Breaking the Traditional Computer Security Research Barriers

Yvo Desmedt*

EE & CS Department, University of Wisconsin – Milwaukee
P.O. Box 784, Milwaukee, WI 53201, U.S.A.

Abstract. The security of networked computers must be dramatically improved. Other research disciplines may be useful in achieving this goal. Some topics being studied in computer security are being investigated in cryptography also. We overview some progress made in cryptography in these topics. We propose how increasing the use of cryptography in computer security can be helpful in designing more secure hardware and software for a future generation of computers.

1 Introduction

Today nobody would doubt that computers are an important aspect of our lives. Efforts to make computers more user-friendly (though the results in some projects are unfortunately poor) have resulted in the use of computers in many daily life applications, *e.g.*, many libraries[2] and railway companies are abolishing their paper (and microfiche) catalogs and paper timetables[3] with computerized ones. Presently a large portion of the population has used a computer directly and the number of users who have programming skills is increasing. By making our society more dependent on computers, the potential impact of computer crime, fraud and unreliability starts to become more frightening.

One could wonder what solutions against computer crime there are, taking into consideration that the computerization of our society is going to go on at a pace that is frightening. Indeed, some developments are frightening. In some countries the Registry of Births, Deaths and Marriages, has been computerized. If the abolishment of paper catalogs and paper railroad timetables is going to be followed by the abolishment of paper backups of birth certificates, death certificates and court orders (to rely solely on computers) then computers should be much better protected or a whole new range of computer crimes will be opened. To make matters even worse, paper documents will soon lose their legal value when color laser printers improve, reducing the legal value of paper backups. It

* Parts of the author's research were supported by NSF Grant NCR-9004879 and NSF Grant NCR-9106327.

[2] The library of the University of Wisconsin – Milwaukee has no longer updated its paper catalog and its paper list of periodicals since 1987 and 1990 respectively.

[3] For example, the Ferrovie Italiane Dello Stato (Italian Railway system) has replaced the national and international timetables in the main railway stations by computers.

Y. Deswarte et al. (eds.), *Computer Security - ESORICS 92*

may seem a challenge to securely achieve the above computerization, but computers are already being used in many other areas where security is even more critical. They control chemical plants, they are being used to design and produce electronic components and mechanical devices using CAD/CAM programs, they monitor air-traffic, etc. Unfortunately nobody has a watertight secure solution and it seems that a lot of ongoing research is necessary before we will achieve it.

Much research on the topic of computer security has already been done and has been reported in international conferences on the topic. To increase the value of the research in the area it may be constructive to consider research done in related disciplines. Such a widened view could enlarge the tools which may be used to achieve the goal of computer security. In this paper we consider which topics being studied in different computer science areas might be of interest to the researcher in computer security. This paper does not pretend to be complete. Indeed a lot of research on the psychology, social behavior and security awareness of computer users and computer criminals is definitely interesting, but we will not discuss this in depth.

In Sect. 2 we discuss briefly which areas related to computer science are useful in general. In Sect. 3 we overview some topics of computer security that are also being investigated by other research groups and compare the results and approaches followed. In Sect. 4 we discuss the potential impact of research in other disciplines on the future of computer security.

2 Influence from Other Areas of Computer Science

Computers have two aspects: hardware and software. Both hardware and software need to be protected. It always seemed easier to protect the first. Because chips are becoming more powerful it may seem that a lot of computer security problems could be solved by reducing them to hardware problems. However this is deceiving. Hardware is being developed using computerized techniques: CAD is being used to design chips and the production of chips is also computerized. Computer crimes (*e.g.*, using computer viruses) could target CAD (and CAM) programs [12]. From a first glance a major advantage of hardware is that it is hard to modify and definitely hard to modify at a distance. However this is a misconception. Embedded hardware could maliciously be completely different from the specified one (specs). Indeed it is considered very difficult (NP-hard) to check whether the embedded chip has the functionality of the specs. This allows a designer to modify the behavior of a chip at a certain time. So one could conclude that trying to achieve secure computers by relying solely on hardware is the wrong approach. Worse yet, hardware security will become a new problem child.

