A Novel Cryptographic Key Exchange Scheme using Resistors

Pey-Chang Kent Lin, Alex Ivanov, Bradley Johnson, Sunil P. Khatri
Department of ECE, Texas A&M University, College Station TX 77843

***Abstract*— Recently, a secure key exchange technique was developed, in which both communicators (Alice and Bob) randomly select between two known resistors. By measuring the resulting thermal noise on a shared wire, they can each determine the resistor chosen by their counterpart, while the eavesdropper (Eve) cannot determine this. By repeating this transaction, they can create a common secure key, one bit a time. Although theoretically elegant, this approach is difficult to realize in practice. In this paper, we present a practical realization of a secure key exchange technique, intended for use over the Ethernet. Our approach is inspired by the above scheme with significant differences. In our approach, Alice and Bob utilize programmable resistors and exchange their resistance values securely. Our technique has been implemented in a hardware FPGA based platform, and was found to be able to exchange 4 secure bits per transaction over a 100ft CAT5 cable.**

I. Introduction and Previous Work

In secure communication, it is customary to for the sender to combine a pseudorandom sequence (*key*) with a message to be sent (*plaintext*) to create an encrypted message, referred to as a *ciphertext*, before transmitting to the recipient. The algorithm used to combine the key and the plaintext is referred to as the *cipher*. A common cipher structure uses a bit-wise XOR operation of the key bits and plaintext bits. The intended receiver would decrypt the plaintext by using the same key that was used during transmission.

The key is a secret known only to the communicators and the ciphertext is ideally undecipherable by unintended listeners. Security is provided since in practice, decoding the ciphertext without the key is computationally complex. As such, before secure communication can begin between two communicators (Alice and Bob), both parties must securely generate the shared secret key through their public communication channel, under the assumption that an eavesdropper (Eve) may also be monitoring the channel.

One recent method of a secure means to generate a shared key, is to use two pairs of identical resistors (at the ends of a wire that is shared by the communicating partners, Alice and Bob) which was initially introduced in [1] and further developed in [2]. Using the statistical physical properties of thermal noise, Alice and Bob create a shared secure key one bit at a time.

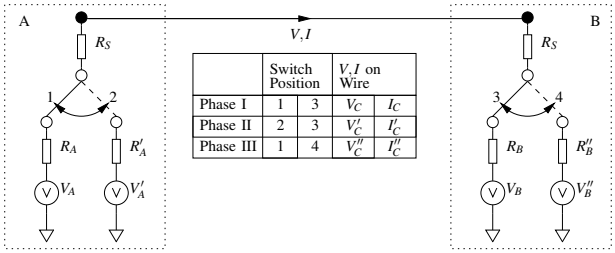
In [1], [2], the resistors on each side of the wire comprise of a small and a large resistor, R_S and R_L, respectively. Alice and Bob each randomly connect one of their resistors to their end of the information channel (a shared wire). The RMS value of the thermal noise in a resistor R [3] is given by formula 1.

$$V_{rms} = \sqrt{4kT\Delta fR} \tag{1}$$

where k is the Boltzmann constant, T is the absolute temperature, and Δf is the measurement bandwidth.

Therefore, when Alice and Bob both connect the same resistor R_L (R_S) to their end of the wire, the RMS voltage on the wire is $\sqrt{4kT\Delta f(\frac{R_L}{2})}$ ($\sqrt{4kT\Delta f(\frac{R_S}{2})}$). However, when Alice and Bob connect different resistors to their end of the wire, the RMS voltage is $\sqrt{4kT\Delta f(R_L \parallel R_S)}$, which is an intermediate value. In this second case, Eve has no way of knowing if Alice chose R_L and Bob chose R_S, or vice versa, while Alice and Bob know the resistance value chosen by their counterpart. Without loss of generalization, they determine that a '0' bit was exchanged if Alice chose R_L and Bob chose R_S and a '1' bit was exchanged for the reversed case.

