Research Directions in Wireless Security

Iris Safaka
I&C, EPFL

***Abstract*—We consider the problem of generating a shared secret key S between two parties over wireless channels, in the presence of an eavesdropping adversary. We focus on information theoretic approaches that, in contrast to existing cryptographic solutions, aim to solve this problem without making any assumptions about the adversary's computational capabilities. We first present information theoretic bounds on achievable pairwise secrecy over broadcast channels with public discussion [8]. Next, we describe two frameworks, that propose different approaches towards building an information theoretically secure practical system [15], [5]. Finally, we present preliminary results of a practical framework that builds on computationally efficient techniques from network coding.**

***Index Terms*—secret key agreement, information theoretic security, secrecy, wireless systems security**

I. Introduction

THE secret key generation over wireless channels in the presence of a passive adversary is a well studied problem, and nowadays it can be solved by using asymmetric key cryptography, like RSA [10] or Diffie-Hellman [4] algorithms. These approaches rely on the adversary's limited computational capabilities, such as prime number generation and factorization, to guarantee that it is computationally infeasible for her to derive the shared secret key.

Proposal submitted to committee: August 21st, 2012; Candidacy exam date: August 28th, 20012; Candidacy exam committee: Serge Vaudenay (exam president), Christina Fragouli (thesis director), Katerina Argyraki (co-examiner).

This research plan has been approved:

Date: _____

Doctoral candidate: _____
(name and signature)

Thesis director: _____
(name and signature)

Thesis co-director: _____
(if applicable) (name and signature)

Doct. prog. director: _____
(R. Urbanke) (signature)

EDIC-ru/05.05.2009

On the other hand, information theoretic security, or unconditional security, does not make any assumptions about the adversary's computing power, but it rather builds on her lack of information that is essential to generate the secret key herself. Wireless networks serve as a good starting point to explain the idea, due to their noisy nature. Most real wireless communication channels are noisy, and it is only for applications that are converted into virtually error-free channels by the use of error-correcting codes. In a wireless setup when an honest node, Alice, broadcasts a message, then it is unlikely that an other honest node, Bob, and an enemy, Eve, will both overhear exactly the same information, given that there exists sufficient noise in the channel. This information mismatch can be exploited for the sake of secret key generation.

In practice though, information theoretic security is not still used in modern security systems. Although there already exist theoretical results on the feasibility of the honest parties to establish a shared secret secure from the enemy (*eg.* [3], [8], [14]), and different efficient schemes for achievability (*eg.* [7], [1]), there is a lack of practical system implementations on actual wireless networks. A growing interest on this systems domain has started only recently, with the appearance of techniques that leverage different characteristics of the wireless physical channel [6], [16], [13].

We first present information theoretic bounds on achievable pairwise secrecy over broadcast channels [8]. As mentioned, the noisy nature of the wireless channel may result in non identical observations between receivers. Some natural question arise: What is the theoretical highest possible rate of secret key generation in such a setup and under which theoretical network conditions does this hold? What is an achievability scheme? *etc.* Next, we describe two approaches towards building practical information theoretically secure systems. The first one [15] relies on the limited ability of Eve to intercept every communication between Alice-Bob and on the link layer retransmission mechanism in order to create a maybe not so perfectly secret key. By XOR-ing many of these though, the key becomes more secure in depth of time . The second one [5] is a physical layer technique that applies *jamming* in order to prevent Eve from getting information not intended for her. The proposed solution includes a customized IEEE 802.11 PHY.

Finally, we describe a secret key agreement protocol. Our protocol is provably information theoretically secure and involves only polynomial time operations. In addition, we discuss the techniques for adapting our protocol to real wireless networks and we present some initial experimental results. We use commodity network devices without any modification to the standard 802.11 MAC and PHY layers.

II. Secret Key Agreement by Public Discussion from Common Information

The work of Maurer established the theoretical bounds of secrecy capacity over noisy broadcast channels with public discussion [8]. Interestingly, by allowing Alice and Bob to exchange feedback over a public channel (assumed to be error-free without loss of generality) in the presence of Eve, the secrecy rate obtains a non-zero value, even if the channel between Alice-Eve is better than the Alice-Bob channel.

A. The discrete memoryless broadcast channel case

We consider a discrete memoryless broadcast channel with input variable X, chosen by Alice according to some distribution P_X, and output variables Y and Z received by the legitimate receiver Bob and the adversary Eve respectively. Variables X, Y, Z take their values from some finite alphabets $\mathcal{X}, \mathcal{Y}, \mathcal{Z}$. The conditional probability distribution $P_{YZ|X}$ defines the channel behavior. The following definition is formally stated in [3].

Definition II.1. The secrecy capacity $C_s(P_{YZ|X})$ of the described broadcast channel is the maximum rate (in number of bits per channel use) at which Alice can reliably send information to Bob such that the rate at which Eve obtains this information is arbitrarily small.

In analogy, one can define the *secrecy capacity with public discussion* as follows.

Definition II.2. The secrecy capacity with public discussion $\hat{C}_s(P_{YZ|X})$ is the maximum rate (in number of bits per channel use) at which Alice and Bob can agree on a secret key by exchanging arbitrary messages over a public channel, such that the rate at which Eve obtains information about the key by observing the public messages and the Z-outputs is arbitrarily small.

