C.B. Jones and J.L. Lloyd (Eds.): Festschrift Randell, LNCS 6875, pp. 344–357, 2011. | © Springer-Verlag Berlin Heidelberg 2011 | Republished with kind permission of Springer Science+Business Media.

Making experiments dependable |

Roy Maxion*

Abstract. In computer science and computer security we often do experiments to establish or compare the performance of one approach vs. another to some problem, such as intrusion detection or biometric authentication. An experiment is a test or an assay for determining the characteristics of the item under study, and hence experimentation involves measurements.

Measurements are susceptible to various kinds of error, any one of which could make an experimental outcome invalid and untrustworthy or undependable. This paper focuses on one kind of methodological error—confounding—that can render experimental outcomes inconclusive, but often without the investigator knowing it. Hence, valuable time and other resources can be expended for naught. We show examples from the domain of keystroke biometrics, explaining several different examples of methodological error, their consequences, and how to avoid them.

1. Science and experimentation

You wouldn't be surprised if, in a chemistry experiment, you were told that using dirty test tubes and beakers (perhaps contaminated with chemicals from a past procedure) could ruin your experiment, making your results invalid and untrustworthy. While we don't use test tubes in cyber security, the same admonition applies: keep your experiments clean, or the contamination will render them useless.

Keeping your glassware clean is part of the chem-lab methodology that helps make experimental measurements dependable, which is to say that the measurements have minimal error and no confounding variables. In cyber security we also need measurements that are dependable and error-free; undependable measurements make for undependable values and analyses, and for invalid conclusions. A rigorous experimental methodology will help ensure that measurements are valid, leading to outcomes in which we can have confidence.

A particularly insidious form of error is the confound—when the value of one variable or experimental phenomenon is confounded or influenced by the value of another. An example, as above, would be measuring the pH of a liquid placed in contaminated glassware where the influence of the contaminant on pH varied with the temperature of the liquid being measured. This is a confound, and to make things worse, the experimenter would likely be unaware of its presence or influence. The resulting pH values might be attributed to the liquid, to the temperature, or to the contaminant; they cannot be distinguished (without further experimentation). Similar measurement error can creep into cyber security experiments, making their measures similarly invalid.

This article describes some of the issues to be considered, and the rationales for decisions that need to be made, to ensure that an experiment is valid—that is, that outcomes can be attributed to only one cause (no alternative explanations for causal relations), and that experimental results will generalize beyond the experimental setting.

In the sections to follow, we first consider the hallmarks of a good experiment: repeatability, reproducibility and validity. Then we focus on what is arguably the most important of these—validity. We examine a range of threats to validity, using an experiment in

* The author is grateful for support under National Science Foundation grant number CNS-0716677. Any opinions, findings, conclusions or recommendations expressed in this material are those of the author, and do not necessarily reflect the views of the National Science Foundation.

keystroke biometrics to provide examples. The experiment is laid out first, and is then critiqued; remedies for the violations are suggested. We close by suggesting simple ways to avoid the kinds of problems described here.

2. Hallmarks of a good experiment

There are clear differences between experiments that are well-designed and those that are not. While there may be many details that are different between the two, the main ones usually reduce to issues of repeatability (sometimes called reliability), reproducibility and validity. While our main focus here will be on validity, we will first look briefly at what each of the other terms means, just to put them all in context.

Repeatability refers to the variation in repeated measurements taken by a single person or instrument on the same item and under the same conditions; we seek high agreement, or consistency, from one measured instance to another [9]. That is, the experiment can be repeated in its entirety, and the results will be the same every time, within measurement error. For example, if you measure the length of a piece of string with a tape measure, you should get about the same result every time. If an experiment is not repeatable, even by the same person using the same measuring apparatus, then there is a risk that the measurement is wrong, and hence the outcome of the experiment may be wrong, too; but no one will realize it, and so erroneous values will be reported (and assumed to be correct by readers).

Reproducibility relates to the agreement of experimental results with independent researchers using similar but physically different test apparatus, and different laboratory locations, but trying to achieve the same outcome as was reported in a source article [9]. Measurements should yield the same results each time they are taken, irrespective of who does the measuring. Using the length-of-string example, if other people can measure that same piece of string in another setting using a similar measuring device, they should get about the same result as the first group did. If they don't, then the procedure is not reproducible; it can't be replicated. Reproduction (sometimes called replication) allows an assessment of the control on the operating conditions of the measurement procedure, i.e., the ability to reset the conditions to some desired state. Ultimately, replication reflects how well the procedure was operationalized.

FIGURE 1. Hallmarks of a good experiment.

