Formal Modeling and Analysis of Bluetooth 4.0 Pairing Protocol

David (Wei) Jia
Stanford University
djia@stanford.edu

Richard Hsu
Stanford University
rhsu@cs.stanford.edu

ABSTRACT

Bluetooth is a wireless technology for exchanging data over short distances and is built into many of the devices used in daily life such as smartphones and laptops. Bluetooth allows communication between paird devices. The pairing is done through Bluetooth's pairing protocol known as Secure Simple Pairing (SSP) which has been established since Bluetooth 2.1+. Bluetooth security is important because sensitive and confidential information such as phone conversations, messages, and key strokes on a keyboard are often communicated through a pairing. We employ a formal model checker to verify the security properties of the pairing protocol used in Bluetooth 4.0, the latest iteration of the protocol. Using our formal Murphi model, we demonstrate and confirm previously known attacks on Bluetooth that have not yet been formalized. We then discuss a new attack found by our model as well as its implications to Bluetooth 4.0 security. Finally, we discuss and recommende possible fixes that could be employe to avoid the attacks we have foundd. Through this formal modeling we expand on thesecurity verificatios og Bluetooth pairing protocol.

1. INTRODUCTION

Bluetooth is a wireless technology that is ubiquitous in many modern applications. Users rely on it to transfer data between smartphones, laptops, on-board car systems, and headsets. Bluetooth technology also exists in more peripheral items such as printers, mice, keyboards, and speakers. Because Bluetooth serves as an underlying systems for so many communication systems that are used on a daily basis, sentivie and confidential information can often times be passed through the supposedly secure paired channel. Thus, we analyze Bluetooth in this paper through a formal modeling and analysis in hopes of shedding light on its security.

In this paper, we focus our attentions on the Bluetooth 4.0 Protocol [**?**], which is the most recent version of the technology at the time of writing.

2. RELATED WORKS

Because of BluetoothÕs prevalence and the emergence of mobile technology, several related works have been done in an effort to analyze the security of Bluetooth. Most recently, in 2010, Phan and Mingard [3] analyzed the Bluetooth 4.0 protocol in ÒAnalyzing the Secure Simple Pairing in Bluetooth v4.0Ó which hand modeled and analyzed Bluetooth 4.0 and described three Man in the Middle (MitM) attacks for the NC and PE modes. In their 2007 paper, ÒFormal Analysis of Authentication in Bluetooth Device PairingÓ, Chang and Shmatikov [4] modeled Bluetooth 2.0 using ProVerif cryptographic protocol verifier and verified a known guessing attack and discovered a potential vulnerability which produces concurrent execution of authentication. Similarly, Haataja and Toivanen [8] discussed in ÒPractical man-in-the-middle attacks against bluetooth secure simple pairingÓ two more MitM attacks on Bluetooth 2.0 handsets and hand-free devices as well as modeled a vulnerability on the OoB mode given that the attacker has visual contact to the victim devices. Other papers are more general and informative, presenting a high-level view of the Bluetooth protocol, or suggesting potential vulnerabilities and risks associated with Bluetooth without providing formal analysis or proof.

While these papers present a good starting point for modeling Bluetooth 4.0, they are insufficient. First, many of the analysis of Bluetooth are either not formal or performed on outdated versions, such as v2.x. The most up-to-date analysis of Bluetooth 4.0 by Phan and Mingard encapsulated a formal analysis of only the NC and PE modes without formal modeling with a cryptographical tool such as Murphi. Furthermore, the paper focused on MitM attacks between only two pairing devices, and did not discuss eavesdropping, denial of service attacks, or multiple devices.

In our paper, we will perform a formal analysis and modeling (with Murphi) of Bluetooth 4.0, which has several improved and modified security features from its predecessors. We will first confirm the findings of other papers, such as that of Phan and Mingard, by modeling MitM attacks on Bluetooth in NC and PE modes using Murphi. We will then explore MitM attacks on JW and OoB modes. We will also investigate other potential attacks and vulnerabilities on the various modes of Bluetooth 4.0, such as eavesdropping, denial of service, and brute force attacks. Finally, although existing papers only discuss Bluetooth pairing between two devices (a receiver with one initiator), Bluetooth can sup-

port up to seven devices. Because of this, and for the sake of completeness, we will also model Bluetooth for multiple devices. This is important because each Bluetooth receiver should guarantee security between multiple devices that are synced to any given receiver.

3. BLUETOOTH OVERVIEW

Bluetooth is a wireless technology standard used for exchanging data over a short distance. It utilizes short-wavelength radio transmission and is implemented in devices ranging from hands-free headsets to computer peripherals such as a mouse or keyboard. These devices are fairly ubiquitous in todayÕs day and age and some examples of the data transferred between devices via Bluetooth are inputs from a wireless keyboard, contacts transferred between phones, files transferred between computers, and even health data in sensors.

The security of the Bluetooth communication is vitally important. The major part of Bluetooth that becomes vulnerable to attackers is the the pairing protocol when two or more devices want to connect with each other. The connection process creates a personal area network and requires authentication and encryption to remain secure. Since Bluetooth 2.1, Secure Simple Pairing (SSP) has been implemented and further improved in subsequent versions leading to the current discussed design version of Bluetooth 4.0.

3.1 Secure Simple Pairing Protocol

The Secure Simple Pairing contains four main association models depending on the devices involved:

1. Just Works (JW) When at least one device does not have display nor input channel. Example: Wireless headset.
2. Numeric Comparison (NC) When both devices have a display and at least one has a binary input channel for a yes or no response. Example: pairing between smartphones.
3. Passkey Entry (PE) When one device has an input channel but no display and the other has a display but no input channel. Example: Wireless keyboard.
4. Out of Band (OoB) When both devices support a common additional wireless or wired communication technology for purpose of device discovery or cryptographic channels. Example: Near Field Communication (NFC).