For the channel described above, in [3] it is proved that $C_s(P_{YZ|X}) \geq \max_{P_X}[I(X;Y) - I(X;Z)] = \max_{P_X}[H(X|Y) - H(X|Y)]$. Applying this result fro the case when the previously described channel consists of two independent binary symmetric channels, one from Alice to Bob and one from Alice to Eve, with bit error probabilities ϵ and δ respectively, the next lemma and proposition hold.

Lemma II.1. The secrecy capacity of the described binary broadcast channel is

$$C_s(P_{YZ|X}) = \begin{cases} h(\delta) - h(\epsilon), & \text{if } \delta > \epsilon \\ 0, & \text{otherwise} \end{cases}$$

Proposition II.1. The secrecy capacity with public discussion of the described binary broadcast channel is

$$\hat{C}_s(P_{YZ|X}) = h(\epsilon + \delta - 2\epsilon\delta) - h(\epsilon).$$

Moreover, $\hat{C}_s(P_{YZ|X}) \geq 0$, with equality if and only if $\epsilon = 0.5, \delta = 0$ or $\delta = 1$, i.e., if X, Y are statistically independent or if Z uniquely determines X.

Lemma II.1 implies that the secrecy capacity is zero when the quality of Eve's channel is better than Bob's, namely when $\delta \leq \epsilon$. In contrast, by allowing feedback from Bob and Alice over the insecure public channel, the secrecy rate is strictly positive even if $\delta \leq \epsilon$, given that Alice and Bob know an upper bound on δ.

The intuition behind the proof of Proposition II.1 is that the legitimate nodes use their common information to publicly discuss in such a way that they eliminate the adversary's advantage ($\delta \leq \epsilon$). The notion of the "conceptual" noisy broadcast channel is introduced at that point. Here we omit the whole proof, due to space limitations.

B. Upper and lower bounds on secret key rate

Consider a secret key agreement protocol, where Alice and Bob are assumed to know P_{XYZ}, as follows. At each step of the protocol either Alice or Bob sends a message over the public channel. During step i, where i is odd, Alice sends a message C_i to Bob that is a function of X and all the messages previously received from Bob, i.e. C^{i-1}. For i even, the roles are exchanged (and also X is replaced by Y). At the end of the t-step protocol, Alice computes a key S (resp. Bob computes S') as a function of X (resp. Y) and $C^t \triangleq [C_1, \ldots, C_t]$. Their goal is first to agree on the same value of S and S' with very high probability , and second keep Eve's uncertainty about the constructed key close to the maximum, given that she has knowledge of Z and also she has been tracking all the public discussion, i.e., she knows C^t. The following equations summarize formally the above.

$$H(C_i|C^{i-1}X) = 0, \quad (1)$$
$$H(C_i|C^{i-1}Y) = 0, \quad (2)$$
$$H(S|C^tX) = 0 \quad (3)$$
$$H(S'|C^tY) = 0, \quad (4)$$
$$P[S \neq S'] \leq \alpha, \quad (5)$$
$$I(S; C^tZ) \leq \beta \quad (6)$$

for some α, β small. Theorem II.1 expresses an upper bound on the uncertainty of the established key S.

Theorem II.1. For any key agreement protocol satisfying (1)-(4),

$$H(S) \leq I(X;Y|Z) + H(S|S') + I(S; C^tZ).$$

If we assume that S, S' are identical and also perfectly secure from Eve, namely we let $P[S \neq S'] = 0$ and $I(S; C^tZ) = 0$, then what Theorem II.1 essentially says is what intuition suggests: the quality of the secret key depends on the amount of the common information that Alice and Bob share, given that Eve knows the jointly distributed variable Z.

For the general case where $\alpha, \beta \neq 0$ and small, the following corollary holds as an immediate consequence of Theorem II.1.

Corollary II.1. For every key agreement protocol satisfying (1)-(6),

$$H(S) \leq \min[I(X;Y), I(X;Y|Z)] + \\ +\alpha + h(\beta) + \beta \log_2(|\mathcal{S}| - 1).$$

A natural assumption to make is that the random experiment generating XYZ is repeated independently N times: Alice, Bob, Eve receive $X^N = [X_1, \ldots, X_N], Y^N =$

$[Y_1,\ldots,Y_N], Z^N = [Z_1,\ldots,Z_N]$ respectively, where $P_{X^N,Y^N,Z^N} = \prod_{i=1}^N P_{X_i,Y_i,Z_i}$ and $P_{X_i,Y_i,Z_i} = P_{X,Y,Z}$ for $1 \leq i \leq N$. For any secret key agreement protocol satisfying equations (1)-(6) (replacing X, Y by X^N, Y^N and requiring that $\frac{1}{N} I(S; C^t Z^N) \leq \beta$ for N sufficient large) the quantity *secrecy key rate* is defined and its bounds are derived.

Definition II.3. The secrecy key rate $S(X;Y||Z)$ is the maximum rate R (in number of bits per symbol per channel use) at which Alice and Bob can agree on a secret key S while keeping the rate at which Eve obtains information about the key arbitrarily small, i.e., for every $\beta > 0$ there exists a protocol achieving $\frac{1}{N} H(S) \geq R - \beta$.

