Random Channel Hopping Schemes for Key Agreement in Wireless Networks

Bin Zan
Winlab, Rutgers University
Technology Center of New Jersey
671 Route 1 South
North Brunswick, NJ 08902-3390
Email: zanb@winlab.rutgers.edu

Marco Gruteser
Winlab, Rutgers University
Technology Center of New Jersey
671 Route 1 South
North Brunswick, NJ 08902-3390
Email: gruteser@winlab.rutgers.edu

***Abstract*—Secure wireless communications typically rely on secret keys, which are hard to establish in a mobile setting without a key management infrastructure. In this paper, we propose a channel hopping protocol that lets two stations agree on a secret key over an open wireless channel and without use of any pre-existing key. It is secure against an adversary with typical consumer radio hardware that only allows receiving on a single (or a few) channel at a time. Theoretical analysis and simulation results indicate that this approach can generate a 128-bit key in 0.3 seconds. This is significantly faster than prior techniques that extract key material from the wireless channel.**

I. Introduction

Key agreement, the process through which two parties share a secret key, is a fundamental challenge in networking security. Traditional key management infrastructure is not typically available in a mobile wireless network, particularly in unmanaged consumer environments. Thus, current Wi-Fi security protocols such as WPA [1], rely on the cumbersome procedure of manually entering keys or passphrases.

Prior work on key agreement in sensor networks and ad hoc networks has largely focused on pre-distribution protocols (e.g., [2]). In such protocols a large pool of symmetric keys is chosen and a random subset of the pool is distributed to each node. Thus, two nodes can establish a session key if they share a common key. However, if nodes are mobile, this is hard to achieve. Prior work in information theory has shown that it is possible to establish keys over open wireless channels, but few practical protocol designs and evaluations exist. One recent study [3] showed that it is possible to achieve strong information-theoretic security by extracting secret bits from a multi-path fading channel at a rate of about 1 bit per second.

In this paper, we propose a faster way to generate secret keys through a randomized channel hopping protocol, which can be implemented on IEEE 802.11 radios without hardware modifications. For faster key agreement, this protocol could be incorporated into a frequency-hopping spread spectrum physical layer implementation. This approach is effective against an adversary who can at most monitor $n-1$ of the n channels at any given time. It provides anonymous key agreement, the parties are not authenticated.

We envision several possible applications of this technique. First, it could be used for key generation in consumer wireless devices. If devices first generate sufficiently long keys and limit the rate of authentication attempts, only entry of much shorter passwords would be required (for authentication purposes). Second, it could also be used to generate additional key material to refresh session keys. Third, it could be used to establish pseudonyms in an ad hoc network, perhaps in conjunction with a reputation system that establishes stronger notions of identity and trust over time.

The remainder of this paper is organized as follows. Section 2 discusses related work. Section 3 describes the proposed channel hopping protocol and gives a theoretical analysis. Section 4 evaluates the schemes by simulation. Section 5 discusses the security of the protocol and Section 6 concludes.

II. Related Work

Traditional key distribution protocols rely on infrastructure with online trusted third parties (TTP), such as the well-known Kerberos [4] scheme and Otway-Rees protocol [5]. However, in mobile ad hoc networks, the lack of infrastructure implies that there is no central authority that can be referred to when it comes to make trust decisions about other parties in the network and when that accountability cannot be easily implemented. Furthermore, since the node mobility is unrestricted, the topology may be unpredictable making central authority assumption infeasible.

Diffie and Hellman discussed a public key distribution system and how it can be transformed into a one-way authentication system [6]. Their scheme, Diffie-Hellman key exchange, was based on the apparent difficulty of solving discrete logarithm problem [7]. An even earlier contribution is Merkle's "puzzle" [8]. Cost-effective processors with limited computational abilities make public-key cryptography almost impractical for embedded intelligence and ubiquitous computing applications, even without power consumption considerations.