The process of SSP follows the same steps for all four association models with the exception of phase 2. The process is described in detail in the subsequent sections:

3.1.1 Phase 1: Public Key Exchange

In this phase, Elliptic Curve Diffie-Hellamn (ECDH) key exchange is completed by the initiating (Device A) and non-initiating device (Device B). Both devices generate a public-private key pair using ECDH. The public key of each device is sent to the other device. Note that according to the official Bluetooth Specifications, this key pair do not need to be generated at each time a pairing occurs. Each device may discard its key pair and generate a new one at any time, but this is not a requirement.

3.1.2 Phase 2: Authentication Stage 1

In this stage, each of the four associated models has a different, but relatively similar authentication process. In each of the associated models, both devices generate a random nonce, N_a and N_b, respectively for Device A and Device B. Device B, the non-initiating device then computes a commitment value $C_b = f_1(PK_b, PK_a, N_b, 0)$, where PK_a is the public key for Device A and PK_b is the public key for Device B, and f_1 is SHA256 dependent hash-function that generates a 128-bit value with the given input.

After Cb is generated, it along with Nb is sent to Device A. Device A also sends Na to Device B. Next, Device A computes the same commitment value Ca = f1(PKb, PKa, Nb, 0) and compares it with Cb. If they do not equal, then the pairing process is aborted. If they are equal, then the process proceeds, and diverges for each of the four associated models as follows until the end of Phase 2.

1. Just Works (JW) In this associated model, since one of the devices can neither display nor input values, Phase 2 ends here and moves on to Phase 3.
2. Numeric Comparison (NC) In this associated model, Phase 2 goes on to generate two verification codes to be displayed by the user. Each device x generates a six-digit verification code Vx = g(PKa, PKb, Na, Nb), and displays it to the user. The user can then determine if Va = Vb and confirm the pairing and moves on to Phase 3.
3. Passkey Entry (PE) In PE, the two devices decide upon a secret value, created by the users of the devices. If we are pairing two devices that both do not have screens, then the values are agreed upon by the user. If one device has a display, then it generates a random value, which is shown to be inputted into the input device. This shared secret is turned into an n-bit number (n depends on the length of the shared secret). Let the i-th bit for device x be denoted as rxi. Since the secret is shared, the assumption is that all rai = rbi for all i. Then for each i, a commitment value, Cxi = f1(PKb, PKa, Nxi, rxi) is generated for device x. The twenty commitment values are sent to the other device and compared. If any commitment values do not equal, then we abort the process. Otherwise, we move on to Phase 3.
4. Out of Band (OoB) In OoB, after the commitment stage described completes, a shared secret comparison is done similar to the PE mode, except the external communication is complete through an external channel such as NFC.

3.1.3 Phase 3: Authentication Stage 2

Phase 3 confirms that both devices have successfully completed the exchange during pairing. Device A generates exchange codes Ea = f3(DHKey, Na, Nb, rb, IOcapA, A, B) and Device B generates Eb = f3(DHKey, Nb, Na, ra, IOcapB, B, A), where DHKey is the Diffie-Hellman key created in Phase 1.. Both are sent to the other device. Each device then computes the exchange code for the other device and verifies that it is the same.

3.1.4 Phase 4: Link Key Calculation

A Link Key (LK) is created by both devices by LK = f2(DHkey, Nmaster, Nslave, "btlk", BD_ADDRmaster, BD_ADDRslave). Since both devices have been authenticated at this stage, the LK generated by both devices should be equal.

3.1.5 Phase 5: LMP Authentication and Encryption

Phase 5 is the actual communication between the paired devices. The Link Manager Protocol (the controller that handles authentication and encryption) utilizes the link key developed in Phase 4 to perform authentication and encryption during the communication between the paired devices. For example if a device wants to verify a paired device it can perform a challenge-response by sending a nonce to the claimant device in which it generates a value and sends back to the verifier who can then verify the identity of the claimant device.

4. MURPHI MODEL CHECKER

5. MODELING SSP

5.1 Design Choices

5.2 Modeling Protocol Entities

5.2.1 Initiators

5.2.2 Responder

5.3 Modeling Protocol Transitions

5.4 Modeling An Adversary

The adversarial model

5.4.1 Adversarial Actions

5.5 Security Properties and Invariants

5.5.1 Authentication

Bluetooth 4.0 provides authentication by providing commitment codes, verification codes to users, and finally an exchange code (all three done in phases 2 and 3 or the authentication steps in SSP). These steps are used to authenticate the two devices being paired. Finally during the actual communication between the devices the link key that was generated by the SSP process for paired devices is used to create a challenge-response authentication between the two devices where a verifier sends a random nonce to a claimant and the claimant must then respond back with a value that is calculated using the nonce, link key, and its own Bluetooth address.

5.5.2 Confidentiality

Confidentiality in Bluetooth 4.0 is provided through the encryption of the communication channel once the link key is established. This link key is refreshed periodically to ensure the confidentiality of communication. However, confidentiality is only upheld once the link key is established, all other communication before this is in the clear. As a consequence any eavesdropper can access the information being passed between the pairing devices; however, the information being passed is usually used for verification processes and therefore an adversary should not be able to gain any information of the secret keys or generated values.

5.5.3 Intention Preservation

5.6 Starting State of Protocol

6. ANALYSIS, RESULTS, AND DISCUSSION

7. CONCLUSIONS

8. REFERENCES

[1] Bluetooth, SIG. Bluetooth core specification v4.0.30. `https://www.bluetooth.org/Technical/Specifications/adopted.htm`, June 2010.