Note that reproducibility doesn't mean hitting `return` and analyzing the same data set again with the same algorithm. It means conducting the entire experiment again, data collection and all. If an experiment is not reproducible, then it cannot be replicated by others in a reliable way. This means that no one will be able to verify that the experiment was done correctly in the first place, hence placing an air of untrustworthiness on the original results. Reproducibility hinges on operational definitions for the measures and procedures employed in the course of the experiment. An operational definition defines a variable or a concept in terms of the procedures or operations used to measure it. An operational definition is like a recipe or set of detailed instructions for describing or measuring something.

Validity relates to the logical well-groundedness of how the experiment is conducted, as well as the extent to which the results will generalize to circumstances beyond those in the laboratory. The next section expands on the concept of validity.

3. Validity

What does the term *valid* mean? Drawing from a standard dictionary, when some thing or some argument or some process is *valid,* it is well-grounded or justifiable; it is logically correct; it is sound and flawlessly reasoned, supported by an objective truth.

To conduct an experiment that was anything other than valid, in the above sense, would be foolish, and yet we see such experiments all the time in the literature. Sometimes we can see the flaws (which some would call *threats to validity*) directly in the experiment, and sometimes we can't tell, because authors do not report the details of how their experiments were conducted. Generally speaking, there are two kinds of validity—internal and external. Conceptually, these are pretty simple.

Internal validity. In most experiments we are trying to find out if A has a given effect on B, or if A causes B. To claim that A indeed causes B, the experiment must not offer any alternative causes nor alternative explanations for the outcome; if this is case, then the experiment is internally valid [8]. An alternative explanation for an experimental outcome can be due, for example, to confounded variables that have not been controlled.

For example, suppose we want to understand the cause of errors in programming. We recruit students in university programming classes (one class uses C, and the other uses Java). We ask all the students to write a program that calculates rocket trajectories. The results indicate that C programmers make more programming errors, and so we conclude that the C programming language is a factor in software errors. Drawing such a conclusion would be questionable, because there are other factors that could explain the results just as well. Suppose, for example, that the Java students were more advanced (juniors, not sophomores) than the C students. The outcome of the experiment could be due to the experience level of the students, just as much as it could be due to the language. Since we can't distinguish distinctly between experience level and language, we say that the experiment confounds two factors—language and experience—and is therefore not valid. Note that it can sometimes be quite difficult to ensure internal validity. Even if all the students are at the same experience level, if they self-selected Java vs C it would still allow for a confound in that a certain kind of student might be predisposed to select Java, and a different kind of student might be predisposed to select C. The two different kinds of students might be differentially good at one language or the other. The remedy for such an occurrence would be to assign the language-student pairs randomly.

External validity. In most experiments we hope that the findings will apply to all users, or all software, or all applications. We want the experimental findings to generalize from a laboratory or experimental setting to a much broader setting. To the extent that a study's findings generalize to a broader population (usually taken to be "the real world"), the experiment is externally valid [8]. If the findings are limited to the conditions surrounding the study (and not to broader settings), then the experiment lacks external validity. Another way to think about this is that external validity is the extent to which a causal relationship holds when there are variations in participants, settings and other variables that are different from the narrow ranges employed in the laboratory.

Referring back to our earlier example, suppose we were to claim that the experiment's outcome (that the C language promotes errors) generalizes to a set of programmers outside the experimental environment—say, in industry. The generalization might not hold, perhaps because the kind of problem presented to the student groups was not representative of the kinds of problems typically encountered in industry. This is an example of an experiment not generalizing beyond its experimental conditions to a set of conditions more general; it's not externally valid.

Trade-off between internal and external validity. It should be noted that not all experiments can be valid both internally and externally at the same time; it depends on the purpose of the experiment whether we seek high internal or high external validity. Typically there is a trade-off in which one kind of validity is sacrificed for the other. For example, laboratory experiments designed to answer a very focused question are often more internally valid than externally valid. Once a research question seems to have been settled (e.g., that poor exception handling is a major cause of software failure), then a move to a broader, more externally valid, experiment would be the right thing to do.

4. Example domain—keystroke biometrics

In this section we introduce the domain from which we draw concrete examples of experimental invalidities—keystroke biometrics.

Keystroke biometrics, or keystroke dynamics, is the term given to the procedure of measuring and assessing a user's typing style, the characteristics of which are thought to be unique to a person's physiology, behavior, and habits. The idea has its origin in the observation that telegraph operators have distinctive patterns, called *fists,* of keying messages over telegraph lines. One notable aspect of fists is that they emerge naturally, as noted over a hundred years ago by Bryan & Harter, who showed that operators are distinctive due to the automatic and unconscious way their personalities express themselves, such that they could be identified on the basis of having telegraphed only a few words [1].