Some research topics in the area of Computer Architecture may be of interest. It is well known that the research on Fault-Tolerant computers is a very important topic as regards obtaining more reliable computers, *i.e.*, reducing the impact of an *accidental* action. It is not immediately clear whether it could help

to protect computers against a *malicious* intended action. In Sect. 3.3 we will indicate that in some cases it does help.

Artificial Intelligence has been used to detect computer crime [10, 33]. Most users type at a specific speed and/or use only a specific subset of the available operating system commands. If a criminal has broken into a computer he might be using other commands and will likely be detected. To be useful the number of false alarms should be minimal. The security obtained is not likely to be enormous, but it seems to be better than the protection provided by a system which only uses passwords (for a discussion about secure identification see also Sect. 3.1). Some Automatic Teller Machine (ATM) systems monitor the distribution of cash. If the amount withdrawn in an area is unusual, the system shuts itself down and an alert is given. Statistics on the distribution of cash depending on the day of the week, the day of the month and the day of the year is being used to define what unusual is.

It is well known that the theory of computation has had an impact on the research on computer security since its early stage, *e.g.*, [9, pp. 240–248, 341]. Traditionally the theory of computation has been used to demonstrate that solving some problems is hard and therefore the theory does not seem to be able to contribute constructively to secure computers. In Sect. 3 we will indicate the need for another attitude.

From all areas it seems that the development towards workstations and PCs (Personal Computers) interlinked by a computer network combined by techniques from cryptography could be the most influential on computer security. To be able to discuss this in sufficient detail we first discuss topics that are being studied by researchers in computer security and other disciplines.

3 Same Research Problem: Two Approaches

Many important topics that are being investigated in computer security are being studied in other areas.

The discipline which seems the closest to computer security is cryptography. While the original goal of cryptography was to obtain secret communication [9], the goal of modern cryptography is information integrity [46]. In other words:

> cryptography is the science and study of protecting data in its raw form against malicious actions.

Having seen this definition, one could say that:

> computer security is the science and study of protecting data on a computer against malicious actions,

and that:

> communication security is the science and study of protecting data being communicated against malicious actions.

Although cryptography and communication security seem equivalent, they are not. The latter includes protection against jamming (such as spread spectrum techniques) and is also strongly related to communication reliability (*e.g.*, to protect against an adversary who destroys communication (switching) nodes and/or links). In view of the above definitions of computer security and cryptography, it is no surprise that there are many research topics that overlap. From those topics the most important is the problem of identification.

The approach followed in *modern* cryptography is completely different from the heuristic one followed a few years ago. When a security problem must be solved one defines (or one tries to define) formally what the goal is. Most of these definitions specify first how the system should work when everybody is "honest" and secondly that a malicious attacker will not succeed in performing a class of attacks. A cryptographic system is proven secure when one can mathematically prove that the system satisfies the security definition based on a physical assumption, *e.g.*, the existence of randomness in nature, or on a computational complexity assumption. So, while in other areas of computer science computational complexity is used to find the *limitations* of what can be done, modern cryptography often *uses* complexity to *achieve* its goal.

Let us discuss some overlaps between the two areas in more detail and in Sect. 4 we discuss what the potential impact of modern cryptography on computer security might be.

3.1 Identification

Identification is a process we are all very familiar with. Driver's licenses, passports and cards that are used for granting access to facilities are all identification tokens. The goal of a computer login process is to identify the user. Secure computer login has been studied for a long time, *e.g.*, [36, 51].

Although no satisfactory formal definition of identification has been given, there are several informal ones, such as [13] (partially based on [22, p. 186]):

> In a secure identification system at least one trusted center knows which unique individual corresponds with a certain public ID. Based on his ID A is able to convince B that he is A, but B can not convince others that he is A.

From the above definition it is clear that password-based login is insecure. Indeed when a user A identifies himself to a machine B, then B (or the system manager of B) could claim to be A later on. A major problem with password-based login is that computers are interlinked in networks, so passwords cross insecure computers. Moreover, the link between the terminals and the computer have always been insecure, *i.e.*, for people inside the building it is very easy to eavesdrop and now with terminal servers on an ethernet cable going to offices to serve workstations it is even simpler.