In this paper, we present a practical realization of a secure classical communication scheme intended for use in Ethernet applications. Our scheme is inspired by [1] [2], but with key differences. In our scheme, Alice and Bob utilize programmable resistors and DC voltage regulators connected to their ends of the wire to exchange their resistance values. By using a shared resistor R_S known only to Alice and Bob, each communicator can measure the channel voltage to determine its counterpart's selected (secret) resistor value. On the other hand, the eavesdropper can only measure and determine the total loop resistance, but without the value of the shared resistor, is unable to determine either communicator's secret resistor. Switching to DC voltage sources instead of relying on the thermal noise of resistors allows the use of low-cost, off-the-shelf (OTS) ICs for regulation and channel measurement, enabling widespread use of our scheme in communication platforms.



	Switch Position		V,I on Wire	
Phase I	1	3	V_C	I_C
Phase II	2	3	V'_C	I'_C
Phase III	1	4	V''_C	I''_C

Fig. 1. Secure Communication Scheme using Programmable Resistors

The scheme of [1] [2] is theoretically stronger since it utilizes no shared secret to begin with, unlike ours in which a shared secret resistor R_S is employed. In practice however, [1] [2] is very hard to realize, requiring matched amplifiers, resistors, and temperatures, and long sampling times. In contrast, our scheme is realized using simple, inexpensive OTS hardware and can be easily integrated into Ethernet routers.

The key contributions of this paper are:

- We show how inexpensive, OTS programmable resistors can be used to implement a classically secure communication scheme which is secure against man-in-the-middle attacks.
- Our scheme is also easily realizable with existing technology and can be integrated into Ethernet routers.
- We experimentally validate our approach via a hardware implementation, which can exchange 4 secure bits per transaction, over 100ft CAT-5 Ethernet cable.

The remainder of this paper is organized as follows. In Section II we describe our approach for secure key exchange. Section III presents experimental results from a hardware implementation, while conclusions are drawn in Section IV.

II. Our Approach

Our secure key exchange scheme is shown in Figure 1. Alice and Bob share a secret resistor R_S and a wire (representing the communication channel). A transaction consists of three phases as shown in Figure 1.

We assume that Eve can monitor and measure the voltage and current on the shared wire, but has no information about $R_A, R'_A, R_B, R''_B, V_A, V'_A, V_B, V''_B$. Alice and Bob know their own voltages and resistor values, and can measure the voltage and current of the shared wire. The goal of the 3-phase transaction is for Alice to learn R_B and for Bob to learn R_A without Eve gleaming these values.

The secure key exchange occurs in 3 phases:

Phase I: Initially, both communicators select a random value of resistance and voltage. The selected values are shown in Figure 1. V_C and I_C are measured and recorded by Alice and Bob (and possibly Eve as well).

$$\frac{V_C - V_A}{R_S + R_A} = -\frac{V_C - V_B}{R_S + R_B} = I_C \tag{2}$$

Phase II: In this phase, Alice determines R_B. Bob holds his resistance and voltage steady (same values as in Phase I) while Alice chooses new resistance and voltage values as shown in Figure 1. Alice can now determine Bob's resistance by solving the following equations for $R_S + R_B$.

$$\frac{V'_C - V'_A}{R_S + R'_A} = -\frac{V'_C - V_B}{R_S + R_B} = I'_C \tag{3}$$

978-1-4577-1954-7/11/$26.00 ©2011 IEEE

$$\frac{V_C' - V_C}{I_C' - I_C} = R_S + R_B \tag{4}$$

Phase III: In this phase, Bob determines the value of R_A. Alice sets her resistance and voltage to the values from Phase I, while Bob changes his values as shown in Figure 1. $R_S + R_A$ can then be known to Bob.

$$\frac{V_C'' - V_A}{R_S + R_A} = -\frac{V_C'' - V_B''}{R_S + R_B''} = I_C'' \tag{5}$$

$$\frac{V_C'' - V_C}{I_C'' - I_C} = R_S + R_A \tag{6}$$

Because both communicators know R_S (a shared secret between Alice and Bob), it is straightforward to determine R_A (R_B). Thus, in a single transaction, Alice and Bob exchange each other's resistor values R_A and R_B securely. The secure key now becomes $r_a \oplus r_b$, where r_a and r_b are the binary representation of R_A and R_B respectively.