These measures of key presses and key releases, based largely on the timing latencies between keystrokes, are compared to a user profile as part of a classification procedure; a match or a non-match can be used to decide whether or not the user is authenticated, or whether or not the user is the true author of a typed sequence. For a brief survey of the keystroke literature, see [7].

We use keystroke dynamics as an example here for two reasons. First, it's easy to understand—much easier, for example, than domains like network protocols. If we're going to talk about flaws and invalidities in experiment design, then it's better to talk about an experiment that's easily understood; the lessons learned can be extended to almost any other domain and experiment. Second, keystroke dynamics shares many problems with other cyber-security disciplines, such as intrusion detection. Examples are classification and detection accuracy; selection of best classifier or detector; feature extraction; and concept drift, just to name a few. Again, problems solved in the keystroke domain are very likely to transfer to other domains where the same type of solution will be effective.

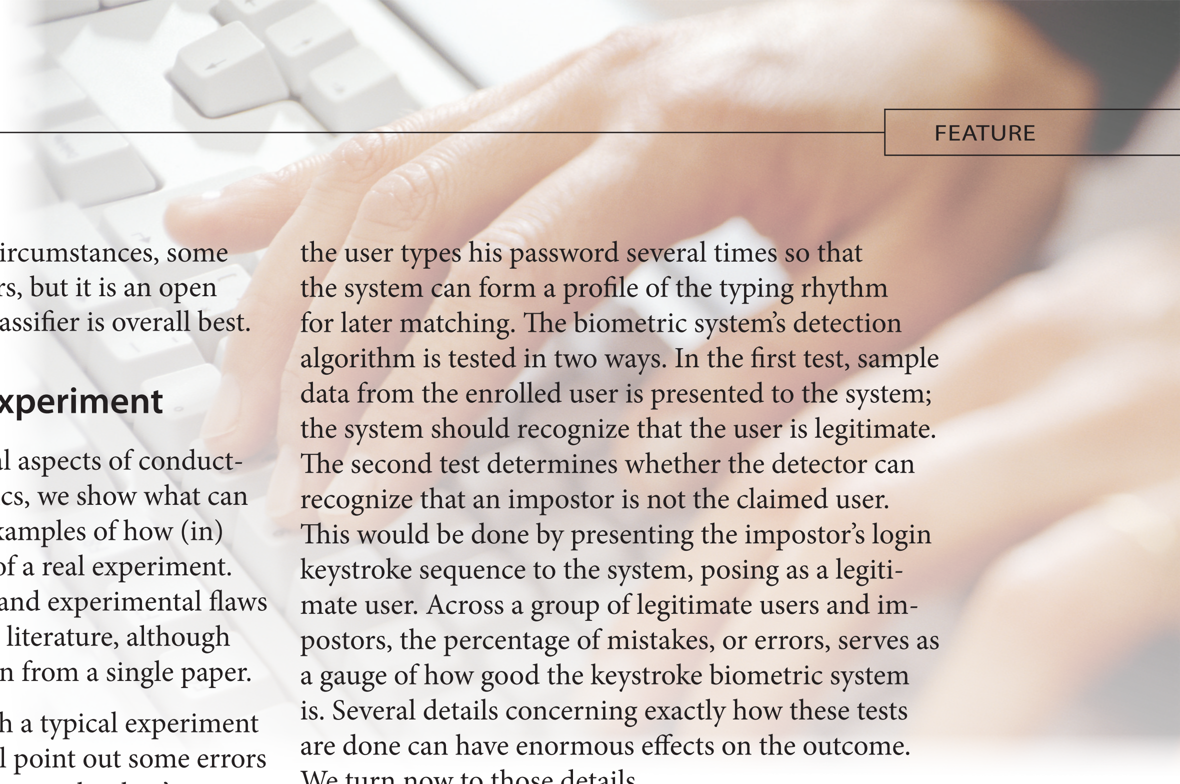
4.1. What is keystroke dynamics good for?

Keystroke dynamics is typically thought of as an example of the second factor in two-factor authentication. For example, for a user to authenticate, he'd have to know not only his own password (the first factor), but he would also have to type the password with a rhythm consistent with his own rhythm. An impostor, then, might know your password, but would not be able to replicate your rhythm, and so would not be allowed into the system. Another application, along a similar line, would be continuous re-authentication, in which the system continually checks to see that the typing rhythm matches that of the logged-in user, thereby preventing, say, insiders from masquerading as you. A third application would be what forensics experts call questioned-document analysis, which asks whether a particular user typed a particular document or parts of it. Finally, keystroke rhythms could be used to track terrorists from one cyber café to another, or to track a predator from one chat-room session to another.

