

# Usability Testing for Secure Device Pairing in Home Networks

Received: date / Accepted: date

**Abstract** Setting up a secure communication channel between wireless devices is a crucial part of secure communications. Usually, it is done by non-expert users without knowledge or interest in security aspects. In the home, users may range from children to the elderly, with varied skills and capabilities. Recently, several methods have been proposed for easy yet secure creation of a security association. However, if user is able to commit errors while accepting the association, the security may break down. With careful design of user interactions involved, it is possible to bring the error rate down significantly. In this paper, we present results of a set of usability tests performed on device pairing between a PC and mobile phone assisted by home users.

**Keywords** Usability · user-centered design · secure device association · security · home network

## 1 Introduction

Secure device pairing means setting up a security association with relevant keys, identifiers and cryptographic algorithms for subsequent secure communications between two devices. Without any previous association between the devices, creating the association in secure, easy and intuitive fashion presents many challenges.

Today, most users have wireless devices such as mobile phones used on daily basis. The security of these systems is often invisible and uninteresting to the user, who should not have to deal with complicated setup methods

---

Jukka Valkonen  
Helsinki University of Technology  
P.O. Box 5400, 02015 TKK, Finland  
E-mail: Jukka.Valkonen@tkk.fi

Aleksi Toivonen  
Helsinki University of Technology  
E-mail: Aleksi.Toivonen@tkk.fi

Kristiina Karvonen  
Helsinki University of Technology  
E-mail: Kristiina.Karvonen@tkk.fi

or to be able to make errors that would break the system security.

Users are often considered the weakest link in computer security [22]. So, usability of device pairing with end-users is essential in ensuring the overall security. This is the case especially when the paired devices form a part of a home network, in which case the users are likely not to have technological expertise [7] and can be of quite varied cognitive ability ranging from small children to the elderly. Typically, a home network consists of multiple PCs, mobile phones and a broadband network connection.

The rest of the paper is organized as follows: In Sections 2 and 3 we present the technical background and the relevant prior work in usability. We then describe our study in Section 4 and analyze the results in Section 5. The paper ends with conclusions.

## 2 Background

### 2.1 Secure Pairing with Short Strings

As a part of the security association, the devices need to create a shared secret. For this purpose, the devices run a key establishment procedure, traditionally Diffie-Hellman key exchange [6], to create the shared secret. If this initial setup is created in an unauthenticated wireless environment, it is easy for an adversary to make the two legitimate devices to believe that they are communicating with each other, while in reality, they are communicating only with the adversary reading, modifying and relaying the messages. Such an attack is known as a Man-in-the-Middle (MitM) attack.

To prevent such attacks, the key exchange needs to be authenticated. In an environment without centralized key management or public key infrastructure, the authentication is left to the user. Recently, various different methods involving user interaction to authenticate the key establishment have been proposed. Most methods fall into the following two categories:

- Methods based on a shared secret passkey (PK): both devices are provided a secret, one time passkey. This passkey is used to authenticate the key exchange [1, 8]. For a successful attack, the attacker must know or guess the used passkey.
- Methods based on numeric comparison (NC): each device computes a verification string from negotiated material. The strings are then compared [16, 15, 19, 3]. An attack is prevented by the properties of computing the verification string: in case the devices are associated with an adversary, the verification strings are most likely to be different.

Several industrial specifications have recently adopted PK and NC based methods as possible authentication methods. These specifications include Bluetooth Simple Pairing [2], Wi-Fi Protected Setup [20] and Wireless USB [21].

Even though security is crucial in wireless networks, the same methods can be useful for wired networks, in particular, if security is not explicit, e.g. with no direct, visible cable between the devices. An example of such network is HomePlugAV [10] standard, which uses powerlines as the communication medium.

## 2.2 Attack Scenarios and Fatal Errors

The Diffie-Hellman key exchange protocol is resistant to passive attacks. Thus in order for an adversary to succeed, it needs to be active during the protocol run. An active attacker is able, for example, to force the devices to share different keys or to trick one device into a less secure method. The latter attack is called “bidding down” attack. For example, Bluetooth Simple Pairing [2] offers multiple methods, including authentication using NC and the unauthenticated “Just Works”-model. The method is negotiated based on the input-output capabilities of the devices.

