. Now the existence of a shared key between a particular pair of nodes is not certain but is instead guaranteed only probabilistically (This probability can be tuned by adjusting the parameters of the scheme).Eschenauer and Gligor note that this is not an insurmountable problem as long as any two nodes can securely communicate via a sequence of secure links.

A generalization of this is the “q-composite” scheme [Chan et al. 2003,] which improves the resilience of the network (for the same amount of key storage) and requires an attacker to compromise many more nodes in order to compromise additional communication links. The difference between this scheme and the previous one is that the q composite scheme requires two nodes to find q (with q > 1) keys in common before deriving a shared key and establishing a secure communication link. It is shown that, by increasing the value of q, network resilience against node capture is improved for certain ranges of other parameters [6].The number of required shared keys makes it exponentially harder for the attacker to compromise a link key with a given subset of already compromised keys. For these two random pairwise keys schemes, a small number of compromised sensor nodes may reveal a large fraction of pairwise keys scheme between non compromised sensor nodes.

Du et al. [7] further improved the random key distribution scheme by integrating Blom’s key pre-distribution mechanism [9] into random key pre-distribution [5]. Liu et al.[8] applied a similar idea with Blundo’s scheme[10]. In these two schemes, the communication between non-compromised sensor nodes keeps secure when the number of compromised sensor nodes is less than a critical value. But once the critical value is exceeded, the adversary would crack all the pairwise keys. Based on the previous works, Liu et al. proposed several location-based pairwise key establishment methods for WSNs [11][12][13].Although Liu's schemes improved the performance of secure, they assume the sensor's location can be predicted before the development. We argue that in most applications, especially in military fields, it is impossible to predict the sensor's location before their deployment.

II. NETWORK MODEL

A WSN is a large network of resource-constrained sensor nodes with multiple preset functions, such as sensing and processing, to fulfil different application objectives [5,13,14]. Usually, sensor nodes are deployed in a designated area by an authority such as the government or a military unit and then, automatically form a network through wireless communications. Sensor nodes are static most of the time, whereas mobile nodes can be deployed according to application requirements. One or several base stations (BSs) are deployed together with the network. A BS can be either static or mobile. Sensor nodes keep monitoring the network area after being deployed. After an event of interest occurs, one of the surrounding sensor nodes can detect it, generate a report, and transmit the report to a BS through multi hop wireless links. Collaboration can be carried out if multiple surrounding nodes detect the same event. In this case, one of them generates a final report after collaborating with the other nodes. The BS can process the report and then forward it through either high-quality wireless or wired links to the external world for further processing. The WSN authority can send commands or queries to a BS, which spreads those commands or queries into the network. Hence, a BS acts as a gateway between the WSN and the external world. An example is illustrated in Fig. 1.

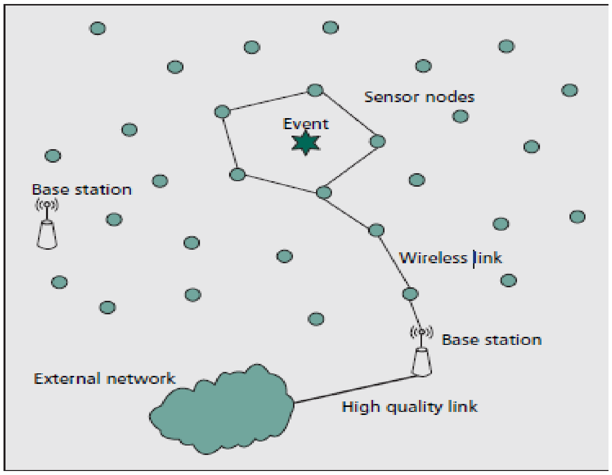

Figure 1: Wireless Sensor Network Model

Because a WSN consists of a large number of sensor nodes, usually, each sensor node is limited in its resources due to the cost consideration in manufacturing. For example, MICA2 MPR400CB [15], which is the most popular sensor node platform, has only 128 KB of program memory and an 8-bit ATmega128L CPU [16]. Its data rate is 38.4 kbaud in 500 feet, and it is powered by only two AA batteries. The constrained resource cannot support complicated applications. On the other hand, usually, BSs are well designed and have more resources because they are directly attached to the external world.