In [9], Rolf Blom presented a symmetric key generation system (SKGS), where each pair of users share one master key that is distributed at the start up time by a key generation authority. This master key is used to generate session keys later on. However, a network with n users implies that each user must have access to $n-1$ keys, if n is large, it becomes

978-1-4244-5213-4/09/ $26.00 ©2009 IEEE

impossible to store all keys securely. The contribution of Rolf Blom is on finding a class of symmetric key based on an MDS code [10], for which the amount of secret data needed by each user is very small. On the other hand, a certain minimum number of K users have to cooperate to determine the keys used by other user pairs. Eschenauer and Gligor [11] proposed a key management scheme that relies on probabilistic key sharing among the nodes of a random graph. Other pre-distribution schemes can be found in [12], [13], [14], [15], [16]. However, the strict requirement for pre-distribution might not be available always. For example, in a mobile ad hoc network, the nodes or the users (sharing no prior secret information) may just meet on the spot where there is likely to be no single trustable proxy or TTP for key pre-distribution.

Hershey et al. [17] first presented the concept of using physical layer characteristics for key management, in which the key is generated based on the impulse response similarity between Alice and Bob. More recent work can be found in [3], [18], [19]. In such key distribution systems, special hardware and equipments are needed, which might not be practical in the real world. Also, the threshold values used to determine the value of each key bit are not reliable.

Recently, some researchers started to exploit multi-channel characteristic of wireless devices to help improving security. For example, in [20], Strasser et al. use frequency hopping to establish a secret key in the presence of a communication jammer. However, this scheme still relies on ECC-based public key cryptography. Miller and Vaidya [21] proposed a method of symmetric key establishment for a sensor network that exploits channel diversity to create link keys for one-hop neighbors. However, the proposed method needs multiple nodes cooperation and requires a relatively dense network limiting the range of its usage.

III. Channel Hopping Key Agreement

In our model, we assume one transmitter (Alice), one legitimate receiver (Bob) and one passive eavesdropper (Eve). Everyone can communicate on multiple, non-interfering channels, but receive on a single (or a few) channel at a given time. As in [21], we assume that the hardware Eve has is similar to Alice's and Bob's. Alice and Bob seek to establish a secret key without any prior shared information.

A. Basic Packet-Based Scheme

For expository reasons, we first describe a basic packet-based protocol before we proceed to the final multi-agreement scheme. The idea underlying this scheme is that both parties of the key agreement process—Alice and Bob—randomly select a channel to send and listen to, respectively. If they choose the same channel, key information is successfully transferred and Bob sends an acknowledgment (ACK). Otherwise, a timeout will occur and Alice selects a new channel and repeats the process. Alice must generate a different key material, which we refer to as a pre-key, for every transmission attempt. If Alice receives an ACK, she knows that this pre-key will be used, otherwise she discards the pre-key.

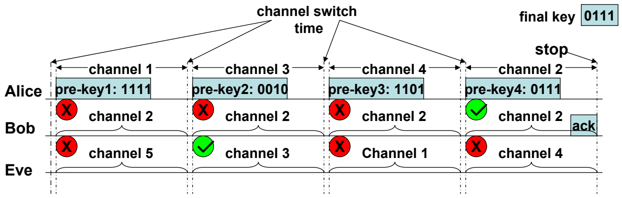

Fig. 1. Illustration of packet based scheme

While Bob could also select a new channel for each attempt, this would require Alice and Bob to maintain synchronized timers. Since Eve has no knowledge of which channel Bob is on, Bob remaining on the same channel will not affect the security of the scheme.

Fig. 1 illustrates this scheme. Alice sends Bob several 4-bit pre-keys through different channels. Bob successfully receives the last pre-key and sends an ACK. Even though Eve happens to overhear the second pre-key, it will not be of any use since the last pre-key is different.

Analysis. The key insight motivating this scheme is that the probability of an eavesdropper Eve being presented on the channel where Alice and Bob meet is smaller than the probability of Alice and Bob selecting the same channel. Given n channels, the probability that Alice and Bob meet on the same channel and exchange key material is $p = \frac{1}{n}$. Assume that Eve does not possess more sophisticated radios than Alice and Bob and can only monitor one channel at a time. Then, Eve can only randomly select one channel to listen to and the probability that all three select the same channel is $\frac{1}{n^2}$, and the probability of Eve overhearing the secret key is $p_e = \frac{1}{n}$.

