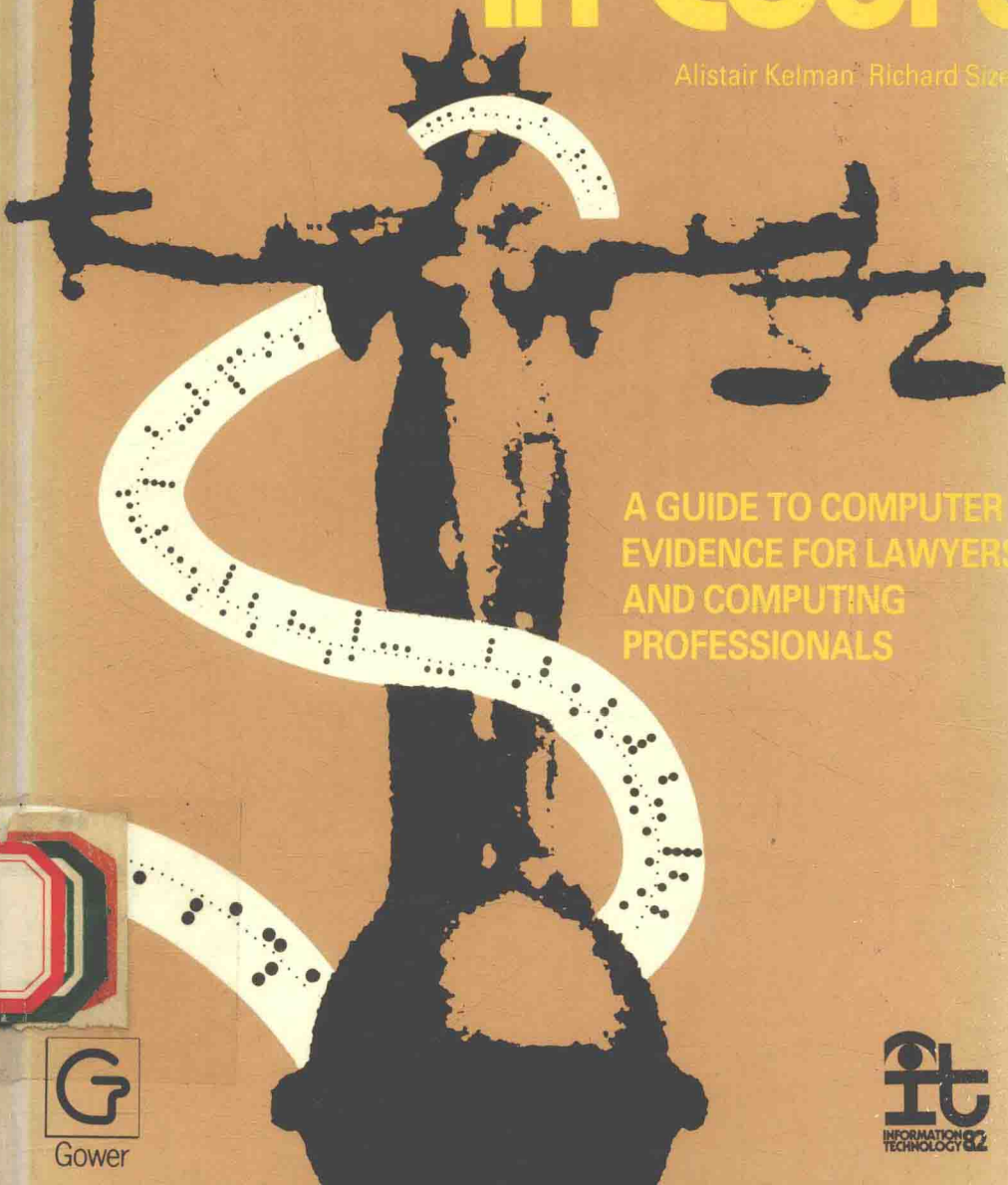
The Computer in Court

Alistair Kelman
Richard Sizer



A GUIDE TO COMPUTER EVIDENCE FOR LAWYERS AND COMPUTING PROFESSIONALS

Gower

INFORMATION TECHNOLOGY

The Computer in Court

A Guide to Computer Evidence for Lawyers and Computing Professionals

ALISTAIR KELMAN
RICHARD SIZER

Gower

Published by
Gower Publishing Company Limited
Gower House, Croft Road, Aldershot,
Hampshire GU11 3HR, England