Theorem II.2. The secret key rate of X and Y with respect to Z is upper bounded by

$$S(X;Y||Z) \leq \min[I(X;Y), I(X;Y|Z)].$$

Theorem II.3. The secret key rate of X and Y with respect to Z is lower bounded by

$$S(X;Y||Z) \geq \max[I(X;Y) - I(X;Z), I(Y;X) - I(Y;Z)].$$

The upper bound stated in Theorem II.2 follows from Corollary II.1. The lower bound in Theorem II.3 shows that if Eve has either less information about Y than Alice or less information about X than Bob, then such a difference of information can be exploited in benefit of the secrecy rate. In particular, the proof of Theorem II.3 builds on the fact that both $I(X;Y) - I(X;Z)$ and $I(Y;X) - I(Y;Z)$ are achievable secret key rates when Alice and Bob publicly discuss over conceptual broadcast noisy channels as discussed in Section II-A.

Throughout this section we assume that X, Y, Z are jointly distributed according to P_{XYZ}. The broadcast channel described in Section II-A is just a generalization of the key agreement scenario described above, since we let Alice choose P_X given the channel transition probabilities $P_{XY|Z}$. Hence, the secrecy capacity with public discussion $\hat{C}_s(P_{YZ|X})$ can be defined accordingly to the secrecy rate $S(X;Y||Z)$ by allowing Alice to send consecutively the binary symbols $X_1,\ldots,X_N$. Finally, let Alice choose P_X, where P_X maximizes $S(X;Y||Z)$.

Theorem II.4. The secrecy capacity with public discussion of a broadcast channel specified by $P_{YZ|X}$ is bounded from below and from above by

$$\begin{aligned}\max_{P_X} S(X,Y||Z) &\leq \hat{C}_s(P_{YZ|X}) \\ &\leq \min[\max_{P_X} I(X;Y), \max_{P_X} I(X;Y|Z)].\end{aligned}$$

C. Secret agreement under realistic conditions

Consider a satellite generating and broadcasting random bits B ($P_B(0) = P_B(1) = 1/2$) at a significantly low SNR, such that an enemy Eve cannot receive without any error probability, regardless her hardware equipment. Such a scenario corresponds to X, Y, Z being symmetrically distributed with respect to P_{XYZ}. It can be shown that for this kind of probability distributions, it equivalent to case when X, Z, Y are generated by three independent BSCs with error probabilities $\epsilon_A, \epsilon_B, \epsilon_E$ respectively, where the input of the channels is B.

Using Theorem II.3, it can be shown that for X, Y, Z generated as above it holds that

$$\begin{aligned}S(X,Y||Z) \geq \max[&h(\epsilon_A + \epsilon_E - 2\epsilon_A\epsilon_E) + \\ &+ h(\epsilon_A + \epsilon_B - 2\epsilon_A\epsilon_B)] - h(\epsilon_A + \epsilon_B - 2\epsilon_A\epsilon_B).\end{aligned}$$

Therefore, the secrecy rate is zero if $\epsilon_E < \epsilon_A$ and $\epsilon_E < \epsilon_B$, i.e., if Eve's channel is superior than both Alice's and Bob's. Nevertheless, even under these conditions, secret key agreement with non zero secrecy rate is possible with public discussion.

Let Alice randomly select a codeword V^N from the codebook of an error-correcting code $\mathcal{C}$ of length N. She sends this to Bob (and thus also to Eve) over the conceptual broadcast channel, i.e., by sending $X^N + V^N$ over the public channel. Bob and Eve receive the bits of V^N with error probabilities $\epsilon_A + \epsilon_B - 2\epsilon_A\epsilon_B$ and $\epsilon_A + \epsilon_E - 2\epsilon_A\epsilon_E$ respectively, where the latter is smaller than the former unless $\epsilon_E \geq \epsilon_B$. The key observation here is that although Eve receives codewords V^N more reliably than Bob on the average, her conceptual channel may be worse than Bob's channel if we apply a specific decision rule at Bob: Bob will accept a received word only if he can make a very reliable decision, i.e., if it is very close to some codeword in $\mathcal{C}$. Namely, if one averages only over the instances correctly received by Bob, then the advantage of the adversary is canceled out. By adding modulo 2 many of these codewords, Alice and Bob can decide on a common key while keeping Eve's information about it arbitrarily small.

III. Secure Wireless Communication with Dynamic Secrets

In this section we present a framework [15] that builds on the inherent noisy nature of the wireless medium to enable two honest parties to agree on a shared secret key under the presence of a passive adversary. It consists of a set of low-complexity algorithms that operate at the link layer level and builds on Maurer's framework of public discussion [8] and privacy amplification [1].

A. System and adversary model

The basic goal of this framework is to secure wireless communications at time periods when the underlying secret of the cryptographic mechanism of the honest parties (symmetric key or public key infrastructure) has been revealed to an enemy (not by computational effort but just by directly accessing it somehow) and thus she can decrypt every communication. The idea is to provide an *additional* security mechanism on the existing cryptographic schemes rather than an alternative to traditional computational cryptography.

Unlike other approaches that focus on the physical layer and attempt to extract secrets by exploiting the channel's properties, the approach described here focuses on the layer above, the data link layer, and exploits frame retransmissions that happen between Alice and Bob. When a frame is not retransmitted, it means that it has been sent over the air only

once and it has been correctly received by the honest node. The idea behind this motivation is that when we consider only frames correctly received form Bob and aired once, it is *likely* that an adversary has missed some of them. This loss of information can be exploited by Bob and Alice to produce in short time intervals bit sequences (referred to as dynamic secrets), some of which are safe (or partially safe) from Eve, and XOR these consecutively with the compromised secret key in order to make it secret again from Eve.