III. BACKGROUND

In this section, we review the basic scheme proposed in [6].This scheme consists of three phases: key pre-distribution, shared-key discovery, and path-key establishment. In the key pre-distribution phase, a large key pool *S* is generated first. Then, each sensor randomly selects *m* distinct keys from the key pool *S*, and stores them in its memory. This set of *m* keys formed sensor's key ring. The number of keys in the key pool, |*S*|, is chosen such that two random subsets of size m in *S* share at least on key some probability *p*. After the sensor nodes are deployed, the key-setup performed. During this phase, each pair of neighbouring sensor nodes attempts to find a common key that they share. Since all the keys are randomly selected from the same key pool, two sensor nodes may have some overlapped keys in their memories. If such a key exists, the key is used to secure the communication link between these two sensor nodes. After key-setup is complete, a connected graph of secure links formed. Sensor nodes can then set up path keys with their neighbours with which they do not share keys. If the graph is connected, a path can always be found from a source sensor to any of its neighbors. The source sensor can then generate a path key and send it securely via the path to the target sensor.

IV. NOTATION

The following notations are used in this paper:
— *N*: the number of sensor nodes in the network
— *H*: Hash function
— *S*: key pool
— *w*: size of the key pool
— *u,v:* sensor nodes
— *idu*: the ID of sensor *u*
— *Ko :* the original key drawn from key pool, and each original key has a unique ID
— *Kd*: the derivative key in sensor *u* , here *Kd*=*H*(*idu*, *Kp*), which have the same key ID as the original key *Kp*
— *Kuv:* the communication pairwise keys between sensor *u* and sensor *v*
— *t*: the number of keys assigned to each sensor
— *s*: the number of derivative keys in every sensor
— *M1*||*M2*: Concatenation of message *M1* and *M2*

V. THE PROPOSED SCHEME

This paper deals with two things. First is how to deploy the sensor nodes. Deployment of sensor node means locating the sensor nodes into different areas so that it can sense the data specific to that environment. Second phase is distributing keys to each sensor, using that keys nodes can find the pair wise keys for communication.

VI. DEPLOYMENT MODEL OF SENSOR NODES

Assume that nodes will be static after they have been deployed. When nodes are dropped from a high place, such as helicopter, they are distributed by group. Clearly, nodes in the same group or neighbour groups will have higher probability to communicate with each other.The two-dimensional Gaussian distribution model as in [7], which is much closer to the real state.

A. The Hexagon Partition Model

Suppose that the total area is *X* meters in width, *Y* meters in length, and it is divided into $t\times n$ groups. Hexagon is used with the same size to divide the whole area, and the group number increases from left to right at the direction of x-axis, from the bottom up in the direction of y-axis. Let <i, j> ($1\leq i \leq t$, $1\leq j \leq n$) denote the group *ID* shown in Fig.2, whose center is locate at the point (x_i, y_j).

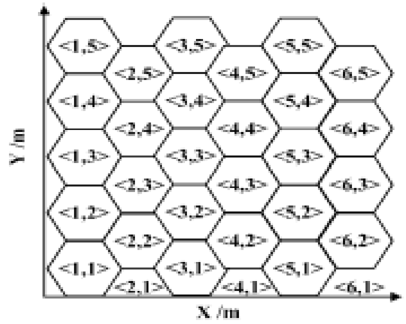

Figure2: Hexagonal partition of groups

VII. KEY DISTRIBUTION PHASE

In most key pre-distribution schemes, the communication pair wise keys between sensor nodes either use the pre-loaded keys directly [5,6], or can be derived from the pre-loaded secret shares[7,8,9,10]. Once some sensor nodes are captured, the adversary may crack other sensor nodes or even the entire network through the compromised keys or secret shares. To address this problem, two kinds of keys are considered in our proposed scheme. On is the original key, which is pre-loaded into sensor nodes just like in [5]. The other is the derived key, which is derived form the original key by using Hash function. In our scheme, part of original keys will be converted into derivate keys in every sensor before sensor nodes deployment. After pairwise keys establishment between sensor nodes all original keys in every sensor are converted into derivative keys. Then all original keys will be erased from every sensor memory. Because it is computational infeasible to revert the Hash function, an attack can't get other sensor nodes information from any compromised sensor nodes. The proposed scheme is presented as follows.