4.2. How does keystroke dynamics work?

The essence of keystroke dynamics is that timing data are collected as a typist enters a password or other string. Each keystroke is timestamped twice; once on its downstroke and once on its upstroke. From those timings we can compute the amount of time that a key was held down (hold time) and the amount of time it took to transition from one key to the next (transition latency). The hold times and the latencies are called *features* of the typed password, and for a given typing instance these features would be grouped into a feature vector. For a 10-character password there would be eleven hold times and ten latencies if we include the `return` key.[a] If a typist enters a password many times, then the several resulting feature vectors can be assembled into a template which represents the central tendency of the several vectors. Each typist will have his or her own such template. These templates are formed during an enrollment period, during which legitimate users provide typing samples; these samples form the templates. Later, when a user wishes to log in, he types the password with the implicit claim that the legitimate user has typed the password. The keystroke dynamics system examines the feature vector of the presently-typed password, and classifies it as either belonging to the legitimate user or not. The classifier operates as an anomaly detector; if the rhythm of the typed password is a close enough match to the stored template, then the user is admitted to the system. The key aspect of this mechanism is the detector. In machine learning there are many such detectors, distinguished by the distance metrics that they use, such as Euclidean, Manhattan and Mahalanobis, among others [4]. Any of these detectors can be used in a keystroke

[a] There are two kinds of latencies—keydown to keydown and keyup to keydown. Some researchers use one or the other of these, and some researchers use both. In our example we would have 31 features if we used both.



dynamics system; under some circumstances, some detectors work better than others, but it is an open research question as to which classifier is overall best.

5. A typical keystroke experiment

In this section we discuss several aspects of conducting a study in keystroke dynamics, we show what can go wrong, and we share some examples of how (in) validity can affect the outcome of a real experiment. We will discuss some examples and experimental flaws that are drawn from the current literature, although not all of the examples are drawn from a single paper.

Walkthrough. Let's walk through a typical experiment in keystroke dynamics, and we'll point out some errors that we've observed in the literature, why they're errors, how to correct them, and what the consequences might be if they're left uncorrected. Note that the objective of the experiment is to discriminate among users on the basis of their typing behavior, not on the basis of their typing behavior plus, possibly unspecified, other factors; the typing behavior needs to be isolated from other factors to make the experiment valid.

A typical keystroke dynamics experiment would test how well a particular algorithm can determine that a user, based on his typing rhythm, is or is not who he claims to be. In a keystroke biometric system, a user would present himself to the system with his login ID, thereby claiming to be the person associated with the ID. The system verifies this claim by two means: it checks that the password typed by the user is in fact the user's password; and it checks that the password is typed with the same rhythm with which the legitimate user would type it. If these two factors match the system's stored templates for the user, then the user is admitted to the system.

Checking that the correct password is offered is old hat; checking that its typing rhythm is correct is another matter. This is typically done by having the user "enroll" in the biometric component of the system. For different biometric systems the enrollment process is different, depending on the biometric being used; for example, if a fingerprint is used, then the user needs to present his fingerprint to the system so that the system can encrypt and store it for later matching against a user claiming to be that person who enrolled. For keystroke biometric systems, the process is similar; the user types his password several times so that the system can form a profile of the typing rhythm for later matching. The biometric system's detection algorithm is tested in two ways. In the first test, sample data from the enrolled user is presented to the system; the system should recognize that the user is legitimate. The second test determines whether the detector can recognize that an impostor is not the claimed user. This would be done by presenting the impostor's login keystroke sequence to the system, posing as a legitimate user. Across a group of legitimate users and impostors, the percentage of mistakes, or errors, serves as a gauge of how good the keystroke biometric system is. Several details concerning exactly how these tests are done can have enormous effects on the outcome. We turn now to those details.

What can go wrong? There are several parts of an experiment where things can go wrong. Most experiments measure something; the measuring apparatus can be flawed, producing flawed measurements. If the measurements are flawed, then the data will be flawed, and any analytical results and conclusions will be cast into doubt. The *way* that something is measured can be unsound; if you measure code complexity by counting the number of lines, you'll get a numerical outcome, but it may not be an accurate reflection of code complexity. The way or method of taking measurements is the biggest source of error in most experiments. Compounding that error is the lack of detail with which the measurement methodology is reported, often making it difficult to determine whether or not something went wrong. We turn now to specific examples of methodological problems.