The assumption on the adversary made above along with some others define the adversary model considered in this framework. First, as already stated, the Alice-Eve channel is assumed to be of worse quality than that of Alice-Bob for at least some non zero period in time. This also implies that Eve does not posses special equipment (directional antennae, multiple antennae in distributed space etc.) that would allow her to ultimately have a better channel than Bob regardless her relative position to Alice. It is also implied, that Eve cannot be physically present in an arbitrarily small distance from Bob or Alice. Second, Eve's hardware is equivalent to that of the honest nodes and she can either receive correctly or partially correctly a MAC frame delivered from her custom PHY layer. She does not apply any optimal guessing strategy for the bits the PHY reports as corrupted. Third, Eve is only a passive eavesdropper and she does not collude with other eavesdroppers with higher capabilities and in such a way that would potentially make her receive correctly the exact same frames that Bob also received correctly. Finally, as already briefly discussed, the compromised key is XORed with a newly generated dynamic secret by the moment it is produced. Eve does not apply any strategy to detect modifications by this procedure on the system's key and try to cancel them out.

B. Extracting dynamic secrets

An Automatic Error Tracing (AET) algorithm, based on a Stop-and-Wait data link layer retransmission protocol, is used to monitor the link layer error retransmission process at both the sender and the receiver. The goal is to identify the "one time frames" (OTFs), namely the frames that are transmitted only once by Alice and correctly received by Bob.

The algorithms for Alice (the sender) and Bob (the receiver) are defined in Algorithm 1 and Algorithm 2 respectively. In each frame, there is a retransmission flag and sequence number, denoted by the postfix $.retran$ and $.serial$ respectively. Ψ_s and Ψ_r are the sender and receiver sets, which are empty before the protocol starts, and they contain a certain number of frames N at the end (depending on how often one would like to produce a dynamic secret, the value of N is selected accordingly). The two algorithms essentially enable the two parties to agree on a set of common frames $\Psi = \Psi_s = \Psi_r$ that they both possess correctly. This technique can be seen in analogy to Maurer's framework, where Alice and Bob publicly agree on a set of commonly correctly received codewords.

Once the set Ψ has been identified by Alice and Bob, the next step is to exploit any lack of information about this set that Eve might have. If Eve has perfectly overheard every frame and acknowledgment between Alice and Bob,

Algorithm 1: AET sender

```
foreach frame m_i do
    m_i.retran = 0;
    send m_i;
    while true do
        wait on ACK or time out;
        if ACK received then
            Jump out the loop;
        m_i.retran = 1;
        send m_i;
    if m_i.retran = 0 then
        add m_i to Ψ_s;
```

Algorithm 2: AET receiver

```
foreach received frame m_i do
    if m_i integrity check pass then
        send ACK;
        if m_i.serial ≠ m_{i-1}.serial, m_{i-1}.retran = 0
        then
            add m_{i-1} to Ψ_r;
```

then of course she can herself identify set Ψ, and therefore generating a secret is impossible. If, nevertheless, a weaker adversary model is considered (as the one described in Section III-A), then it is possible that Eve misses some information about Ψ. The next step is to deal with the fact the we do not really know which frames Eve has (maybe partially) lost. The solution is provided by Bennett et al. in [1], where the technique of privacy amplification is used, which depends on the concept of universal hashing introduced in [2].

Definition III.1. A class $\mathcal{G}$ of functions $\mathcal{A} \to \mathcal{B}$ is *universal* if, for any distinct x_1 and x_2 in $\mathcal{A}$, the probability that $g(x_1) = g(x_2)$ is at most $1/|\mathcal{B}|$ when g is chosen at random from $\mathcal{G}$ according to the uniform distribution [2].

Definition III.2. Let X be a random variable with alphabet $\mathcal{X}$ and distribution P_X. The *collision probability* of X is defined as $P_X = \sum_{x \in \mathcal{X}} P_X(x)^2$, i.e., the probability that X takes the same value twice in two independent experiments. The *Rényi entropy* is defined as $R(X) = -\log_2 P_c(X)$ [9].

Theorem III.1. [2] Let X be a random variable with alphabet $\mathcal{X}$, distribution P_X and Rényi entropy R(X). Let G be the choice of a member of a universal class of hash functions $\mathcal{X} \to \{0,1\}^r$, and let $Q = G(X)$. Then

$$H(Q|G) \geq R(Q|G) \geq r - \log_2(1 + 2^{r-R(X)}) \geq r - \frac{2^{r-R(X)}}{\ln 2}.$$

Corollary III.1. [2] Let P_{VW} be an arbitrary probability distribution and let v be a particular value of V observed by Eve. If Eve's Rényi entropy $R(W|V = v)$ is known to be at least c and Alice and Bob choose $S = G(W)$ as their secret

key, where G is chosen at random from a universal class of hash function from W to $\{0,1\}^r$, then

$$H(S|G,V=v) \geq r - \log_2(1+2^{r-c}) \geq r - \frac{2^{r-c}}{\ln 2}.$$

We see from corollary III.1 that when $r < c$ Eve's entropy of the secret key is close to maximal, i.e. the distribution of $p(S|G,V=v)$ is close to uniform. In particular her information about the key S, namely $H(S) - H(S|G,V=v)$, is arbitrarily small. In addition, in [1] it is shown that if the probability that V takes on a value v satisfying $R(W|V=v) \geq c$ is at least $1-\delta$, then we have

$$H(S|G,V) \geq (1-\delta)(r - \log_2(1+2^{r-c})).$$

Let now variable W represent the common information of Alice and Bob in set Ψ, obtained after running the AET algorithms, and let V be the knowledge of Eve on Ψ. Let also Alice and Bob publicly agree on a function g from a universal class of hash functions. Then, according to Corollary III.1, the generated shared secret key S is secure from Eve, given that a lower bound on c is provided, since we can choose r (the length of the secret key) accordingly.