In a practical realization, a few items need to be kept in mind.

- In a real-world noisy environment, only a few bits of r_a and r_b can be reliably exchanged, and multiple transaction may be required to generate a long secure key. In our implementation, 4 bits of r_a and r_b are exchanged per transaction.
- The shared secret resistor R_S can be made variable as well, but must be identical in both communicators, and constant during any key exchange transaction.
- In practice, the values of $R_A, R_A', R_B, R_B'', V_A, V_A', V_B, V_B''$ are changed randomly in each transaction since Alice and Bob both have variable resistors and voltage sources at their side of the wire. In our implementation, the resistors are realized as digital potentiometers, and the voltage sources are realized using ADCs.

The above scheme is immune to the two forms of man-in-the-middle [4] attacks that Eve may launch.

- *Passive attacks:* Eve can measure the current and voltage on the public wire in each of the 3 phases of a transaction. This will allow her to find $(R_S + R_A)$ and $(R_S + R_B)$ but since R_S is a shared secret, Eve cannot determine R_A or R_B.
- *Active attacks:* Eve can maliciously inject current into the channel, but this action disturbs the voltage and current measured by Alice and Bob. Hence, if the injected current is sufficiently large, and occurs when the voltages and currents have settled in any phase, then Eve's presence is detected.
- If Eve's current injection is not detected (perhaps occurs during a phase transaction, or if its magnitude is smaller than the resolution of Alice's and Bob's measurement equipment) then the keys generated by Alice and Bob will be different. Now, by initiating a *challenge response*, where in Alice/Bob encrypts and sends a simple random challenge for Bob/Alice to respond. Bob/Alice then returns the encrypted answer to Alice/Bob, where a correct answer will confirm to Alice and Bob the key exchange was successful. In both scenarios above, Eve still does not learn R_A and R_B since R_S is a secret.
- Eve may interpose herself on the wire and impersonate herself as Bob (Alice) to Alice (Bob). In this case, her attack is also detected by Alice and Bob by a challenge response. Further, Eve still does not learn R_A or R_B.

III. Implementation

To execute the secure key exchange in hardware, we implemented the transaction protocol of Section II. Let us assume the communicator that initiates the key exchange transaction is Alice (A), while the communicator that responds is Bob (B). In addition to the communication channel, let us also assume that Alice and Bob also share a rendezvous channel used to signal the start of a transaction. In this scenario, the transaction protocol is as shown below.

Step 0 : Wait for the rendezvous channel to be released (or high-Z).

Step 1 : [A] Pull-down rendezvous channel to initiate transaction.

Step 2 : When rendezvous channel is pulled-down randomly select R_A, V_A, R_B, and V_B. Measure V_C and I_C.

Step 3 : [A] Randomly select R_A' and V_A' where $R_A' \neq R_A$ and $V_A' \neq V_A$. Measure V_C' and I_C'. Reset back to R_A and V_A.

Step 4 : [B] Randomly select R_B'' and V_B'' where $R_B'' \neq R_B$ and $V_B'' \neq V_B$. Measure V_C'' and I_C''.

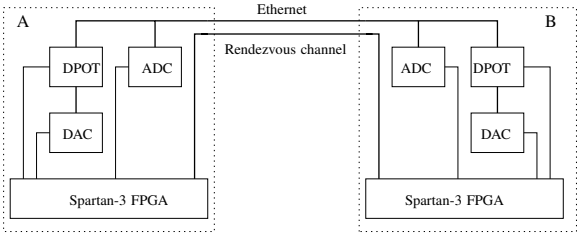


Fig. 2. Block Diagram of ICs and FPGA in the Communicator Pair

Step 5 : Calculate R_A and R_B (Eq 4, 6) and save key $r_a \oplus r_b$.