Clock resolution and timing. Keystroke timings are based on operating-system calls to various timers. In the keystroke literature we see different timers being used by different researchers, with timing accuracies often reported to several decimal places. But it's not the accuracy (number of decimal places) of the timing that's of overriding importance; it's the resolution. When keystroke dynamics systems are written for Windows-based machines (e.g., Windows XP), it's usually the tick timer, or *Windows-event clock* [6] that's used; this has a resolution of 15.625 milliseconds (ms), corresponding to 64 updates per second. If done on a Unix system, the resolution is about 10 milliseconds. On some Windows systems the resolution can be much finer if the QPC timer is used. The reason that timing resolution matters is not because people type as fast as one key every 15 milliseconds (66 keys per second); it's because the time *between* keystrokes can differ by less than 15 milliseconds. If some typists make key-to-key transitions faster than other ones, but the clock resolution is unable to separate them, then detection accuracy could suffer. One paper has reported a 4.2% change in error rate due to exactly this sort of thing [3]. A related issue is how you know what your clock resolution is. It's unwise to simply read this off the label; better to perform a calibration. A related paper explained how this is done in a keystroke dynamics environment [5]. A last word on timing issues concerns how the timestamping mechanism actually works; if it's subject to influence from the scheduler, then things like system load can change the accuracy of the timestamps.

The effect of clock resolution and timing is that they can interact with user rhythms as a confound. If different users type on different machines whose timing resolutions differ, then any distinctions made among users, based on timing, could be due to differences in user typing rhythms (timings) or they could be due to differences in clock resolutions. Moreover, since system load can influence keystroke timing, it's possible that rhythmic differences attributed to different users would be due to load differences, not to user differences. Hence we would not be able to claim distinctiveness based on user behavior, because this cannot be separated from timing errors induced by clock resolution and system load. If the purpose of the experiment is to differentiate amongst users on the basis of typing rhythm, then the confounds of clock resolution and system load must be removed. The simplest way to achieve this is to ensure that the experimental systems use the same clock, with the same resolution (as high as possible), and have the same operating load. This is possible in the laboratory by using a single system on which to collect data from all participants.

Keyboards. Experiments in keystroke dynamics require people to type, of course, and keyboards on which to do that typing. Most such experiments reported in the literature allow subjects to use whatever keyboard they want; after all, in the real world people do use whatever keyboard they prefer. Consequently, this approach has a lot of external validity. Unfortunately, the approach introduces a serious confound, too—a given keyboard, by its shape or character layout, is likely to influence a user's typing behavior. Different keyboards, such as standard, ergonomic, laptop, kinesis, natural, kinesis maxim split and so forth will shape typing in a way that's peculiar to the keyboard itself. In addition to the shape of the keyboard, the key pressures required to make electrical contact differ from one keyboard to another. The point is that not all keyboards are the same, with the consequence that users may type the same strings differently, depending on the keyboard and its layout. In the extreme, if everyone in the experiment used a different keyboard, you wouldn't be able to separate the effect of the keyboards from the effect of typing rhythm; whether your experimental results showed good separation of typists or not, you wouldn't know if the results were due to the typists' differences or to the differences among the keyboards. Hence you would not be able to conclude that typing rhythms differ among typists. This confound can be removed from the experiment by ensuring that all participants use the same (or perhaps same type of) keyboard. The goal of the experiment is to determine distinctiveness amongst typists based on their individual rhythms, not on the basis of the keyboards on which they type.

Stimulus items—what gets typed. Participants in keystroke biometrics experiments need to type something—the stimulus item in the experiment. While there are many kinds of stimuli that could be considered (e.g., passwords, phrases, paragraphs, transcriptions, free text, etc.), we focus on short, password-like strings. There are two fundamental issues: string contents and string length.

String contents. By contents we mean simply the characters contained in the password being typed. Two contrasting examples would be a strong password, characterized by containing shift and punctuation characters, as opposed to a weak password, characterized by a lack of the aforementioned special characters. It's easy to see that if some users type strong passwords, and other users type weak passwords, then any discrimination amongst users may not be solely attributable to differences among users; it may be attributable to intrinsic differences between strong and weak passwords that cause greater rhythmic variability in one or the other. The reason may be that strong passwords are hard to type, and weak ones aren't. So we may be discriminating not on the basis of user rhythm, but on the basis of typing difficulty which, in turn, is influenced by string content. To eliminate this confound, the experimenter should not allow users to choose their own passwords; the password should be chosen by the experimenter, and should be the same for each user.

String length. If users are left to their own devices to choose passwords, some may choose short strings, while others choose longer strings. If this happens, as it has in experiments where passwords were self-selected, then any distinctiveness detected amongst users cannot be attributed solely to differences among user typing rhythms; the distinctions could have been caused by differences in string lengths that the users typed, or by intrinsic characteristics that cause more variability in one length than in another. So, we don't know if the experimental results are based on user differences or on length differences. To remove this confound, the experimenter should ensure that all participants type same-length strings.