A bidding down attack in Bluetooth Simple Pairing is depicted in Figure 1. When Device 1 initiates communication, it sends its display capabilities to Device 2. This message is captured by the MitM and altered so that the capabilities are downgraded for example from “display” to “no display”. As Device 2 responds to the message by its display capabilities, MitM again captures and alters the capabilities to match what Device 1 sent. Now, Device 1 starts to run NC method while Device 2 runs the weaker “Just Works”. As a result, only one device displays a string to be compared while the other device just asks for permission to accept the connection. A user educated only to detect mismatch in checksums might get confused and accept the connection.

An error made by the user is said to be fatal if it results in violation of security goal [18]. In NC protocols, if a user gives false acceptance for a verification string, the device ends up accepting a key with the adversary. In the bidding down attack, a potential fatal error is

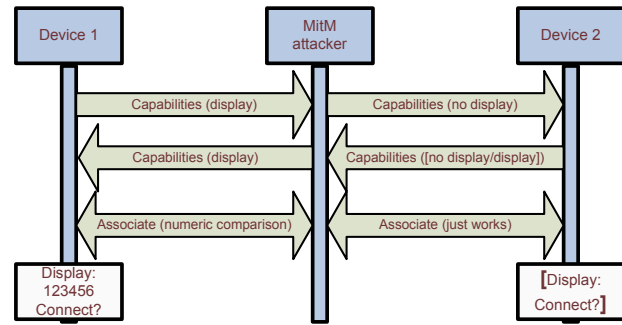


Fig. 1 “Bidding down” -attack [17]

that the user accepts the string on the displaying device without actually comparing anything.

MitM or bidding down attacks are not visible to the user in PK-based protocols. Instead, the fatal user errors are caused by the user selecting weak passkeys such as “0000” or reusing the same passkey multiple times. They can be eliminated by letting the devices with displays generate the one-time passkeys.

## 2.3 Test Framework

As a test platform, we used a framework designed for comparative usability testing of distributed applications [12]. The basic idea of the framework is to provide an easy way of testing different interaction models and graphical user interfaces (GUIs) of pairing methods. It provides the necessary tools to which the tested GUIs can easily be plugged in. The framework provides a Bluetooth channel between the devices, automated test sessions in random order, logging of all user actions and easy simulation of error conditions. In addition, the framework is designed to easily integrate mockups of GUIs into the framework.

The basic framework is implemented between two devices, called master device and remote device, communicating through a control channel. The master device is responsible for orchestrating the test session by determining the order in which the tests are performed and by issuing commands to the remote device to show the corresponding GUI. The order can be either predetermined or randomized. In addition, the master device takes care of event logging including timings and user actions.

## 3 Related Work

Work on usability testing of pairing protocols is only starting and few reports exist. The most notable examples are [14] and [18]. The latter serves as a starting point for this work.

The security and usability of setup in the Bluetooth SIG’s Simple Pairing specification [2] and the Wi-Fi Alliance’s Protected Setup specification [20] are evaluated

in [14]. The usability enhancements are based on careful analysis of these specifications. The authors suggest enhancing the overall usability in these procedures by introducing a common, consistent user interaction flow in all types of UIs involved and making the user experience as similar and simple as possible for all the devices paired. This good suggestion needs to be addressed with care. Consistency in the interaction logic is indeed a desirable quality that can aid the memorability and learnability of the procedure. However, in the UI level, the design should make best use of the properties and characteristics of each device [5]. For example, if one of the devices to be paired has a big screen and the other a small one, the ways to present the information on these two different size screens should make best use of the space available.

Further, context of usage when launching the pairing procedure may vary and have an effect. This should be reflected in the UI design. For example, a mobile phone can give a lot of information about the context, which can affect how the procedure will take place, depending on whether the pairing can be considered a bit safer (home) or not (public place). A PC screen can have a lot of room for visualizing the procedure and providing help to get through the procedure. Work in combining all these issues and addressing them is still scarce.