British Library Cataloguing in Publication Data

Kelman, Alistair
The computer in court.
1. Evidence (Law) — England
2. Electronic digital computers
I. Title II. Sizer, Richard
344.207'6 KD7499

ISBN 0 566 03419 0

Printed and bound in Great Britain by
Biddles Ltd, Guildford and King's Lynn

Foreword

We are living through a revolution in the way information is handled. This is the inevitable consequence of the advances in microelectronic technology and it is certain that our methods of capturing, storing, retrieving and receiving information will be completely changed before the end of the century. The effect of the use of computers cannot be left undebated and the need for constructive criticism of the interface between the computer industry and the judicial system is apparent and essential for our system of justice to continue to work.

This book sits astride the interface and is the first stage in that debate. It explains the technology and its faults in as simple a manner as is practically possible. It explains the law of evidence in a similar way. It combines the two in an entertaining form by use of an imaginary court case. But its lighthearted approach should not be considered an indication of a lack of seriousness. For the book uses courtroom drama only to illustrate and illuminate very real problems that beset lawyers and computer professionals alike. Soon, because of the growth in the use of

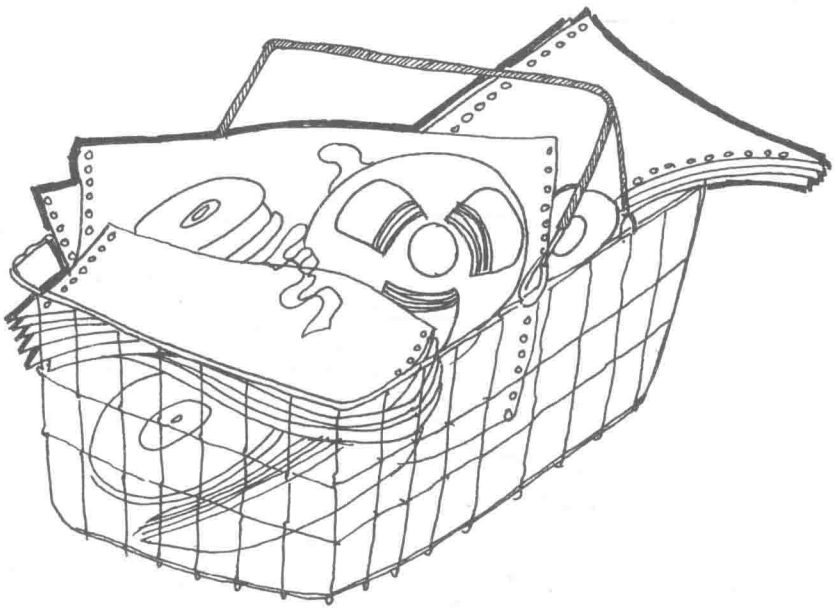
microcomputers and word processors, almost every document which comes before the courts will have been produced by a computer. Already lawyers need to know how to cross-examine computer personnel, what questions to ask to find latent errors in a computer's output. Computer professionals running computer installations need to appreciate the kind of cross-examination they may encounter when placing printouts in evidence. Computer professionals, such as myself, who find themselves called to give expert evidence in the courts need a limited understanding of the law of evidence and procedure. This book provides us all with the introductions we need.

I commend it as essential reading to lawyers and computer professionals alike. Its suggested method of proving that computer evidence is reliable is of particular importance. The gaps it indicates in the law caused by out-of-date statutes should be noted and filled at the earliest opportunity by Parliament. Above all the book is a filip to professional standards in the computer industry, something which I can fully endorse.