Typing expertise and practice. Everyone has some amount of typing expertise, ranging roughly from low to high. Expertise comes from practice, and the more you practice, the better you get. This pertains to typing just as much as it pertains to piano playing. Two things happen when someone has become practiced at typing a password. First, the total amount of time to type the password decreases; second, the time variation with which particular letter pairs (digrams) are typed diminishes. It takes, on average, about 214 repetitions of a ten-character password to attain a level of expertise such that typing doesn't change by more than 1 millisecond on average (less than 0.1%) over the total time (about 3–5 seconds) taken to type a password. At this level of practice it can be safely assumed that everyone's typing is stable; that is, it's not changing significantly. Due to this stability, it is safe to compare typists using keystroke biometrics. A classifier will be able to distinguish among a group of practiced typists, and will have a particular success rate (often in the region of 95–99%).

But what if, as in some studies, the level of expertise among the subjects ranges from low to high, with some people very practiced and others hardly at all? If practiced typists are consistent, with low variation across repeated typings, but unpracticed typists are inconsistent with high variability, then it would be relatively easy for a classifier to distinguish users in such groups from one another. This could make classification outcomes more optimistic than they really are, making them misleading at best. In one study 25 people were asked to type a password 400 times. Some people in the study did this, but others typed the password only 150 times, putting a potentially large expertise gap between these subjects. No matter what the outcome if everyone had been at the same level of expertise, it's easy to see that the classification results would likely be quite different than if there was a mixture of practice levels among the subjects. This is an example of a lack of internal validity, where the confound of differential expertise or practice is operating. There is no way that the classifier results can be attributed solely to users' typing rhythms alone; they are confounded with level of practice.

Instructions to typists. In any experiment there needs to be a protocol by which the experiment is carried out. This protocol should be followed assiduously, lest errors creep into the experiment whilst the researcher is unaware. Here we give two examples in which instructions to subjects are important.

First, in our own experience, we had told subjects to type the password normally, as if they were logging in to their own computer. This should be straightforward and simple, but it's not. We discovered that some subjects were typing with extraordinary quickness. When we asked those people if that's how they typed every day, they said no—they thought that the purpose of our experiment was to see who could type the fastest or the most accurately, even though we had never said that. This probably happened because we are a university laboratory, and it's not unusual in university experiments (especially in psychology) to have their true intentions disguised from the participant; otherwise the participant may game the experiment, and hence ruin it. People in our experiment assumed that we had a hidden agenda (we didn't), and the people responded to what they thought was the true agenda by typing either very quickly or very carefully or both. When we discovered this, we changed our instructions to tell subjects explicitly that there was no hidden agenda, and that we really meant it when we said that we were seeking their normal, everyday typing behavior. After the instructions were changed to include this, we no longer observed the fast and furious typing behavior that had drawn our attention in the first place. If we had not done this, then we would have left an internal invalidity in the experiment; our results would have been confounded with normal typing by some and abnormally fast typing by others. Naturally, a classifier would be able to distinguish between fast and slow typists, thereby skewing the outcomes unrealistically.

Second, if there is no written protocol by which to conduct an experiment, and by which to instruct participants as to what they are being asked to do, there is a tendency for the experimenter to ad lib the instructions. While this might be fine, what can happen in practice is that the experimenter will become aware of a slightly better way to word or express the instructions, and will slightly alter the instructions for the next subject. This might slightly improve things for that subject. However, for the subject after that, the instructions might change again, even if ever so slightly. As this process continues, there will come a point at which some of the later subjects are receiving instructions that are quite different from those received by the earlier subjects. This means that two different sets of instructions were issued to subjects, and these subjects may have responded in two different ways, leading to a confound. Whatever the classification outcomes might be, they cannot be attributed solely to differences in user typing rhythms; they might have been due to differences in instructions as well, and we can't tease them apart. Hence it is important not only to have clear instructions, but also to have them in writing so that every subject is exposed to exactly the same set of instructions.

6. What's the solution for all these problems?

All of the problems discussed so far are examples of threats to validity, and internal validity in particular. The confounds we've identified can render an experiment useless, and in those circumstances not only has time and money been wasted, but any published results run a substantial risk of misleading the readership. For example, if a study claims 99.9% correct classification of users typing passwords, that's pretty good; perhaps we can consider the problem solved. But if that 99.9% was achieved because some confound, such as typing expertise, artificially enhanced the results, then we would have reached an erroneous conclusion, perhaps remaining unaware of it. This is a serious research error; in this section we offer some ways to avoid the kinds of problems caused by invalidity.