In not very clearly stated though, nor experimentally demonstrated, how this framework guarantees that the value of c is provided for the legitimate nodes or that the probability of V being v is at least $(1-\delta)$ under realistic network conditions. The only experimental result provided is the minimum time for Eve to loose a frame at a random position, but this does not provide any intuition of the practical secrecy rate.

IV. Physical Layer Wireless Security Made Fast and Channel Independent

In this section we discuss iJam [5], a physical-layer approach for pairwise secret key generation under the presence of a passive adversary Eve. Although most state of the art physical layer protocols exploit channel variations to extract secret bits, the one described here follows a different direction by applying the idea of *jamming*, that is, the deliberate injection of artificial noise in the channel. The goal is to ensure that an adversary cannot demodulate a wireless signal not intended for her.

A. System and adversary model

The iJam framework defines a set of physical layer algorithms than enable Alice and Bob to agree on a secret key of B bits. Each honest node is equipped with specialized hardware on top of which the stack of iJam is implemented. The basic idea underlying the framework is the following: the sender repeats its transmission as shown in Figoure 1. For each sample in these repeated transmissions, the receiver randomly jams either the sample in the original transmission or the corresponding sample in the repetition. Since the eavesdropper does not know which signal sample is jammed and which one is clean, she cannot correctly decode the data. In contrast, the legitimate receiver can pick the clean samples from the signal and its repetition, rearrange them to get a clean signal, and then decode. The bits decoded correctly by Bob and not Eve are the commonly shared secret bits between Alice-Bob.

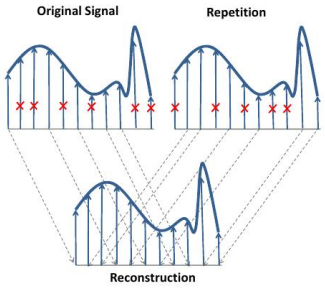

Fig. 1. The sender repeats its transmission. The receiver randomly jams complementary samples in the original signal and its repetition, and stitches together the unjammed samples to create a clean symbol.

At the signal level, the honest nodes use OFDM transmissions, that is, BPSK, 4/16/64 QAM over OFDM, that are typical PHY configurations for the IEEE-802.11 standard. The motivation of using OFDM results not only by the fact it is nowadays widely used in commercial wireless devices, but also for an interesting property: the OFDM signal time samples approximately take values from a zero-mean random Gaussian distribution [12]. If one picks the jamming signal also from a zero-mean random Gaussian distribution, then the combination of the jamming and the original signal will also have Gaussian statistics, namely zero-mean and variance equal to the sum of the two variances. This will incommode Eve in her task of guessing which sample is clean or jammed.

The adversary considered in this framework is passive, static, has hardware equally powerful as that of the legitimate nodes and she could be anywhere in the range of the communicating nodes, i.e. she can listen to all communications in the network. Then, as implied above, Eve applies guessing techniques in an attempt to distinguish jammed samples from clean samples. More precisely, she applies an optimal hypothesis testing strategy: Let S_1, S_2 denote two OFDM samples received by Eve corresponding to two samples from two consecutive transmissions by Alice. Let H_1, H_2 denote the hypothesis that S_1, S_2 is jammed, respectively. A maximum likelihood test would be:

$$H(S_1\ is\ jammed|S_1,S_2)) \overset{H_1}{\gtrless_{H_2}} H(S_2\ is\ jammed|S_1,S_2).$$

Using Bayes' rule, this reduces to:

$$H(S_1,S_2|S_1\ is\ jammed)) \overset{H_1}{\gtrless_{H_2}} H(S_1,S_2|S_2\ is\ jammed).$$

After substituting the Gaussian probabilities the maximum likelihood test reduces to:

$$|S_1|^2 \overset{H_1}{\gtrless_{H_2}} |S_2|^2.$$

Therefore, Eve's best guess is to assume the sample with the smaller magnitude is the clean sample. Eve can apply this test to all the samples and their repetitions to optimally estimate the bits of the Alice-Bob secret key.

A theoretical evaluation of the performance of such an eavesdropper is shown in Figure 2 . It is assumed that Eve can receive the transmitted signal with infinite SNR. The plot in Figure 2 depicts the bit error rate of Eve as a function of the ratio of the jamming power to the sender's power at

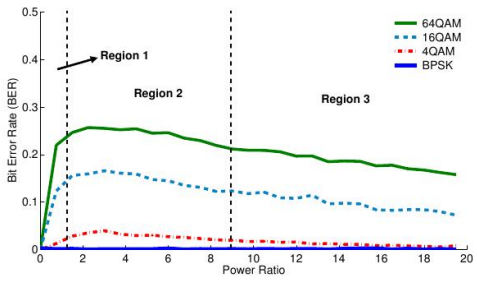

Fig. 2. The figure shows the BER for different modulations as a function of the ratio of the jamming power to the transmitter power at the eavesdropper. The graph can be divided in three regions. Region 1 where the power of the jammer is lower than the transmitter, Region 2 where the power ratio si such that it maximizes the BER, and Region 3 where the power from the jammer is significantly higher than the transmitter.