To reach a high degree of security, this protocol requires a very large number of channels. This is impractical because the number of available channels is often limited by the radio hardware and the time required for a successful key exchange increases with the number of channels. In fact, the probability that Alice and Bob successfully exchange a secret key in x attempts follows the Geometric distribution $P_X(x) = p(1-p)^{x-1}$. Thus the expected number of exchange attempts is $E[X] = \frac{1}{p} = n$, where n is the number of channels. This means that halving the probability of key overhearing p_e will require twice the number of channels and double the time required for key agreement.

B. Multi-Agreement Scheme

To address the limitation, we introduce a multi-agreement scheme. In this scheme, Bob and Alice will repeat the pre-key agreement multiple times. The process will end when Bob receives k pre-keys and the final secret key will be a XOR of all the pre-keys.

Analysis. By using k multiple pre-keys, the probability function for Alice and Bob to create a secret key in x packet exchange attempts changes from the Geometric distribution to a Pascal distribution: $P_X(x) = \binom{x-1}{k-1} p^k (1-p)^{x-k}$ and the expected number of packet exchange attempts becomes $E[X] = \frac{k}{p} = kn$ where $p = \frac{1}{n}$. The probability for Eve to obtain knowledge of final secret key decreases to the

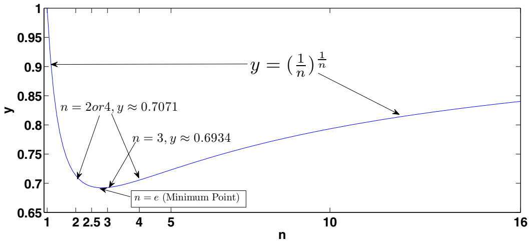

Fig. 2. Curve of function $y = (\frac{1}{n})^{\frac{1}{n}}$

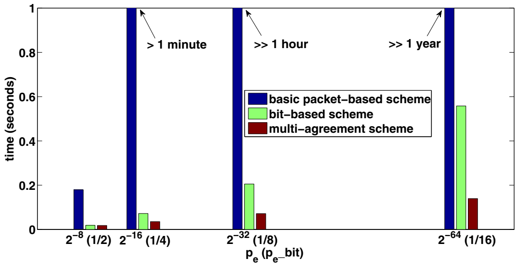

Fig. 3. Time cost to generate a secret key for different key agreement protocols

$p_e = (\frac{1}{n})^k$. Note that now doubling the value of k doubles the expected time to create a secret key, however, the value of p_e is squared. This allows us to achieve a higher degree of security in less time.

More properties of this multi-agreement scheme are discussed in detail as below:

1) Optimal Number of Channels: Given that a security level p_e can be reached by increasing the number of channels n or the number of rounds k, what number of channels minimizes the expected number of packet exchange attempts? This is equivalent to the problem of minimizing p_e when given fixed $E[X]$.

Since $E[X] = kn$, $p_e = (\frac{1}{n})^k = (\frac{1}{n})^{\frac{E[X]}{n}}$. Therefore, obtaining the minimum value of p_e is equal to obtaining the minimum value of $y = (\frac{1}{n})^{\frac{1}{n}}$ for $n \geq 1$ when $E[X]$ is fixed. As shown in Fig. 2, y has a minimum value when $n = e$. When $n \geq e$, y is monotonously increasing. Since the number of channels n has to be an integer, p_e reaches its minimum at $n = 3$ (We tested that it has a smaller y value than $n = 2$).