Douglas A. Eyeions
Director General
Computing Services Association

Contents

CHAPTER ONE
Faces of Computing



Computers can perform complex tasks, play games, draw pictures, solve incredibly difficult mathematical problems, and print the written word by the metre extremely quickly. They are now commonplace; some are expensive, many are cheap and a part of everyday life. A computer must be told how to perform any task. The instruction is carried out by a 'program' often called the 'software'. Understanding the difference between hardware (the boxes and the electronics therein) and software is important and germane to the purpose of this book. Some explanation of terms is given in Appendix A which also explains how the hardware and software interact with each other. The computer is inherently a moronically simple device in that it can only recognise one of two states such as yes or no, on or off. We show later how this works in practice. Accurate and reliable most of the time, computers can be wrong.

At one time to communicate with a computer was difficult; obscure 'languages' had to be learnt. However, great strides have been made in simplifying means of communication. Languages

approximating to English and other means, are now available to, and within the understanding of, most people.

Before dealing with the book's theme of evidence and computers it is of interest to examine in what ways computers have invaded the social structure. We do this by illuminating a number of different faces of computing: pedantic, provocative, Orwellian, factual, protective, logical, proliferative and skid-row.

Working with computers tends to make one pedantic, no doubt due to the fact that everything is seen in precise terms — black or white, yes or no, on or off (the authors are no exceptions). This can lead to problems of relationships between the computer specialist and the non-specialist even though the latter may be someone who dearly wishes to be a user of a computer. So we have the first face — the pedantic face — typified by the acknowledged inventor of the computer, Charles Babbage, in his note to Lord Tennyson after the latter's poem 'The Vision of Sin' was published. The note read:

> Sir,
> In your otherwise beautiful poem there is a verse which reads
>
> Every moment dies a man
> Every moment one is born.
>
> It must be manifest that if this were true, the population of the world would be at a standstill. In truth the rate of birth is slightly in excess of that of death. I would suggest that in the next edition of your poem you have it read
>
> Every moment dies a man.
> Every moment $1\frac{1}{16}$ is born.
>
> Strictly speaking this is not correct, the actual figure is so long that I cannot get it into a line, but I believe the figure $1\frac{1}{16}$ will be sufficiently accurate for poetry.
>
> I am, Sir, Yours etc. . . .

The provocative face of computing is illuminated by the words of Jean-Paul Parrot of the Canadian Union of Postal Workers delivered to the delegates of a computer conference in 1977:

> 'Computer systems are designed largely to increase the profits of corporations and reduce the numbers of workers. Computerisation may do wonders for the profit margins of

> the banks and trust companies, and it may greatly assist airline companies to inform businessmen in knowing their reservations, but this "information" does not provide workers with houses they can afford, food and clothing for their children, or safe and healthy places to work.'

The Orwellian face is illuminated by the news item concerning an alleged leakage of computer data from 4,000 US data centres concerned with medical data. Dr Gabrielli, head of a group specialising in the computerisation of medical records said:

> 'Computers have as much potential for good or ill as atomic energy. Unless privacy problems in handling computerised medical records are solved at once, 1984 is here already. We believe computers represent a tremendous threat to the basic human right to keep medical information private.'

The factual face is illuminated by a column in the New York Times when the initial results of the 'first-ever comprehensive computerised survey of data' was published on persons arrested in connection with the city-wide looting during the New York blackout of 1977.

Contrary to the widely publicised media analyses made immediately after the events, the computer analysis which appeared some weeks later presented a series of facts which overturned sociological theories, showing instead that many of those arrested were far from 'hopeless', having 'stronger community ties' and 'somewhat higher incomes' than those normally arrested in New York City. Specifically, the computer statistics showed that looting suspects had a rate of 45 per cent employment, 14·4 per cent high school and college enrolment, 10·4 per cent on welfare and 30·2 percent unemployed, but not on welfare. This compared with 30 per cent employment, 12·6 per cent high school and college enrolement, 15·7 per cent on welfare and 41·6 per cent unemployed, but not on welfare, in the general defendant population.