Eve, for different modulation schemes over OFDM. Ideally the adversary should experiences 50% bit error rate, i.e. she cannot do better than a random guess for every bit of the secret key. Figure 2 reveals that there are scenarios where Eve experiences very low BER. The iJam protocol described next, follows a design philosophy oriented towards solving these issues.

Finally, the adversary could launch an interference cancellation attack, by simultaneously attempting to decode the jamming and the original transmission. Nevertheless, by making the jammer transmit at very high rate (by making the jamming signal samples i.i.d's and using a very dense modulation, for example 65536 QAM), then Eve cannot decode both signals since the total information rate is outside the capacity region [12].

B. Protocol description

Let Alice and Bob want to exchange a secret key of B bits, referred to as *salt*. They perform the following steps:

1) Alice generates L different sequences, each one of which consists of M consecutive salts (of B random bits each)
2) Alice sends the L sequences consecutively and Bob jams each of these sequences with a different power level.
3) Alice and Bob now exchange roles. Bob performs steps (1), (2) while Alice acts as receiver/jammer.
4) The final key is constructed by XOR-ing all the ACKed salts out of the $2ML$ salts totally exchanged.

Each one of the above operations aims to ensure high bit error rate, as much closer as it gets to 50%, to Eve regardless her location and the modulation scheme used for transmission.

(a) Location independence: As already remarked in Figure 2, Eve's BER depends on the power ratio $\frac{P_{j\to e}}{P_{s\to e}}$, the jammer's power level at Eve over the sender's power level at Eve, namely on her location with respect to Alice and Bob. Both scenarios where this ratio is either very low (less than 1), i.e. the jamming power is not enough to jam what Alice sends, or very high (more than 9), i.e. the jamming power is too high so Eve can identify the jammed samples with high probability, are problematic. The problem is that for a given jamming power there are eavesdropping locations for which this ratio is either very low or very high. By making the jammer use L different power levels, so as to cover the whole range from maximum hardware supported power to noise power, and by repeating the protocol while exchanging roles of Alice and Bob, iJam attempts to solve this issue. With this technique, there exists at least one salt for which the power ration is neither too high nor too low.

(b) BER amplification: Figure 2 shows that even if the power ratio is in Region 2 still Eve's BER is not that close to the ideal 50%. For this reason at each power level L there is a sequence of M salts sent and not only one: say the BER in each of the individual salt is p, the probability of the i^{th} bit not being corrupted in all M salts decreases exponentially with M as $(1-p)^M$. By choosing M large enough the probability that Eve knows exactly the value of a bit, after XOR-ing all these salts, gets very small.

C. Experimental evaluation

The iJam protocol has been implemented on USRP2 nodes on top of GNU Radio software and was evaluated in an indoor testbed, consisting of 20 nodes. One should remark that the nodes are not extremely closely spaced and most of them do not have pairwise line-of-sight. Some interesting outcomes are the following:

1) For all modulation schemes, the BER, of an adversary applying the optimal hypothesis testing, versus the power ratio $\frac{P_{j\to e}}{P_{s\to e}}$ follows the simulated behavior of Figure 2. Also for smaller values of SNR (not infinite as implied in Figure 2) the BER is even higher.
2) The BER amplification is possible for all modulations. The number of salts M per power level, needed to reach a 50% BER, can be empirically defined. It is not explicitly stated, though ,if these values can be reused in future setups.
3) For every modulation scheme and for all adversary locations in the testbed, aggregated results show that the BER experienced experimentally is between 40% and 60%, with median of 50%, showing that half of the adversaries cannot do better than a random guess for the secret key of Alice and Bob. There exist adversaries that can do slightly better than that, hence making the framework not completely information theoretically secure.
4) The measured secrecy rate is 3-18 Kbps. These values occur from measurements and are not compared against theoretical ones, since these are not derived for the system and adversary model considered in the iJam framework.

V. Preliminary Results

In this section we present our preliminary results [11] of a pairwise secret key agreement protocol, along with algorithms and techniques that enable this protocol to work on an actual wireless network, consisting of commodity wireless devices.

We consider n nodes, $T_1, \ldots, T_n$ and a passive adversary Eve, all connected to the same broadcast channel. We slightly differentiate from the standard pairwise setup discussed so far and we consider a simultaneous everyone-with-everyone

pairwise key generation. We hence assume that each terminal T_i is "honest but curious" toward the other terminals, i.e., T_i runs the protocol honestly but may eavesdrop on other terminals' communications. The protocol described next enables each pair of terminals $T_i - T_j$ to create a secret S_{ij}, such that any other terminal $T_{l\neq i,j}$ or Eve obtain very little information on S_{ij} .

A. Protocol description

The protocol consists of two phases. In the *initial* phase the terminals exchange traffic (packets of fixed length) to ensure that each terminal pair shares some information, and in the *privacy amplification* phase the terminals create pairwise secrets by compressing this shared information.

Each terminal T_i maintains $n-1$ queues $\mathcal{Q}_{ij}$, $j \neq i$. In the beginning, these are empty.