Next we provide a more general proof for the optimal value of channel n with a new parameter d which indicates the number of channels Eve can listen to simultaneously. Since Eve can listen to d number of channels at the same time, the probability of Eve overhearing a pre-key changes to $p_e = (\frac{d}{n})^k$. Assume $x = \frac{1}{n}$, we have

$$\begin{aligned} y(n) &= (\frac{d}{n})^{\frac{1}{n}} \\ &= (dx)^x \\ lny &= xlndx \\ \frac{1}{y}y' &= 1 + lndx \\ y' &= (dx)^x(1 + lndx) \\ &= (\frac{d}{n})^{\frac{1}{n}}(1 + ln(\frac{d}{n})) \end{aligned}$$

Since $n = de \Rightarrow y' = 0$, for all $n > de$, $y' > 0$ and $y(de) < y(\infty) = 1$, the y curve monotonously increases at the right side of de and converges to 1. On the other side, for all $n < de$, $y' < 0$ and $y(de) < y(1) = d < y(0) = \infty$, so the y curve monotonously decreases at the left side of de and after passing the point $y(1) = d$, it diverges. Since n must be an integer, the optimal value is $n = \lceil de \rceil$ or $n = \lfloor de \rfloor$.

Now, given d and p_e, the expected shortest time to create a secret key in terms of the minimum value of packet exchange attempts can be calculated. For example, when $d = 20$ and $p_e = 2^{-128}$, we have $n = 55$ and $k = \log_{p_e} \frac{d}{n} = 88$, and $E[X] = nk = 4840$. For an IEEE 802.11 device, using a secret key length of 128 bits, each packet exchange attempt can be finished in 0.0012 seconds[1]. Therefore, on 55 channels with 88 rounds, Alice and Bob can create a secret key in less than 6 seconds.

2) Storage Concern: Since the only useful pre-keys are those received by Bob, it would be a waste of space if Alice stored all pre-keys sent. To reduce the storage space consumption, Alice could only store those pre-keys which are ACK'ed by Bob. In this way, the storage requirement is only equal to the number of agreements. The storage requirement can be further reduced to one by XOR'ing ACK'ed pre-keys on the fly.

IV. Performance Evaluation

Speed is one of the critical factors to determine if a key agreement protocol is practical. Thus, we compare the key generation time in an IEEE 802.11 network using the NS2 simulator. We also include a baseline bit-based scheme, in which single bit is transferred each time and in the end, Bob concatenates all the received bits into a secret key. In Fig. 3, the probability that Eve obtains all the information about the final secret key, p_e, is plotted. However for the bit based scheme, since each packet exchange attempt only carries one bit, the adversary may overhear parts of a key. For this reason, we use p_e_bit for the probability of Eve obtaining any bit of the final key. Lower p_e_bit is obtained through multi-agreement on single bit. In the figure, as p_e (or p_e_bit) varies from 2^{-8} to 2^{-64} (1/2 to 1/16), the secret key size varies from 8 to 64 bits. It can be seen that the basic packet based scheme is impractical in terms of the time to complete and among all three schemes, the multi-agreement scheme spends the least time to generate a secret key.

[1] According to the IEEE 802.11 FHSS specification, the hop time is $224\mu s$, the overhead for preamble and PLCP header are $96\mu s$ and $32\mu s$. So the time to transmit a 128 bits data packet can be calculated as $224\mu s + 96\mu s + 32\mu s + \frac{128+34*8+28*8}{1.0Mbit/s} \approx 0.000947s$, where we assume data rate is 1.0Mbit/s and 34 and 28 bytes are MAC and UDP/IP header overhead. The ACK time can be calculated as $96\mu s + 32\mu s + \frac{14*8}{1.0Mbit/s} \approx 0.000235s$. The total time including the time duration for SIFs is about $0.000947s + 0.000235s + 0.000028s \approx 0.0012s$

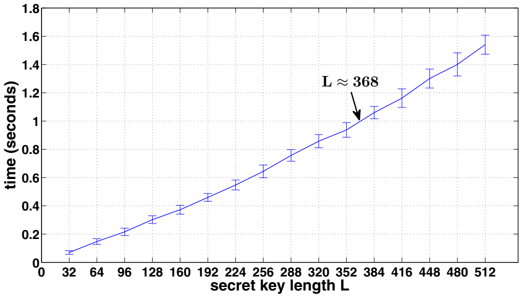

Fig. 4. Time cost to generate a secret key for different key length L with $p_e = 2^{-L}$