In [18] the usability analysis is based on actual implementations of the security procedures and their reflective GUIs, tested with real users. The pairing was done between either two PDAs or two mobile phones using several methods. The work at hand takes a further step in analyzing the usability of the pairing, now between a laptop and a mobile phone, again with real implementations tested with real users. Even if a new device - the laptop - was introduced to the test setting, the GUIs were changed as little as possible in order to allow for comparison between the two test settings. However, the context of usage remains still a lab setting only and the work needs to be extended to include more real environments. Further, this test design decision was a compromise since it meant violating the principle mentioned above about taking maximum advantage of the space and capabilities of the device used. However, the possibility to compare the results between the two tests was chosen as a priority for this test. It would be interesting and important to change the GUI on the laptop to a richer one, run the test anew, and then again compare the results.

In [18] it is argued that the familiarity with PIN usage may both advance and hinder the overall usability of handling the number sequences on mobile phones especially. The terminology used [4,11] is indeed one of the key issues in how these resemblances of former experiences are ignited: in [18] the number sequences were referred to as PINs, which caused some understandable confusion in the test users who would expect these number sequences to be more like the memorable PINs they are used to when authenticating themselves to their mo-

bile phones or at the ATM. This means care should be taken by not activating unfitting mental models in the users' minds by careless usage of UI metaphors and terminology. Because of this, we cleared all reference to PINs from the test setting, since it had clearly been confusing to the users in the [18] test. Instead, we referred to the numbers as "number sequences" to avoid the confusion.

The final major difference compared to [18] is that we included a bidding down attack described in Section 2.2 in the study.

## 4 The Study

### 4.1 Participant Profile

We recruited 38 participants (13 female, 25 male) for the tests via university news groups, mailing lists, and flyers. Most of the participants were technology students or researchers. The tests were conducted at two different locations at the premises of the university. The participants represented many nationalities, with a clear majority being Finnish citizens. The age span was 18 to 40+ years. The participants had average computer usage history of 15 years and the average computer usage per day was 8 hours. All participants in this study owned a mobile device, such as a mobile phone or a PDA.

### 4.2 Test Setting

We made a further change in chunking the 6-number sequence into chunks of three number each. This was done because in [18] it was found that a 6-number sequence was experienced as difficult and elaborate by the users, and can be identified as a clear usability problem. The chunking was expected to make this task easier and more pleasant. So, our hypothesis for the results comparison between these two studies is that even if some of the results gained in [18] are likely to be valid for the new test setting also, some of the results will probably not be valid since the test settings differ in the aforementioned points. Further, any similarities in the user feedback between the two test settings could be very valuable, since they might be interpreted as suggesting similarity in design guidelines across different types of devices, thus corroborating to the claims of [14] on the need for overall consistency of the procedure.

We selected two PK-based and three NC-based authentication methods to be tested. In the PK-based methods, named as Type, one device displayed a 6-digit string, which was then entered into the other device by the user. As we were interested in to find out which devices the users prefer to type number sequences, we ran two variants of the protocol. In PK1, the displaying device was the phone and in PK2 it was the PC. In the first NC-based method, NC1, named as Compare, both devices

**Table 1** The pairing protocols and device settings tested. PK refers to Passkeys, NC refers to Numeric Comparison.

Protocol	Method	PC	Phone	String Length
PK1	Type	Keypad	Display	6
PK2	Type	Display	Keypad	6
NC1	Compare	Display + Button	Display + Button	4
NC2	Type-and-Confirm	Keypad + Signal	Display + Button	4
NC3	Type-and-Confirm	Display + Button	Keypad + Signal	4

displayed a 4-digit string and asked for the user to compare and acknowledge or reject the values. In NC2 and NC3, one device displayed the 4-digit string and asked the user to type in the string to the other device, which then performed the comparison of the digits. The device asked the user to acknowledge or reject the value on the displaying device. The method was named Type-and-Confirm. In NC2, the displaying device was the phone and in NC3 it was the PC. See Table 1 for summary of the tested protocols and string lengths.

To achieve about the same authentication strength, the numeric string used in the PK-based protocol of Wi-Fi Protected Set Up must be twice as long as the string used for NC used in Bluetooth Simple Pairing [17]. But the required security levels vary. Bluetooth mandates the use of 6 digits while 8 digits is the recommended length in the Wi-Fi Standard, while 4 digits is also allowed. To reflect the difference between the two methods we used 4 digits for numeric comparison and 6 digits for passkeys.