Initial Phase

In round $k = 1 \dots n$:

1) Terminal T_k transmits N random packets (we will call them x*-packets*).
2) Each terminal $T_{i\neq k}$ reliably broadcasts the identities of the x-packets it received.
3) Each terminal T_i adds to queue $\mathcal{Q}_{ij}$ the identities and contents of the x-packets it shares with terminal $T_{j\neq i}$.

At this point, $\mathcal{Q}_{ij}$ contains all the packets shared by terminals T_i and T_j.

Privacy Amplification Phase

For $i = 1 \dots n-1$:

1) Terminal T_i constructs M_{ij} linear combinations of the packets in the queue $\mathcal{Q}_{ij}$, for all $j > i$ (we will call them y*-packets*).
   It determines the number of y-packets M_{ij} and constructs the y-packets as described in Appendix-A.
2) Terminal T_i reliably broadcasts the coefficients it used to construct the y-packets.
3) Each terminal $T_{j>i}$ uses the broadcasted coefficients and the contents of its queue $\mathcal{Q}_{ji}$ to reconstruct the M_{ij} y-packets.

At this point, terminals T_i and $T_{j>i}$ share M_{ij} y-packets. Their secret $\mathcal{S}_{ij}$ is the concatenation of these y-packets.

We define the *theoretical network conditions* as follows:

1) When a terminal T_i transmits, a terminal T_j (or Eve) either misses the entire packet with probability δ_{ij} or receivers the entire packer correctly.
2) The erasure probability δ_{iE} of the $T_i - Eve$ channel is known, for all i.
3) The T_i – T_j channel is independent from any T_i – $T_{l\neq j}$ channel[1] and the T_i – Eve channel, for all i, j, l.

Under these theoretical network conditions, by using the result in [7] it is proved that our protocol is information theoretically secure against a passive adversary. In addition, since the most demanding operations a terminal need to perform is linear combining to create the $y-$packets, our protocol executes an algorithm that is polynomial with respect to the number of $x-$packets transmitted N and the number of terminals n.

Finally, let us denote the efficiency of our protocol as:

$$E = \frac{M_{ij}}{Nn}.$$

Lemma V.1. If the theoretical conditions hold and we assume non-colluding eavesdroppers, then there exists a sufficient large N for which our protocol achieves $E = \delta_E(1-\delta)$, for $n = 2$ terminals.

The above lemma essentially states that the efficiency we achieve for $n = 2$ reaches Maurer's upper bound (see Theorem II.2).

B. Adapting to real networks

When considering real wireless networks, we do not assume that the theoretical network conditions hold. Instead we are trying conservatively to estimate the amount of information missed by Eve, based on the amount of information missed by the honest terminals. We do the following operation in order to be able to compute M_{ij}, the number of $y-$packets that are constructed at step 1 of the privacy amplification phase:

Terminals T_i and T_j estimate that, at the end of the initial phase, from their shared x-packets, Eve misses the following number:

$$V_E = \sum_{k=1}^{n} \min\{\ V_1^k, V_2^k, \dots V_n^k\ \},$$

where:

- V_l^k is the number of new x-packets shared by terminals T_i/T_j and missed by terminal T_l during round k of the initial phase.

In short, we assume that, in each round of the initial phase, Eve missed as few (of the x-packets newly shared by T_i/T_j in this round) as any other terminal. This of course is an empirical estimation, thus we cannot guarantee its accuracy theoretically. Nevertheless, we can evaluate its performance through experimental results.

We experimentally evaluate our adapted secret key agreement protocol on a wireless indoor testbed. The testbed comprises 6 nodes (HTC Wildfire Android Smartphones) set to 802.11 ad-hoc mode and with high fixed transmission rate to 36Mbps. In order for our approach to work, the wireless network must provide a certain level of channel variability. The simplest scenario where such variability exists is when the nodes are not in direct line of sight, e.g., they are separated by office walls. This is the scenario we implement in our testbed.

We are interested in measuring the secrecy rate achieved by our protocol and also the level of *reliability* of the generate keys. In other words, we are interested in knowing how well we do with our technique (for estimating how many packets Eve missed in the end) compared to what we could do if we were assisted by an *oracle* to know exactly that information (in our setup we choose one node to be Eve and we collect the actual information loss from her). We define, thus, reliability as the ratio: ideal number of secret bits over estimated number

[1]Assuming that the channels between terminals are independent is not necessary for any of our results, but simplifies the proofs of our theoretical results. Our protocol works as long as we know the joint distribution of the erasure channels between the terminals (which we can measure in practice).

of secret bits. Ideally, we would like to achieve reliability 1. Some interesting experimental results we got are the following:

1) The minimum ideal secrecy generation rate (among all the pairs and different positions of Eve) at 15dBm (a typical transmission power for the 802.11 standard) was 35Kbps. The corresponding minimum estimated secrecy generation rate was 38 Kbps, yielding a 0.92 reliability ratio.
2) The minimum reliability observed was 0.82 for 5dBm transmission power.

VI. Discussion and Research Plan

We first investigated information theoretic bounds on achievable pairwise secrecy over broadcast channels [8], and we saw that by allowing the two honest nodes to exchange feedback over a public channel we obtain non zero secrecy capacity, even if Eve's channel is better. Next, we presented two approaches towards building practical information theoretically secure systems [15], [5].