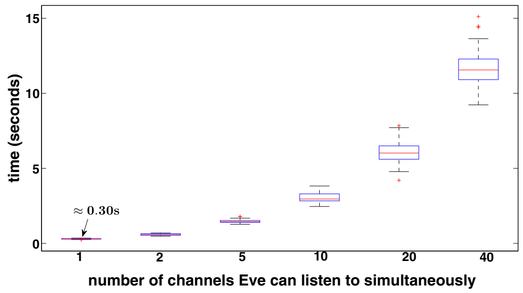

Fig. 5. Optimal number of channels, $p_e = 2^{-128}$, key length 128 bits

The runtime of the bit-based and basic packet-based schemes are very large for small values of p_e (high security requirements). Therefore we only present the following results for the multi-agreement scheme with lower p_e values.

We study the key generation time for different key lengths L with $p_e = 2^{-L}$. It can be seen from Fig. 4 that the proposed scheme approximately generates security levels up to 368-bit keys per second. This is much faster than the prior technique that extracts key material from the wireless channel with a rate of 1 bit per second [3].

Fig. 5 shows the key generation time for a 128-bit key with $p_e = 2^{-128}$, when Eve can listen to multiple number of channels simultaneously. As shown in the figure, it takes only about 0.3 second to create a secret key when Eve has no superior power than a regular user. As the number of channels Eve can listen to (d) increases, it takes longer to create a key. However, even with 40 channels Eve can listen to, the key generation takes less than 15 seconds. For both $d = 1$ and $d = 40$, the scheme only needs to store about 85 pre-keys.

In Fig. 6, we study the key generation time for a 128-bit key with a relaxed requirement for p_e. The higher the p_e is, the faster a secret key can be generated. The difference of key generation time becomes more obvious when Eve can listen to a higher number of channels at the same time. This implies that for some cases, we could allow a higher p_e to achieve a faster key agreement[2].

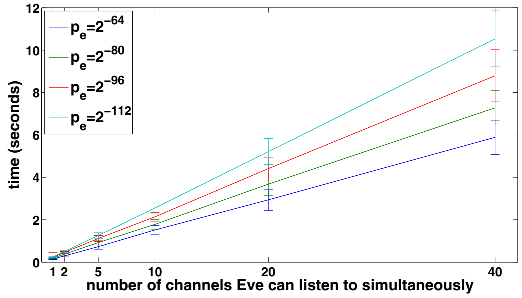

Fig. 6. Time cost to generate a secret key for different p_e and d

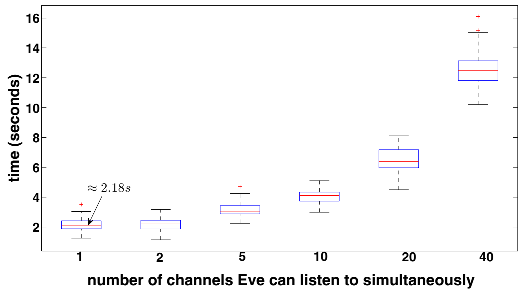

Fig. 7. 79 channels, $p_e = 2^{-128}$, key length 128 bits

In Fig. 7, Alice and Bob always use 79 channels[3] to execute the key agreement protocol. A 128-bit key can still be created in about 2 seconds for regular users, and when Eve can listen up to 40 channels at the same time, the key can be generated in 16 seconds.

V. Discussion

A. Optimal Size of Pre-key

The analysis in Section III is based on the assumption that Eve cannot decode a pre-key if she doesn't receive the packet from the beginning of its transmission. In real world, even when Eve doesn't overhear the packet completely, she still can get some information from the incomplete data. Thus the longer the pre-key transmission takes, the higher the possibility that the pre-key will be exposed. In such a case, shorter pre-keys may be preferred and several pre-keys can be merged to form intermediate keys first and then all intermediate keys are combined (XOR) together to create the final secret key. Fig. 8 illustrates a specific example of this method.