Step 6 : [A] Release rendezvous channel.

Step 7 : Conduct a challenge response to ensure secure key was correctly exchanged.

In any transaction, it is possible that $(I_C' - I_C)$ or $(I_C'' - I_C)$ or $(V_C' - V_C)$ or $(V_C'' - V_C)$ are extremely small. This would result in an extremely large error in the calculation of $(R_S + R_B)$ or $(R_S + R_A)$. To detect this situation, Alice and Bob uses a challenge response.

To demonstrate our secure communication scheme, we have implemented it using off-the-shelf components. Each communicator consists of three IC's: a 10-bit DAC, a 10-bit digital POT, and a 12-bit ADC. The DAC generates V_A and V_B with a range of 0V to 4.096V. The digital POT has a resistance range from 0Ω to 50KΩ. All IC's are driven by a Xilinx Spartan 3 FPGA board (as shown in Figure 2). Note, the digital POT implements $R + R_S$ as we can offset the digital POT input by the equivalent series resistance of R_S.

The Xilinx Spartan 3 FPGA handles component drivers, key exchange protocol, resistor calculation, and key display. Linear feedback shift registers (LFSRs) are implemented in the FPGA to drive the random selection of the V and R values for the DAC and POT inputs. LFSRs were chosen due to their desirable properties in the context of random number generation (such as ideal bit frequency and maximal-length period) and simplicity of implementation. The duration of a complete transaction is approximately 10ms.

The intention of the proposed system is to operate in existing Ethernet-based wired communication networks. We note that for 10BASE-T and 100BASE-TX networks, the GND and VDD wire pairs in the CAT-5/6 Ethernet cable are unused in non-POE (Power over Ethernet) applications. Our implementation uses the GND wire pairs for the communication and rendezvous channels.

Each communicator was implemented in hardware as described above, and the secure communication scheme has been tested in lab using a 100ft CAT-5 cable as the channel. It was experimentally observed that the network can reliably exchange 4 key bits per transaction. The limiting factors for key generation throughput are noise in the channel and power supply, as well as precision of the ICs. As such, larger key exchanges could be achieved with better noise filtering and higher quality components. However, we have shown that even in the noisy environment of Ethernet network, and using inexpensive components, the scheme can securely exchange a key between communicators.

IV. Conclusions

We present a secure exchnage to meet the increased necessity for practically realizable, yet highly secure communication. Our method uses a simple shared secret resistor along with programmable resistors and voltage regulators. It provides security agains active and passive eavesdropping. This method is a cost effective and flexible solution, which can be easily augmented into existing the Ethernet infrastructure. In particular, our hardware implementation, which consists of commercially available ICs and FPGAs, has been able to exchange 4 secure bits per transaction over a 100ft standard CAT-5 Ethernet cable.

Acknowledgement

We would like to thank the authors of [1] for their helpful discussions and feedback in the development of the programmable resistor based secure communication scheme.

References

[1] L. B. Kish, "Totally secure classical communication utilizing Johnson (-like) noise and Kirchoff's law," *Physics Letters A*, vol. 352, no. 3, pp. 178 – 182, 2006.
[2] L. B. Kish and R. Mingesz, "Totally secure classical networks with multipoint telecloning (teleportation) of classical bits through loops with Johnson-like noise," *Fluctuation and Noise Letters*, vol. 6, no. 2, pp. C9 – C21, 2006.
[3] J. A. Connelly, *Low-Noise Electronic System Design*. New York, NY, USA: John Wiley & Sons, Inc., 1st ed., 1993.
[4] L. B. Kish, "Protection against the man-in-the-middle-attack for the Kirchhoff-loop-Johnson(-like)-noise cipher and expansion by voltage-based security," *Fluctuations and Noise Letters*, vol. 6, pp. L57 – L63, Dec. 2005.