A. Key Pre-Distribution

The Key Distributions Server (KDS) generates a very large size of key pool *S*. For each sensor, KDS randomly selects *t* secret keys from *S* and stores it into the sensor memory. Then each sensor randomly chooses *s* original keys to generate derivative keys according following methods: $K_d=H(i_{du},K_o)$

B. Directed Key Establishment

After deployment, each sensor needs to discovery whether it has a common key ID with its neighbors. To do this, each sensor broadcast a message containing the following information: the sensor's ID, the ID of all keys it carries, as well as the type of each key. Assuming that sensor *u* and sensor *v* are neighbours, and have sent the above broadcast messages. If they determine that have a common key identifier. They can compute the pairwise secret key as follows. There are three cases need to be considered:

Case 1: Both keys are original keys. In this case, sensor *u* and sensor *v* can calculate the communication pairwise key K_{uv} as follows: $K_{uv}=H(K_o \|i_{du} \|i_{dv})$

Case 2: The key sensor *u* is a derivative key, and the key in sensor *v* is an original key. In this case, sensor *u* keeps the derived key K_d, while sensor *v* keeps original key K_o. Senor *u* can calculate the pairwise key: $K_{uv}=H(K_d\|i_{dv})$, while sensor *v* calculates the pairwise key: $K_{uv}=H(H(k_o\|i_{du})|i_{dv})$. It is obviously that $K_{uv}=K_{vu}$;

Case 3: The two keys in sensor *u* and sensor *v* are derivative keys. In this case, these two sensor nodes can't establish a pairwise key directly

C. Phase Key Establishment

If direct key establishment failed, the two sensor nodes can try to establish a pair wise key in the path key establishment phase. When a source sensor broadcast the ID of a destination sensor , an intermediate sensor can establishment a path key for the two sensor nodes if it holds the pairwise with the source and with the destination sensor nodes, respectively. Otherwise, the intermediate sensor broadcast the message continuously until it discovers a sensor that shares pairwise keys with the previous sensor and the destination sensor, respectively. The path key can be establishment along the message broadcast path in the reverse direction.

D. Original Key Conversion

After pairwise keys establishment between sensor nodes all original keys K_o in sensor *u* are converted into derivative keys as in K_d, as $K_d=H(i_{du},K_o)$. Then all original keys K_o are erased from sensor *u*.

E. Sensor Addition and Revocation

Some sensor nodes may be destroyed or compromised after a period of time. Then, they can no longer work properly. This problem can be deal with by adding new sensor nodes. Due to all the sensor nodes in the network only have derivate keys now, the adding sensor can only have original keys. After the new sensor establishes pairwise key with the working sensor nodes, the original keys in the new sensor nodes will be changed to deviate keys and the all the original keys will be erased from the new adding sensor. Sometimes is necessary to revoke sensor nodes from the sensor networks possibly due to sensor nodes compromise. To revoke sensor, the others sensor nodes that have a shared pairwise key with the revoked sensor only need to remove the shared pairwise keys from their memory.

VIII. PERFORMANCE ANALYSIS

In this section, the security property and networks performance of the proposed scheme are evaluated, and compared put scheme with several key pre-distribution scheme. The analytical result on the two metrics: local connectivity and resilience against sensor capture are analysed.

A. Local Connectivity

The local connectivity P_c *is* the probability of two neighboring sensor nodes can establish communication pairwise keys directly. For any pair of nodes to be able to find a secret key between them, the key sharing graph needs to be *connected*. The following three-step approach, adapted from [Eschenauer and Gligor 2002].

Step 1: Computation of required local connectivity. Let P_{global} be the probability that the key sharing graph is connected. This is called as *global connectivity*. Let *local connectivity* P_c refer to the probability of two neighboring nodes sharing at least one space. The global connectivity and the local connectivity are related: to achieve a desired global connectivity P_{global}, the local connectivity must be higher than a certain threshold value called the *required local connectivity*, and denoted by $p_{required}$. Using results from the theory of random graphs [Erdos and R´enyi 1959], The average node degree d to the global connectivity probability P_{global} in a network of size N (for N large) are related:

$$d = \frac{(N-1)}{N}\,[\ln(N) - \ln(-\ln(P_{global}))] \qquad (1)$$

For a given density of sensor network deployment, let n be the expected number of neighbours within wireless communication range of a node. Since the expected node degree in key sharing graph should be at least d as calculated above, the required local connectivity $p_{required}$ can
be estimated as:

$$p_{required} = \frac{(d)}{N} \qquad (2)$$

Step 2: Computing actual local connectivity. After selecting the values for ω and τ , the actual local connectivity is determined by these values. p_{actual} is used to represent the actual local connectivity; namely, p_{actual} is the actual probability of two neighbouring nodes sharing at least one key space (which is the same as the probability that they can establish a common key). Since p_{actual} = 1 − Pr(two nodes do not share any space).