The choice of short pre-keys at high frequency hopping rate vs. long pre-keys at low frequency hopping rate depends on many factors. Assume that the probability that Eve can eavesdrop a pre-key is a function of three parameters: pre-key length, number of frequency hopping channels and number of channels Eve can listen to simultaneously, $f(l, n, d)$. The time cost for a packet exchange attempt is a function of pre-key length: $t(l)$. For a fixed time period $T = E[X]{*}t(l) = kn{*}t(l)$

[2] If assume key exchange happens every one hour and $p_e = 2^{-20}$, Eve may successfully overhear all the necessary keys once in more than 100 years!

[3] According to the IEEE 802.11 FHSS specification, North America can use a maximum of 79 channels.

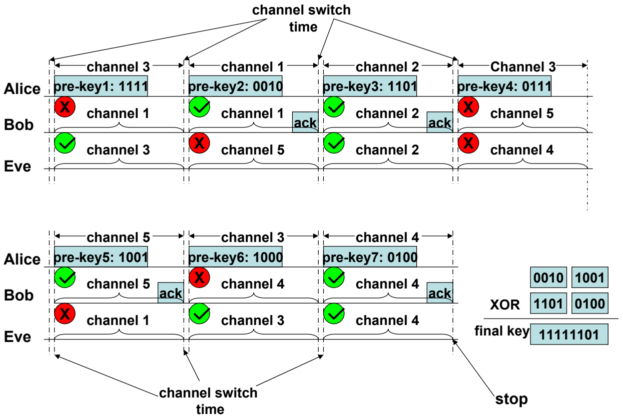

Fig. 8. Illustration of optimal size of pre-key

and $p_e = (f(l,n,d))^k = (f(l,n,d))^{\frac{T}{nt(l)}}$, if d is also fixed, the optimal value of l can be decided by changing the value of n.

B. Active Attacks and Interference

We consider three active attack scenarios. In the first scenario, when Eve overhears a pre-key which was not received by Bob, she might pretend to be Bob and send back an ACK. This will not help Eve to get the final secret key. First, we assumed that Eve cannot monitor all channels. Hence, once Bob receives a pre-key from Alice while Eve is not at that channel, then Eve is guaranteed to fail to get the secret key. Second, it is easily detectable by Bob, since Alice will finish the key agreement procedure before Bob receives enough pre-keys. In the second scenario, Eve pretends to be Alice and sends Bob some fake pre-keys. First, Bob still can hear some legitimate pre-keys from Alice that are not received by Eve. Then the final secret keys generated at each party will be different. Second, Alice will detect the problem sooner or later since she would be waiting for more ACKs. In the third scenario, Eve pretends to be both Alice and Bob, but it still cannot completely prevent Alice and Bob from exchanging a pre-key on a channel where she was not listening.

Our current work is based on an assumption that there is no node competition on any channel. This assumption may be impractical for some cases, and it remains to be among our current work. However, a possible solution would be to use a common channel to exchange channel usage information in order to reduce interference during the period of key agreement.

VI. Conclusion

In this work, we have presented a secret key agreement protocol using random channel hopping. The two parties randomly select channels and exchange key material whenever they meet on the same channel. Using many rounds of agreement, the probability that an eavesdropper observes the key becomes negligible. Both analysis and simulations showed that this method can establish keys faster than other wireless key extraction methods and is secure as long as the adversary can only monitor a few channels simultaneously. The results also show that the multi-agreement scheme outperforms the basic packet-based scheme and can establish a 128-bit secret key in 0.3 seconds.

References

[1] "Wi-fi protected access." [Online]. Available: http://en.wikipedia.org/wiki/Wi-Fi_Protected_Access

[2] L. Eschenauer and V. D. Gligor, "A key-management scheme for distributed sensor networks," in *ACM Conference on Computer and communication Security(CCS)*, 2004.