To avoid this problem Fiat and Shamir [22, 21] proposed to use zero-knowledge protocols [24]. In a zero-knowledge protocol (of knowledge) a prover proves that

he knows some secret s corresponding to some public number x, without revealing any computational knowledge about s. Evidently for it to be useful for identification purposes it is necessary that it be hard for somebody not knowing s to claim he knows. A real-time fraud [2, p. 177] is possible in which the proof is forwarded and a dishonest person "parasitizes" an honest prover. So while A is proving he is A to B, B helps C, in real-time, to prove to D the false claim that C is A. Solutions to avoid this problem have been proposed [3, 2]. The remaining problem is that to perform this zero-knowledge proof a token is used, so in fact it is the token which proves to be the individual. Thus, linking the token with the actual individual is not 100% secure [2, p. 178].

3.2 Covert Channels

Lampson [29, p. 614] discussed covert channels, which he defined as:

> *Covert* channels, i.e. those not intended for information transfer at all, such as the service program's effect on the system load.

A more general definition can be found in [49, p. 110].

In 1983, Simmons [42] discovered that a secret message can be hidden inside the authenticator. He called this "hidden" communication channel the *subliminal channel*. To make the problem easier to understand, Simmons illustrated the subliminal channel by comparing it with two prisoners who are communicating authenticated messages in full view of a warden. The warden is able to read the messages. The subliminal consists in hiding a message in the authenticator such that the warden cannot detect its use nor read the hidden part. In some authentication systems the messages sent are observable by the warden, such as messages originating in the verification of treaties [45]. Subliminal channels were also introduced inside other signature systems, *e.g.*, [25, 44, 43].

It is debatable whether subliminal channels are covert channels or not. Indeed, messages are transmitted in authentication systems, so such systems are intended for information transfer and therefore subliminal channels do not satisfy the Lampson definition. However something additional is transferred which was not intended. Indeed, the authenticator should just be a function of the key and the message, and the bits of the authenticator should not hide an additional message.

A Massive amount of research has emphasized how to analyze covert channels and how to limit them, *e.g.*, [27, 26, 35]. In the context of verification of treaties Simmons [41, p. 20] analyzed how some of these channels could be *prevented* in a communication system transmitting authenticated messages that are observable by a warden. He mentioned that the analog aspect of a modulated message could be used to hide a covert channel. His solution against this was that the "warden", who is located between sender and receiver, would convert the signal back to digital and then back to analog. Solutions against subliminal channels in authentication and signature systems have been proposed [15, 11, 19]. The proposed solutions guarantee that either the warden will detect an attempt to

send a subliminal message or, if he does not, then the receiver will not be able at all to observe one either. The basic idea to achieve this last aspect is that the warden is active and modifies the data strings to enforce a proper probability distribution (of what is being sent). In some of these subliminal-free schemes the participants have to prove, using zero-knowledge proofs, to the warden that they actually executed the deterministic aspect of the protocol as specified. The proofs must be zero-knowledge (see also Sect. 3.1) because the warden is potentially dishonest as is clear from Simmons' prisoners' example. In the context of verification of treaties, where devices owned by A's country measure in B's country the compliance of B to the treaty and send these authenticated message back home, the warden being the counterintelligence of B has definitely a potential interest in trying to obtain the secret key of A's device to impersonate it.

Research has extended the concept of subliminal channels to a more general context than authentication [18, 16]. Many cryptosystems (besides authentication system) can now be made subliminal-free [17, 11, 6, 5]. However not all systems (cryptosystems and others) can be made subliminal-free [19] indicating that the techniques used are not going to prevent all covert channels.

3.3 Threshold Schemes and Reliability

The topic of reliability seems unrelated to security, because the first reduces the impact of accidents while the second has to deal with malicious actions. However, compromises between reliability and security are often necessary. Let us illustrate this.

How do you keep a backup of a secret key, k, *e.g.*, a password[4] or PIN for a password-based identification system (see also Sect. 3.1)? The worst method is to write it down somewhere. The best method is to first choose $l-1$ ($l \geq 2$) random numbers $r_1, \ldots, r_{l-1}$ modulo n, n being the bases (radix) in which the key is written (there are chips that generate truly random numbers [20, 48]). Then calculate $r_l = k - \sum_{i=1}^{l-1} r_i \bmod n$ and keep all the numbers r_i ($1 \leq i \leq l$) stored at completely different locations. Recovering k when necessary from all r_i is easy.