$$p_{actual} = 1 - \frac{\binom{\omega}{\tau}\binom{\omega-\tau}{\tau}}{\binom{\omega}{\tau}^{2}} = 1 - \frac{((\omega-\tau)!)^{2}}{(\omega-2\tau)!\,\omega!} \qquad (3)$$

and this equation can be rewritten as

$$p_{actual} = 1 - \frac{((\omega-1.5\tau)!)^{2}}{(\omega-3\tau)!\,\omega!}$$

Values of p_{actual} have been plotted in Fig. 2 for τ = 2, 4, ω varying from τ to 100. For example, one can see that when τ = 4, the value of ω must be at most 25 in order to achieve local connectivity $p_{actual} \geq 0.5$.

$$p_{actual} = 1 - \frac{((\omega-\tau)!)^{2}}{(\omega-2\tau)!\,\omega!}$$

Step 3: Computing ω and τ . Knowing the required local connectivity $p_{required}$ and the actual local connectivity p_{actual}, in order to achieve the desired global connectivity P_{global}, we should have $p_{actual} \geq p_{required}$. Thus:

$$1 - e^{-\frac{\tau^{2}}{\;}} \geq \frac{(N-1)}{nN}\,[\ln(N) - \ln(-\ln(Pglobal))]$$

So, in order to achieve a certain P_c for a network of size N with n expected neighbours for each node, we just need to find values of ω and τ such that Inequality (4) is satisfied.

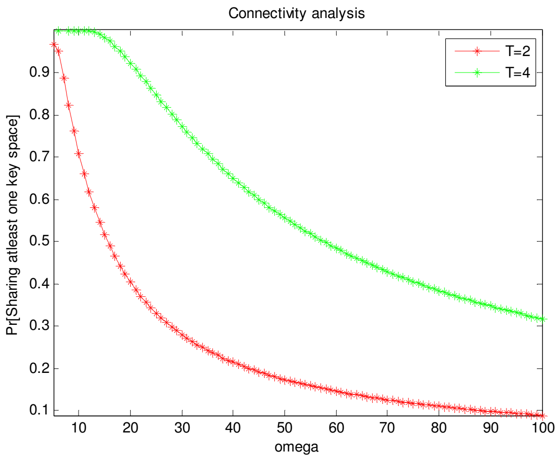

Figure 3. Probability of two nodes sharing a key when each node hold _ key spaces chosen randomly from a set of □ key space.

B. Resilience Against Sensor Capture

We assume that an adversary can mount a physical attack on a sensor after it is deployed and read secret information from its memory. According to our key pre-distribution scheme, an adversary can't get any key information from a compromised sensor after pairwise key establishment, because it is computational infeasible to revert the hash function. An adversary can only get key information before pairwise establishment, so we only need to analyze the resilience before pairwise establishment. The resilience of the scheme is measured as the fractions of total network communication that are compromised when *x* sensor nodes are captured [6]. Here *x* is the number of total captured sensor nodes. Hence, we have the probability *Pb*, that any secure link between two uncompromised sensor nodes is compromised when *x* sensor
nodes have been captured is

$$P_b = 1 - \left(1 - \frac{t-s}{2w}\right)^{x}$$

Figure 4. shows the resiliency of the scheme during pairwise key establishing. The figure shows that during pairwise key establishment the resiliency become stronger when the number of derivative key increase. This is the more derivative key the less key information disclosing from the comprised sensor.

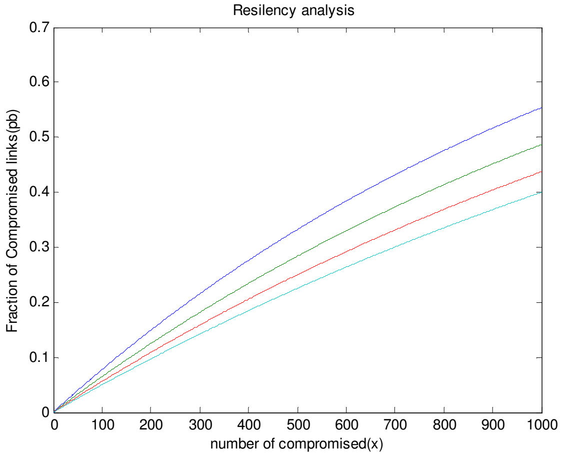

Figure 4 Fraction of compromised links compromised between non-compromised sensor nodes v.s Number of compromised sensor nodes.

C. Comparision with other Scheme

We selected strong key pre distribution scheme for comparison with our scheme. The security is mainly compared here, which is the fraction of affected non-compromised.