The protective face is illuminated by a column in the Guardian which stated:

> 'A man, convicted three times for indecent attacks on young girls, was allowed to become an "official uncle" to a ten-year-old girl in care whom he then assaulted. The judge

> asked what checks had been made and prosecuting counsel said a much wider check would have been made today because the information is now available on a regional basis. And very shortly there will be a national computer which will make available details of any conviction in any part of the British Isles.'

The logical face is illustrated by the case of the Dallas construction worker who was moving to a new construction job in Chicago. Two days before he was due to leave Dallas, his new contract was cancelled because he had failed to comply with Clause R-3 concerned with maternity — his record said he was pregnant. He called Chicago and claimed that there was a mistake — he was a man and could not be pregnant. Chicago were not impressed by the claim and refused to alter their record because the industrial health clinic which had performed the physical check-up had given a positive R-3 rating. The man called the clinic who checked the computer record and confirmed pregnancy.

The following three weeks can be glossed over; the solution is the main interest. The man eventually started work in Chicago but with an amended clause forbidding maternity benefit for the otherwise statutory period.

The proliferative face is illuminated by the cartoon which showed a young boy at a desk attempting his homework. The desk was piled high with pocket computers. By the boy's side sat an anxious mother saying 'Now we must get this right — if you take 12 computers away from 20 computers how many are left?'

Computers, alas, have a 'skid-row' face as illustrated by the following scene witnessed by one of the authors. On the outskirts of Gandia, a small town in the province of Valencia, Spain, a typical Spanish market was in progress. In the hot sun, sitting along the edges of the paseo was a line of vendors — gypsies selling wild garlic, others offering live snails, yet others with oranges, chickens, rabbits, water melons and so on. Between the garlic vendors and the live-chicken vendor a man had spread a colourful rug on part of which he sat. Piled on the rest of the rug was a collection of pocket calculators and personal computers. A piece of cardboard had scrawled on it '250 pts', but there

seemed little prospect of a sale. Every other vendor and shopper in sight had one of their own.

This book is about evidence from computers. It owes its existence to the effect that the decision by the Court of Criminal Appeal — that a computer-generated listing was not admissible as evidence — had on one of the authors who saw it as a potential threat to the judicial process.

The decision in January 1980 has since been reported in both the Criminal Law Review and in the official case reports. It is described later but we record here that it raises important, possible fundamental issues. These arise from the rapid growth of the computer industry with its almost impenetrable jargon and its associated specialist personnel (systems analysts, programmers, hardware engineers, operators) who have produced a problem that must be solved if trial by judge and jury is to remain respected as due process of law.

Lawyers can have difficulty in cross-examining computer industry personnel or highlighting hidden errors in computer outputs because they do not know what questions to ask. Yet the British legal system is founded on the basis of two lawyers asking questions and then arguing against each other in front of a judge and jury whose job it is to decide between the arguments. Today because of this lack of technical knowledge by many members of the legal profession certain events concerning the computer industry could be outside the law. This book tries to rectify this state of affairs. Lawyers can learn very quickly and we have presented our view of the law in a form that is familiar to both lawyers and laymen — an imaginary court case. Evidence is a huge subject and criminal and civil procedure can best be appreciated by considering examples.

In 1981 the authors produced a report,[1] which amongst other things, exposed the fallacy in the argument that it would always be possible for a lawyer to call an expert to testify as to the state of computer output, and that the state of accuracy would always be self-evident.

The report deals also with arguments for and against the need to modernise legislation and in fact contained a conclusion and recommendation that on balance there is probably a case for such amendment. The report stimulated debate as opinions differed, and still do, as to whether for example Section 1 of the

Criminal Evidence Act of 1965 should be extended to include computer records other than those from a trade or business. We believe that an opinion needs to be formed as to whether 'software' (the computer programs in all their forms) should be mentioned specifically. In the Civil Evidence Act 1968 the computer is defined only in terms of hardware — not a tenable definition as we attempt to show in this book.