[3] S. Mathur, W. Trappe, N. Mandayam, C. Ye, and A. Reznik, "Radio-telepathy: extracting a secret key from an unauthenticated wireless channel," in *MobiCom '08: Proceedings of the 14th ACM international conference on Mobile computing and networking*. New York, NY, USA: ACM, 2008, pp. 128–139.

[4] J. Steiner and J. I. Schiller, "An authentication service for open network systems," in *Usenix Conference Proceedings*, 1988, pp. 191–202.

[5] D. Otway and O. Rees, "Efficient and timely mutual authentication," *SIGOPS Oper. Syst. Rev.*, vol. 21, no. 1, pp. 8–10, 1987.

[6] W. Diffie and M. E. Hellman, "New directions in cryptography," *IEEE Transactions on Information Theory*, vol. IT-22, no. 6, pp. 644–654, 1976. [Online]. Available: citeseer.ist.psu.edu/diffie76new.html

[7] (2009) Discrete logarithm. [Online]. Available: http://en.wikipedia.org/wiki/Discrete_logarithm_problem

[8] R. C. Merkle, "Secure communications over insecure channels," *Commun. ACM*, vol. 21, no. 4, pp. 294–299, 1978.

[9] R. Blom, "An optimal class of symmetric key generation systems," in *Proc. of the EUROCRYPT 84 workshop on Advances in cryptology: theory and application of cryptographic techniques*. New York, NY, USA: Springer-Verlag New York, Inc., 1985, pp. 335–338.

[10] F. J. MacWilliams and N. Sloane, *Theory of Error-correcting Codes*. North-Holland, 1977.

[11] L. Eschenauer and V. D. Gligor, "A key-management scheme for distributed sensor networks," in *CCS '02: Proceedings of the 9th ACM conference on Computer and communications security*. New York, NY, USA: ACM Press, 2002, pp. 41–47. [Online]. Available: http://dx.doi.org/10.1145/586110.586117

[12] H. Chan, A. Perrig, and D. Song, "Random key predistribution schemes for sensor networks," in *SP '03: Proceedings of the 2003 IEEE Symposium on Security and Privacy*. Washington, DC, USA: IEEE Computer Society, 2003, p. 197.

[13] A. C.-F. Chan, "Distributed symmetric key management for mobile ad hoc networks," *IEEE INFOCOM*, vol. 4, pp. 2414–2424, 2004.

[14] R. Rivest, L. Adleman, and M. Dertouzos, "On data banks and privacy homomorphisms." Academic Press, 1978, pp. 169–177.

[15] P. Erdos and A. Renyi, "On the evolution of random graphs," *Publ. Math. Inst. Hung. Acad. Sci*, vol. 5, pp. 17–61, 1960.

[16] C. Blundo, L. A. F. Mattos, and D. R. Stinson, "Trade-offs between communication and storage in unconditionally secure schemes for broadcast encryption and interactive key distribution," in *CRYPTO*, 1996, pp. 387–400.

[17] J. Hershey, A. Hassan, and R. Yarlagadda, "Unconventional cryptographic keying variable management," *Communications, IEEE Transactions on*, vol. 43, no. 1, pp. 3–6, Jan 1995.

[18] A. Kitaura, T. Sumi, K. Tachibana, H. Iwai, and H. Sasaoka, "A scheme of private key agreement based on delay profiles in uwb systems," March 2006, pp. 1–6.

[19] T. Aono, K. Higuchi, T. Ohira, B. Komiyama, and H. Sasaoka, "Wireless secret key generation exploiting reactance-domain scalar response of multipath fading channels," *Antennas and Propagation, IEEE Transactions on*, vol. 53, no. 11, pp. 3776–3784, Nov. 2005.

[20] M. Strasser, C. Pöpper, S. Capkun, and M. Cagalj, "Jamming-resistant key establishment using uncoordinated frequency hopping," in *SP '08: Proceedings of the 2008 IEEE Symposium on Security and Privacy*. Washington, DC, USA: IEEE Computer Society, 2008, pp. 64–78.

[21] M. J. Miller and N. H. Vaidya, "Leveraging channel diversity for key establishment in wireless sensor networks," April 2006, pp. 1–12.