Control. We use the term "control" to mean that something has been done to mitigate a potential bias or confound in an experiment. For example, if an experimental result could be explained by more than one causal mechanism, then we would need to control that mechanism so that only one cause could be attributed to the experimental outcome. As an example, the length of the password should be controlled so that everyone types a password of the same length; that way, length will not be a factor in classifying typing vectors. A second example would be to control the content of the password, most simply by having every participant type the same password. In doing this, we would be more certain that the outcome of the experiment would be influenced only by differences in people's typing rhythms, and not by password length or content. Of course while effecting control in this way makes the experiment internally valid, it doesn't reflect how users in the real world choose their passwords; certainly they don't all have the same password. But the goal of this experiment is to determine the extent to which individuals have unique typing rhythms, and in that case tight experimental control is needed to isolate all the extraneous factors that might confound the outcome. Once it's determined that people really do have unique typing rhythms that are discriminable, then we can move to the real world with confidence.

Repeatability and reproducibility (again). We earlier mentioned two important concepts: repeatability—the extent to which an experimenter can obtain the same measurements or outcomes when he repeats the experiment in his own laboratory—and reproducibility, which strives for the same thing, but when different experimenters in other laboratories, using similar but physically different apparatus, obtain the same results as the original experimenters did. If we strive to make an experiment repeatable, it means that we try hard to make the same measures each time. To do this successfully requires that all procedures are well defined so that they can be repeated exactly time after time. Such definitions are sometimes called operational definitions, because they specify a measurement in terms of the specific operations used to obtain it. For example, when measuring people's height, it's important that everyone do it the same way. An operational definition for someone's height would specify exactly the procedure and apparatus for taking such measurements. The procedure should be written so that it can be followed exactly every time. Repeatability can be ensured if the experiment's measurements and procedures are operationally defined and followed assiduously. Reproducibility can be ensured by providing those operational details when reporting the experiment in the literature, thereby enabling others to follow the original procedures.

Discovering confounds. There is no easy way to discover the confounds lurking in an experimental procedure. It requires deep knowledge of the domain and the experiment being conducted, and it requires extensive thought as to how various aspects of the experiment may interact. One approach is to trace the signal of interest (in our case, the keystroke timings and the user behaviors) from their source to the point at which they are measured or manifested. For keystroke timings, the signal begins at the scan matrix in the keyboard, traveling through the keyboard encoder, the keyboard-host interface (e.g., PS2, USB, wireless, etc.), the keyboard controller in the operating system (which is in turn influenced by the scheduler), and finally to the timestamping mechanism, which is influenced by the particular clock being used. At each point along the way, it is important to ask if there are any possible interactions between these waypoints and the integrity of the signal. If there are, then these are candidates for control. For example, keyboard signals travel differently through the PS2 interface than they do through the USB interface. This difference suggests that only one type of keyboard interface be used—either PS2 or USB, but not both. Otherwise, part of the classification accuracy would have to be attributed to the different keyboard interfaces. A similar mapping procedure would ask about aspects of the experiment that would influence user typing behavior. We have already given the example of different types of keyboards causing people to type differently. Countering this would be done simply by using only one type of keyboard.

Method section. A method section in a paper is the section in which the details are provided regarding how the experiment was designed and conducted. Including a method section in an experimental paper has benefits that extend to both reader and researcher. The benefit to the reader is that he can see exactly what was done in the experiment, and not be left to wonder about details that could affect the outcome. For example, saying how a set of experiment participants was recruited can be important; if some were recruited outside the big-and-tall shop, it could constitute a bias in that these people are likely to have large hands, and large-handed people might have typing characteristics that make classification artificially effective or ineffective. If this were revealed in the method section of a paper, then a reader would be aware of the potential confound, and could moderate his expectations on that basis. If the reader were a reviewer, the confound might provoke him to ask the author to make adjustments in the experiment.

For the experimenter the method section has two benefits. First, the mere act of writing the method section can reveal things to the experimenter that were not previously obvious. If, in the course of writing the section, the experimenter discovers an egregious bias or flaw in the experiment, he can choose another approach, he can relax the claims made by the paper, or he can abandon the undertaking to conduct the experiment again under revised and more favorable circumstances. If the method section is written before the experiment is done—as a sort of planning exercise—the flaws will become apparent in time for the experimental design to be modified in a way that eliminates the flaw or confound. This will result in a much better experiment, whose outcome will stand the test of time.

Pilot studies. Perhaps the best way to check your work is to conduct a pilot study—a small-scale preliminary test of procedures and measurement operations—to shake any unanticipated bugs out of an experiment, and to check for methodological problems such as confounded variables. Pilot studies can be very effective in revealing problems that, at scale, would ruin an experiment. It was through a pilot study that we first understood the impact of instructions to subjects, and subsequently adjusted our method to avoid the problems encountered (previously discussed). If there had been no pilot, we would have discovered the problem with instructions anyway, but we could not have changed the instructions in the middle of the experiment, because then we'd have introduced the confound of some subjects having heard one set of instructions, and other subjects having heard a different set; the classification outcome could have been attributed to the differences in instructions as well as to differences amongst typists.