Of particular importance, and the *raison d'être* in writing the book, is our belief that the computing and legal professions should jointly develop educational guidance and appropriate codes of practice. As a step towards these goals, the book has been written to assist members of the legal profession in correct examination of computer evidence, and to help computer professionals understand the needs of the courts, to anticipate likely courses of events and adapt their own professional activities accordingly. They could, after all, find themselves in the same position as our character, Professor Chocolate-Chip.

In our third chapter we give a brief outline of the law of evidence. Before leaving generalities we thought it appropriate to comment now on the procedures relating to criminal and civil cases so far as evidence is concerned. Non-members of the legal profession will find the comment useful background for the next chapter and for our court case described in chapters 4 to 7.

At the start of each Law Year the judges file into the Royal Courts of Justice, bewigged and gowned in a procession, following a tradition that is centuries old. Every day they administer justice according to law, following the strict rules of evidence, requiring statements to be proved by time-honoured processes, giving weight to each piece of conflicting evidence, determining questions of law and directing juries.

In a criminal case, before a prosecutor can begin to persuade a jury he must surmount the legal problems associated with the admissibility of evidence. A good defence counsel will take a number of procedural points while the jury is out of court to win the trial for his client at this stage. If he manages by any means to stop the prosecution adducing any evidence against his client, the judge has no alternative but to acquit the accused on the grounds of there being no case to answer. However, once evidence is held to be admissible the jury is recalled and it is put before them. The prosecution have to prove their case beyond

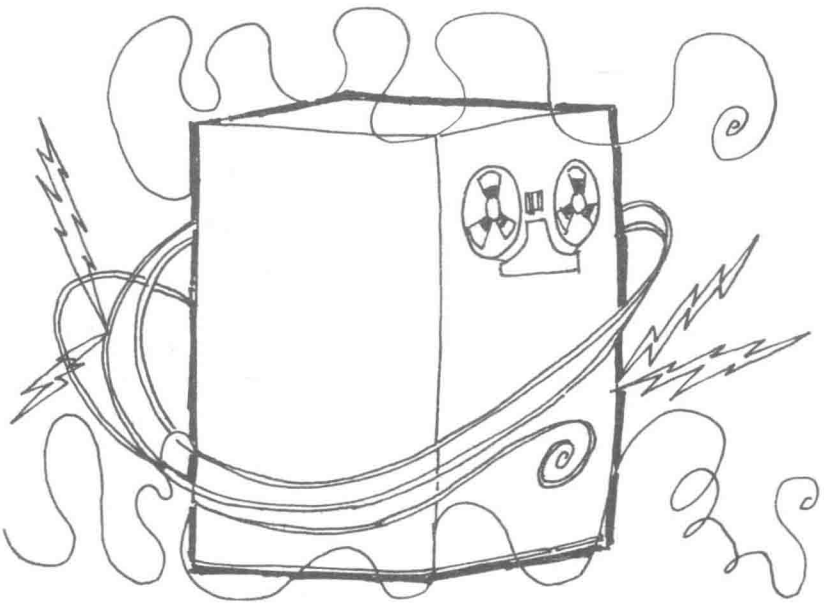
reasonable doubt and defence counsel attack this construction of the evidence, suggesting instead that the evidence is both unreliable and insufficient to support a conviction.

In civil cases, generally speaking, there is no jury and the judge decides questions of fact and law alone.

Reference:

(1) *Computer Generated Output as Admissible Evidence*. BCS 'Monographs in Informatics' Series, Heyden and Sons, 1982.