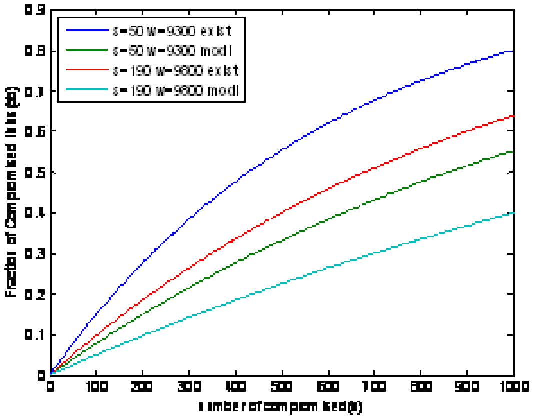

Figure 4 Comparison with Existing and our scheme

Since our scheme is perfectly secure after pairwise key establishment, Figure 4 only show the resiliency of our scheme before pairwise key establishment. The figure clearly shows the advantages of our scheme. What is more important is that in most case the life time of sensor networks is usually much longer that the time of pairwise key establishment. And in some case comparing with the life time of the sensor networks, the time of pairwise establishment can be negligible.

IX. CONCLUSIONS

In this paper, the hashed key management scheme and deployment model for wireless sensor networks was proposed. The proposed scheme uses Hash function to prevent attackers get information of non compromised sensor nodes from the compromised sensor nodes. Compared to existing key pre-distribution schemes, the proposed scheme is substantially more resiliency against sensor nodes capture. And the main advantage of our scheme is that the sensor networks are perfectly secure again sensor nodes capture after pairwise establishment. Taking into out that the time of establishing pairwise keys is usually very short our scheme can perform well in reality. Also this scheme gives better local connectivity and resilience against the node capture attack.

X. REFERENCES

[1] A. Liu and P. Ning, "TinyECC: a configurable library for ellipticcurve cryptography in wireless sensor networks," in Proc. IPSN 2008, Washington, DC, Apr. 2008, pp. 245-256.

[2] Neha Jain and Dharma P. Agrawal, "Current trends iwireless sensor networks. International Journal of Distributed Sensor Networks", 1(1)(2005)101-122

[3] S.Zhu, S. Setia, and S. Jajodia, "LEAP: Efficient security mechanism for large-scale distributed sensor networks". in: Proc. of the 10the ACM Conference on Computer and Communications Security, Washington D.C, USA, Oct.(2003)62-72

[4] A. Perrig, R.Szewczyk, V. Wen, D. Culler, and J.D. Tygar, "SPINS:Security protocls for sensor networks", Wireless Networks, 8(5)(2002)521-534

[5] Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A survey on sensor networks", IEEE Communication Magazine, vol.40. no8, Aug(2002)102-116

[6] L. Eschenaure and V.D. Gligor, "A key-management scheme for distributed sensor networks". in: Proc. of the 9the ACM Conference on Computer and Communications, Washington DC, USA, Nov. (2002)

[7] H. Chan, A. Perrig and D. Song, "Random key predistribution schemes for sensor networks", in: Proc. 20003 IEEE Symposium on Security and Privacy, 11-14 May (2003)197-313

[8] W.Du, J. Deng, Y.S. Han, P.K. Varshney, J. Katz, and A. Khalili, "A pairwise key predistribution schemes for sensor networks networks". ACM Transactions on Information and System Security, Vol.8. No2, May (2005)228-258

[9] D. Liu, P. Ning, and R. Li, "Establishing pairwise keys in distributed sensor networks". ACM Transactions on Information and System Security, vol.8, No.1, Feb. (2005)41-77

[10]R. Blom, "An optimal class of symmetric key generation systems. Advance in Cryptography". London, UK: Springer-Verlag, 1985.335- 338

[11]C. Blundo, A. D. Santis, A. Herzberg. S. Jutten, U. Vaccaro, and M. Yung. "Perfectly secure key distribution for dynamic conference", Information and Computation, 1995,146 (1):1-23

[12]D.Liu D, P. Ning. "Location-based pairwise key establishments for static sensor networks." in: Proc of the 1st ACM Workshop on Security of Ad Hoc and Sensor networks. New York, NY, USA: ACM Press, 2003. 72~82

[13]J. M. Kahn, R. H. Katz, and K. S. J. Pister, "Next Century Challenges:Mobile Networking for Smart Dust," Proc. ACM Int'l.Conf. Mobile Computing and Networking (MobiCom'99), Aug. 1999, pp. 217–78.

[14]G. J. Pottie and W. J. Kaiser, "Wireless Integrated Network Sensors," Commun. ACM, vol. 43, no.5, May 2000, pp.51 58.

[15]Crossbow Technology; http://www.xbow.com/, 2006.

[16]Atmel Corporation; http://www.atmel.com/, 2006.

[17]ERD˝OS AND R´ENYI. 1959. On random graphs I. Publ. Math. Debrecen 6, 290–297.