Since the traditional usability measures include dealing with errors [22], we were also interested on user actions in error conditions. This is why we simulated such conditions in NC methods, where MitM and bidding down attacks can be visible for the user. The simulation was done by providing different verification strings for the devices to be compared. The users had either to compare these different values, or the device where the number sequence was typed by the user signaled an error, regardless of whether the user enters the values right or not. If the user signals false acceptance for either or both of the devices, the device accepts the connection with the adversary.

In addition, we were interested in observing the user behavior in a situation, where the devices do something unexpected, like the bidding down attack described in Section 2.2.

We used IBM Thinkpad T60 as the PC and Nokia E60 running Symbian S60 3rd edition as the phone. As a test platform, we used a framework described in [12].

### 4.3 Test Design and Procedure

**Introduction of the tests to users:** According to [13], if test users know that they are testing something security-related, their behavior may change - they care more for the security than they would normally do. This is why mentioning security was avoided in the tests.

**Choice of devices:** In the test, one user was pairing two devices, a mobile phone and a laptop, since the same user controlling both devices is the most probable real-life scenario. Only these two devices were shown to the user, although two more auxiliary devices were involved.

**Test procedure:** Users were first given a brief introduction to the study and asked to fill out the same background questionnaire as in [18]. Next, users were introduced to the basic operations needed during the test: how to move the cursor, erasing a character, etc. There were altogether nine test tasks: one non-attack scenario for each of the five methods, an attack scenario for each of the NC-based methods and a last bidding down attack test. The tests were presented to the user sequentially in random order, except for the last test, which was always the bidding down attack test. A context script of type: “You have downloaded some good songs to your laptop and want to listen to them on your way to work.” or “Your friend comes over, and you want to move photos you’ve taken with your mobile phone to his laptop” reflecting real life situations was added between each test task. This was done in order to give the users a better grasp of the effects and goals of the pairing.

To end with, the users filled out the same post-test questionnaire as in [18]. The post-test questionnaire included screenshots of each tested method for easy reference. Users were asked to associate given adjectives (e.g., “easiest”, “professional” etc.) with the methods, and to choose the method they would like to have in their own device. They were also asked what they found difficult during the test.

The tests were run in a lab-type environment: a private room with no disturbances. Tests were run individually by one test moderator. The testing time was 15-20 minutes per user, with at least 5 minutes free discussion at the end. The tests were conducted in Finnish or English. Finnish was used with native Finnish speakers and English for all others.

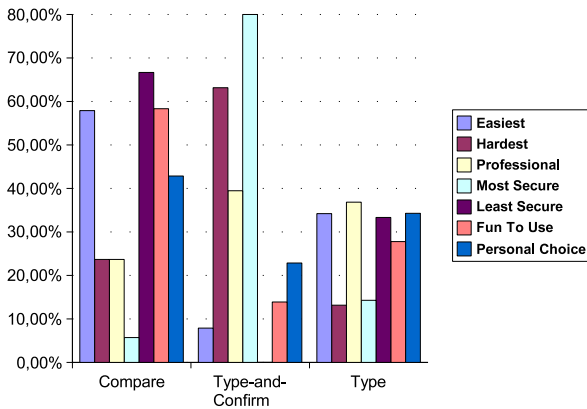
## 5 Analysis of the Results

The data collected by the framework is summarized in Table 2. Participants were asked to associate given adjectives with the methods, see Figure 2. The graph shows the percentage of the participants who associated certain adjective with a certain method variant.

The following observations can be made:

**Table 2** Summary of the tests. “Fatal errors” represents errors resulting in violation of security goal. “Total errors” represents the total number of user errors (fatal + non-fatal).

Method	Method	Average Completion Time (sec.)		Fatal Errors	Total Errors
		Match	No Match		
Type	PK1	15.4	N/A	N/A	2
	PK2	18	N/A	N/A	2
Compare	NC1	13.7	13.8	2	2
Type-and-Confirm	NC2	19	24	2	2
	NC3	21.5	24.1	2	2
Bidding Down		N/A	39.2	29	29



**Fig. 2** Summary of the participant opinions

- *Compare* was perceived as easy and fun to use, but also insecure. It was the preferred choice for a pairing method users would like to have on their own devices.
- *Type-and-Confirm* was perceived hard and by far the most secure. No-one considered it the least secure.
- *Type-and-Confirm* and *Type* were both considered professional, *Compare* less so.
- *Compare* and *Type-and-Confirm* had low fatal error rates, *Type* had no fatal errors.
- *Compare* had the lowest completion time.
- The *Bidding Down Attack* test was found to be very confusing; 72.5% of the users failed to reject the connection in both devices thus allowing at least one of the devices to connect with the MitM.
- *The effect of chunking the number sequence* This was found confusing by 8 users, probably because it was not possible to type a 'space' into the phone.