Along to these lines, we presented preliminary results of a pairwise secret key agreement protocol and the techniques for adapting it to a real wireless network. As a first step of investigation, we aim to build protocols that leverage the secret key generation from common information between the honest nodes. We use well established techniques [7] to prove that our protocol is information theoretically secure and we aim to practical implementations that involve polynomial time operations. For adapting to real networks we propose a heuristic algorithm and we test its efficiency. Unlike to the other two state-of-the-art practical frameworks presented here, our approach, first, works on commodity network devices without demanding any software or hardware modification, second we evaluate our performance against theoretical results, aiming to optimal scheme construction, and finally, we achieve high secrecy rates (in the magnitude of Kilobits). We envision the use of our scheme in realistic scenarios where there is a continuous need for freshly generated secrets keys, and the participating nodes would prefer not to spent resources in computationally expensive operations.

They key challenge when considering real wireless networks is that the theoretical network conditions may not hold any more. We expect that investigating this problem may lead us to system development that simulate these theoretical network conditions or impose specific channel conditions known to the honest parties. Introducing artificial noise by jamming as in [5], might bring us one step closer towards this direction. Also, developing sophisticated inference techniques based on collaborative schemes (as our proposed heuristic) for estimating accurately the capabilities of Eve, is of great interest too.

The above discussion may not stay restricted to pairwise secret key generation, but it can be extended to multiterminal shared secret key generation [7]. In addition, so far we assume that all nodes are connected to the same wireless broadcast domain. Exploring multi-hop schemes and the corresponding practical implementations would be a natural future step of this work.

References

[1] C.H. Bennett, G. Brassard, C. Crépeau, and U.M. Maurer. Generalized privacy amplification. *Information Theory, IEEE Transactions on*, 41(6):1915–1923, 1995.

[2] J.L. Carter and M.N. Wegman. Universal classes of hash functions. *Journal of computer and system sciences*, 18(2):143–154, 1979.

[3] I. Csiszár and J. Korner. Broadcast channels with confidential messages. *Information Theory, IEEE Transactions on*, 24(3):339–348, 1978.

[4] W. Diffie and M. Hellman. New directions in cryptography. *Information Theory, IEEE Transactions on*, 22(6):644–654, 1976.

[5] S. Gollakota and D. Katabi. Physical layer wireless security made fast and channel independent. In *INFOCOM, 2011 Proceedings IEEE*, pages 1125–1133. IEEE, 2011.

[6] JE Hershey, AA Hassan, and R. Yarlagadda. Unconventional cryptographic keying variable management. *Communications, IEEE Transactions on*, 43(1):3–6, 1995.

[7] M. Jafari Siavoshani, C. Fragouli, S. Diggavi, U. Pulleti, and K. Argyraki. Group secret key generation over broadcast erasure channels. In *Signals, Systems and Computers (ASILOMAR), 2010 Conference Record of the Forty Fourth Asilomar Conference on*, pages 719–723. IEEE, 2010.

[8] U.M. Maurer. Secret key agreement by public discussion from common information. *Information Theory, IEEE Transactions on*, 39(3):733–742, 1993.

[9] A. Rényi. On measures of entropy and information. In *Fourth Berkeley Symposium on Mathematical Statistics and Probability*, pages 547–561, 1961.

[10] R.L. Rivest, A. Shamir, and L. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, 21(2):120–126, 1978.

[11] I. Safaka, C. Fragouli, K. Argyraki, and S. Diggavi. Exchanging pairwise secrets efficiently. In *under submission INFOCOM, 2013 Proceedings IEEE*.

[12] D. Tse and P. Viswanath. *Fundamentals of wireless communication*. Cambridge Univ Pr, 2005.

[13] R. Wilson, D. Tse, and R.A. Scholtz. Channel identification: Secret sharing using reciprocity in ultrawideband channels. *Information Forensics and Security, IEEE Transactions on*, 2(3):364–375, 2007.

[14] A.D. Wyner. The wire-tap channel. *Bell System Technical Journal*, 54(8):1355–1387, 1975.

[15] S. Xiao, W. Gong, and D. Towsley. Secure wireless communication with dynamic secrets. In *INFOCOM, 2010 Proceedings IEEE*, pages 1–9. IEEE, 2010.

[16] C. Ye, S. Mathur, A. Reznik, Y. Shah, W. Trappe, and N.B. Mandayam. Information-theoretically secret key generation for fading wireless channels. *Information Forensics and Security, IEEE Transactions on*, 5(2):240–254, 2010.

Appendix
Secret construction in Section V-A

Terminals T_i and T_j construct the following number of y-packets in the privacy amplification phase:

$$M_{ij} = \min\{\, V_E, V_1, V_2, \dots V_n \,\},$$

where:

- V_E is the expected number of x-packets that are shared by terminals T_i/T_j and missed by Eve.
- V_l is the number of x-packets shared by terminals T_i/T_j and missed by terminal T_l.

We compute V_E as $\sum_{k=1}^{n} U_{Ek}$, where $U_{Ek} = \delta_{kE} \cdot U_k$, and U_k is the number of x-packets transmitted by terminal T_k and received by both terminals T_i/T_j in round k of the initial phase. In short, we count, for each terminal and for Eve, how many of T_i/T_j's shared x-packets this terminal/Eve has missed (or is expected to have missed, in Eve's case), and we set M_{ij} to the smallest of these numbers.

Terminals T_i and T_j construct the y-packets using simple constructions as described in [7].