CHAPTER TWO
Examples of Computer Errors



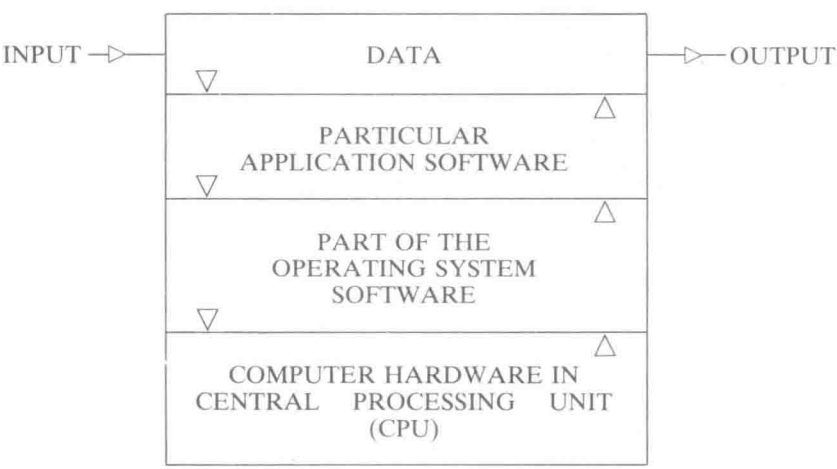
Errors in computer output can arise from a number of sources. The hardware or software may be faulty or may develop faults when interacting with other components in a particular computer network; the hardware may develop faults because it is working in an unsuitable environment (e.g. faulty air-conditioning); the software may be inadequately tested and contain hidden errors; faults may be produced by telecommunication lines used for transmission of data between computers in a network, and users may be inadequately trained. Most computer systems have safeguards that prevent undetectable 'corruption' of output. It is, however, possible for errors of a particular kind to bypass these safeguards.

In the course of manufacturing, rigorous quality and testing procedures are employed by most manufacturers. Nonetheless hardware can fail in service, often through external causes such as sudden high voltages or extreme heat. In America in the summer of 1980, the failure of a microprocessor within a large computer made by a reputable manufacturer resulted in two

nuclear alerts. It was a computer operator who eventually spotted the machine error and overrode its instructions.

Software can be divided into application software and operation system software. The application software communicates with the computer through its operating system software and trains the computer for a specific task. It is useful to consider a computer system as a multi-layer sandwich as shown in the following diagram. At the top is the data, either input or stored. The data is controlled by the application software and fed into the operating system software which converts it into the series of ones and noughts (binary notation) already described, which can be manipulated by the computer. The computer sends back its response, which is converted by the operating system software into a form understood by the application software, which then controls the formation of output data for the user. (See Appendix A for an explanation of the terms mentioned in this chapter.)

Both the application software and the operating system software can contain errors in themselves but errors can also be created by the way they interact with each other and the hardware. The operating system software is closely associated with the make of computer and the way that brand of computer



performs its operations. The application software can frequently be used on different makes of computer provided it works through a specific operating system's software. Off-the-shelf microcomputers used to be supplied with very little operating system software and often no application software. The user bought 'packages' of software to make his system more versatile and train it for specific tasks. This was like putting extra layers in the sandwich in the diagram. The rate of progress is such that operating systems and packages for micros are now available cheaply and in vast quantities from outlets (shops in the High Street) unheard of only a few years ago.

In the case of a large computer the operating system will usually be produced by the computer manufacturer. The purpose of the operating system is to control all jobs submitted to the computer, file handling, and messages passing between the computer and its peripherals, maintain a record of the processor and terminal time used by each user for accounting purposes; keep a record (log) of errors, functions and other operational details. Operating system software for computers is rarely in a finalised state and is constantly being updated and improved by the manufacturer. There is strict control over the distribution of amendments by the manufacturers who also provide managers of installations with advice and instructions on how the said amendments are to be implemented. A computer installation run in a professional manner will have tight documentary control over amendments, and management should know exactly which version of an operating system is running at a given time.

A claim that a given operating system is free from errors should never be taken seriously. Operating systems are tested by the supplier before delivery to the owners of the computer but errors can always be present which have not been detected by any of the tests carried out by the manufacturer. The likelihood of such errors being present in a given operating system decreases with time as it is used on various computer installations and errors are notified and rectified. Nevertheless it is always possible that an unusual set of circumstances may produce a computer error that was not discovered in the original testing procedures or spotted subsequently in use. The use of the 'latest release' of an operating system should be treated with caution until time (and progressively fewer errors) build up a level of confidence. More confi-