Users were also asked in which device, the phone or the laptop, they found it *easier to type numbers*. All in all, 28 users preferred the phone and 7 users the computer as input devices in this study. Moreover, 6 preferred the phone because the computer (a laptop) had no numeric keypad, while 5 thought it was easiest to type to whatever you had at hand first (the phone in this study) and 3 users had no opinion. It is of notable importance to know that some users reported that the unfamiliar keypad of the laptop was a big reason for them feeling

uncomfortable about the test setting, and claimed that they might have changed their preferences if they had been using their own laptops.

## 6 Conclusions

When compared with the results of [18], we can see that the *Compare* method was again considered as the most popular way to do the pairing. This was also the case in the first round of usability studies on PDAs reported in [18], whereas on the second round of usability studies with mobile phones, the *Type* was considered as most desirable. This change in user preferences is clearly due to the length of the number sequence used: when only 4 numbers were used for comparison, this method was perceived as easy and fun to use. With 6 number sequence, the users found it too difficult and would prefer some different method. The chunking of the number sequence was not enough to raise the preference of the *Type* method, which was a bit disappointing outcome.

The reason why *Type* was experienced as most secure may be habitual: firstly, it is not very common to compare things across the screens of a mobile phone and a laptop, whereas typing something on mobile phone is very common - and familiarity is one known ingredient in enhancing perceived usability [9].

Secondly, again here, as in earlier tests dealing with security and usability, the experienced difficulty of the user interaction gets associated with feeling secure. Users have learned to connect “hard-to-use” with “secure”, and to some extent also with “professional”. This is in line with the findings that users tend to consider security as hard-to-understand and difficult (e.g. [22]). If something is easy, it cannot be secure. This is probably why *Compare* method that was experienced easy, was not found to be the most secure one, whereas *Type* that was experienced difficult, was also rated as most secure.

### 6.1 Future Work

When analyzing the results, it is important to keep in mind that the conclusions drawn from the data collected can be considered only as indicative of the whole user base due to the test user profile focusing on persons technically above average, and the relatively small number of

participants (38). We are planning to add some users of less technical background and from different age groups in order to validate the results of the tests.

**Extending the user base** We have also run a pilot test with one family with children in order to gather data about more real-life use situation, as well to test the understandability of the procedure among all family members. The test persons were: father (41 yrs), daughter (12 yrs) and son (9 yrs). This test setting proved quite fruitful: both children were able to get through the tests without failure, whereas the father, conducting the pairing surrounded by his children, noising around, got distracted and made errors. The “bidding down” attack scenario was beautifully mastered by the kids, the daughter finding “the trick”, as she named it, quite amusing and something children are used to on daily basis - they joke around and trying to trick others is quite common and acceptable among them. The father, however, got quite confused and frustrated with this scenario, as he expected it to be more logical like the other test tasks.

As our GUI is currently based on traditional user interaction methods using numbers and text, we are able to include only those able to read. However, we are planning to introduce other types of user interaction methods, such as graphical elements, blinking lights or equivalent, which will not only allow everyone to participate, but may also provide for a significant change - and break - in the traditional association made between “hard-to-use” and “secure”, since they present completely new types of interacting with security that may in fact be even considered as fun by their users.

**Acknowledgements** We would like to thank all the users participating in the user studies. We are grateful to Kaisa Nyberg and N. Asokan for their feedback on our work. We would also like to thank Ersin Uzun for adjusting the usability test framework to enable testing on laptop also and helping to set up the tests. Finally, we would like to thank the anonymous reviewer for the comments.