7. Conclusion

We have shown how several very simple oversights in the design and conduct of an experiment can result in confounds and biases that may invalidate experimental outcomes. If the details of an experiment are not fully described in a method section of the paper, there is a risk that the flaws will never be discovered, with the consequence that we come away thinking that we've learned a truth (that isn't true) or we've solved a problem (that isn't really solved). Other researchers may base their studies on flawed results, not knowing about the flaws because there was no information provided that would lead to a deep understanding of how the experiment was designed and carried out. Writing a method section can help experimenters avoid invalidities in experimental design, and can help readers and reviewers determine the quality of the undertaking.

Of course there are still other things that can go wrong. For example, even if you have ensured that your methods and measurements are completely valid, the chosen analysis procedure could be inappropriate for the undertaking. At least, however, you'll have confidence that you won't be starting out with invalid data.

While the confounding issues discussed here apply to an easily-understood domain like keystroke biometrics, they were nevertheless subtle, and have gone virtually unnoticed in the literature for decades. Your own experiments, whether in this domain or another, are likely to be just as susceptible to confounding and methodological errors, and their consequences just as damaging. We hope that this paper has raised the collective consciousness so that other researchers will be vigilant for the presence and effects of methodological flaws, and will do their best to identify and mitigate them.

Richard Feynman, the 1965 Nobel Laureate in physics, said, "The principle of science, the definition almost, is the following: The test of all knowledge is experiment. Experiment is the sole judge of scientific 'truth'" [2]. Truth is separated from fiction by demonstration—by experiment. In doing experiments, we want to make claims about the results. For those claims to be credible, the experiments supporting them need first to be free of the kinds of methodological errors and confounds presented here.

References

[1] Bryan, W.L., Harter, N.: Studies in the physiology and psychology of the telegraphic language. Psychological Review 4(1), 27–53 (1897)

[2] Feynman, R.P., Leighton, R.B., Sands, M.: The Feynman Lectures on Physics, vol. 1, p. 1–1. Addison-Wesley, Reading (1963)

[3] Killourhy, K., Maxion, R.: The effect of clock resolution on keystroke dynamics. In: Lippmann, R., Kirda, E., Trachtenberg, A. (eds.) RAID 2008. LNCS, vol. 5230, pp. 331–350. Springer, Heidelberg (2008)

[4] Killourhy, K.S., Maxion, R.A.: Comparing anomaly-detection algorithms for keystroke dynamics. In: IEEE/IFIP International Conference on Dependable Systems and Networks (DSN 2009), pp. 125–134. IEEE Computer Society Press, Los Alamitos (2009)

[5] Maxion, R.A., Killourhy, K.S.: Keystroke biometrics with number-pad input. In: IEEE/IFIP International Conference on Dependable Systems and Networks (DSN 2010), pp. 201–210. IEEE Computer Society Press, Los Alamitos (2010)

[6] Microsoft Developer Network: EVENTMSG structure (2008), http://msdn2.microsoft.com/en-us/library/ms644966(VS.85).aspx

[7] Peacock, A., Ke, X., Wilkerson, M.: Typing patterns: A key to user identification. IEEE Security and Privacy 2(5), 40–47 (2004)

[8] Shadish, W.R., Cook, T.D., Campbell, D.T.: Experimental and Quasi-Experimental Designs for Generalized Causal Inference. Houghton Mifflin, Boston (2002)

[9] Taylor, B.N., Kuyatt, C.E.: Guidelines for evaluating and expressing the uncertainty of NIST measurement results. NIST Technical Note, 1994 Edition 1297, National Institute of Standards and Technology (NIST), Gaithersburg, Maryland 20899-0001 (September 1994)

About the author

Roy Maxion is a research professor in the Computer Science and Machine Learning Departments at Carnegie Mellon University (CMU). He is also director of the CMU Dependable Systems Laboratory where the range of activities includes computer security, behavioral biometrics, insider detection, usability, and keystroke forensics as well as general issues of hardware/software reliability. In the interest of the integrity of experimental methodologies, Dr. Maxion teaches a course on Research Methods for Experimental Computer Science. He is on the editorial boards of *IEEE Security & Privacy* and the *International Journal of Biometrics,* and is past editor of *IEEE Transactions on Dependable and Secure Computing* and *IEEE Transactions on Information Forensics and Security.* Dr. Maxion is a Fellow of the IEEE.