## References

1. S. M. Bellare and M. Merritt. Encrypted Key Exchange: Password-Based Protocols Secure Against Dictionary Attacks. In *1992 IEEE Computer Society Symposium*, pages 72–84, 1992.
2. Bluetooth SIG. Simple Pairing Whitepaper. Technical report, Bluetooth SIG, 2006. [http://www.bluetooth.com/Bluetooth/Apply/Technology/Research/Simple\\_Pairing.htm](http://www.bluetooth.com/Bluetooth/Apply/Technology/Research/Simple_Pairing.htm).
3. M. Čagalj, S. Čapkun, and J.-P. Hubaux. Key Agreement in Peer-to-Peer Wireless Networks. *Proceedings of the IEEE (Special Issue on Security and Cryptography)*, 92(2):467–478, February 2006.
4. J. P. Chin, V. A. Diehl, and K. L. Norman. Development of an instrument measuring user satisfaction of the human-computer interface. In *CHI '88: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 213–218, New York, NY, USA, 1988. ACM Press.
5. B. de Groot. Leveraging the context of use in designing networked services. *interactions*, 13(4):45–48, 2006.
6. W. Diffie and M. E. Hellman. New Directions In Cryptography. *IEEE Transactions on Information Theory*, IT-22:644–654, 1976.
7. P. Dourish, E. Grinter, J. D. de la Flor, and M. Joseph. Security in the wild: user strategies for managing security as an everyday, practical problem. *Personal Ubiquitous Comput.*, 8(6):391–401, 2004.
8. C. Gehrmann, C. J. Mitchell, and K. Nyberg. Manual Authentication for Wireless Devices. *RSA Cryptobites*, 7(1), 2004.
9. C. A. D. Gough, R. Green, and M. Billingham. Accounting for user familiarity in user interfaces. In *CHINZ '06: Proceedings of the 6th ACM SIGCHI New Zealand chapter's international conference on Computer-human interaction*, pages 137–138, New York, NY, USA, 2006. ACM Press.
10. HomePlug AV whitepaper. HomePlug Powerline Alliance. <http://www.homeplug.org/>, 2005.
11. T. Jokela, J. Koivumaa, J. Pirkola, P. Salminen, and N. Kantola. Methods for quantitative usability requirements: a case study on the development of the user interface of a mobile phone. *Personal Ubiquitous Comput.*, 10(6):345–355, 2006.
12. K. Kostainen, E. Uzun, N. Asokan, and P. Ginzboorg. Framework for comparative usability testing of distributed applications. Technical Report NRC-TR-2007-005, Nokia Research Center, 2007. <http://research.nokia.com/tr/NRC-TR-2007-005.pdf>.
13. C. Kuo, A. Perrig, and J. Walker. Designing an evaluation method for security user interfaces: lessons from studying secure wireless network configuration. *interactions*, 13(3):28–31, 2006.
14. C. Kuo, J. Walker, and A. Perrig. Low-cost manufacturing, usability, and security: An analysis of bluetooth simple pairing and wi-fi protected setup. In *Usable Security (USEC)*, February 2007.
15. S. Laur and K. Nyberg. Efficient mutual data authentication using manually authenticated strings. In *The 5th International Conference on Cryptology and Network Security, CANS 2006*, volume 4301 of *Lecture Notes in Computer Science*, pages 90–107. Springer, 2006. A shortened version of ePrint Report, <http://eprint.iacr.org/2005/424>.
16. S. Pasini and S. Vaudenay. SAS-Based Authenticated Key Agreement. In *PKC 2006*, 2006.
17. J. Suomalainen, J. Valkonen, and N. Asokan. Security Associations in Personal Networks: A Comparative Analysis. In *Security and Privacy in Ad-hoc and Sensor Networks 4th European Workshop, ESAS 2007, Cambridge, UK, July 2-3, 2007*, number 4572 in *Lecture Notes in Computer Science*, pages 43–57, 2007.
18. E. Uzun, K. Karvonen, and N. Asokan. Usability analysis of secure pairing methods. In *Usable Security (USEC)*, February 2007.
19. S. Vaudenay. Secure Communications over insecure Channels Based on Short Authenticated Strings. In *Crypto 2005*, 2005.
20. Wi-Fi Alliance. Wi-Fi Protected Setup Specification. Wi-Fi Alliance Document, January 2007.
21. Association Models Supplement to the Certified Wireless Universal Serial Bus Specification, 2006. [http://www.usb.org/developers/wusb/wusb\\_2006\\_0302.zip](http://www.usb.org/developers/wusb/wusb_2006_0302.zip).
22. K.-P. Yee. Aligning usability and security. *Security & Privacy Magazine, IEEE*, 2:48–55, 2004.