Now, what happens if in the above system you forgot where you stored one of the r_i or the media on which you stored it was destroyed? Clearly you may lose your key. So a compromise between security and reliability is necessary. Threshold schemes [4, 40] (for a bibliography on the topic until 1990 consult [47, pp. 492–497]) achieve this compromise. However the threshold idea has some deeper security aspects. A parliament makes new laws based on a majority threshold and in many other organizations too, majority rules. From this arose

[4] It is a myth that you should never make a backup of your password. Because governments, banks, telephone companies make incompatible systems you are forced to remember a massive number of PIN numbers, which is impossible. Moreover some system managers do not allow you to choose your own password on a computer. So a backup is necessary, but one should explain to people how to do it appropriately.

probably the idea that one can trust a majority, or a number of people equal to or above a certain threshold, while one would not trust a group of cardinality less than the threshold.

Goldreich-Micali-Wigderson [23] used the idea of threshold to be able to play any game over the telephone assuming that a majority of the players is honest. As observed in [14] this concept is useful for example when one wants that many individuals be involved in making a digital signature (*e.g.*, a law), or that one individual cannot read an encrypted message, but many can (see also [31]). The idea that a threshold of computers is "honest" was recently used to propose a theoretical solution against viruses [34]. The paper starts from the assumption that computers recover from viruses and that a sufficient number of computers are linked on the network that are either not infected or recovered.

3.4 Other Topics

A lot of other topics that have originated from computer security have been studied in cryptography, *e.g.*, computer viruses as we already mentioned partially (see also [12, 1, 50]). Nowadays (cryptographic) protocols are an important topic.

Many topics that have originated from cryptography are being used and studied in computer security, such as: authentication, privacy, key distribution, etc.

4 Potential Impact of Other Disciplines

To study how research in other disciplines can help make more secure computers let us compare computers to cities. In a city (computer) a multiplicity of tasks are being performed. Often there is communication between cities (computers). During the Middle Ages (the first days of computers) the need to communicate between cities (computers) was small. Due to the small number of inhabitants (users) of these cities (computers) one could easily trust another and the main threat came from outside. Fortified walls were built to protect against outsiders, therefore cities (computers) were hard to access. However, the inhabitants (users) were vulnerable to internal crimes, though they were rather rare and the criminals were often found.

Due to the development of trade (technology) the need to easier access to cities (computers) was necessary (attractive). At the same time, due to the growth in population (popularity), cities had more inhabitants (users and programs). The effect is well known: the crime rate went up, probably due to lack of checks and opening of communications, *e.g.*, the crime rate in the U.S., where no internal boarder control exists is much higher than in the present-day E.C. So the natural question is: can we not go back to small fortified cities (computers) and, if we do this, will we obtain improved security. Let us try to answer this question in the computer world.

Due to hardware development one can easily foresee that computers will mostly be used by a single user (virtually more users could be allowed: the

same user being "root" and the same user being "user"). The question then is why do we need multi-user environments? An obvious answer is that many applications need multi-user environments: *e.g.*, databases, joint development of programs, etc. However the rate of progress in the speed of computer networks is much higher than the rate of progress in the number of CPU instructions per second [28]. So a scenario in which there is *no* need for a user to have accounts on different computers than his own is not excluded. The question is then how a user being absent from his office can use a local facility to print data, read e-mail, update his data (*e.g.*, bookkeeping), etc. Will a traveling user be forced to carry around a laptop much less powerful than his workstation? Let us first describe how fast new technology and fast networks will allow us to solve this problem. In the meanwhile we start to discuss how cryptography can help increase computer security by building virtual walls around computers, and we will then discuss what aspects remain vulnerable.

Suppose a user, A, wants to use his workstation (due to our assumption he will not be able to login on another one) when traveling. He takes his electronic notebook, types in his PIN (the PIN protects against fraud due to a lost or stolen notebook) and connects it then to a "networked-screen". A networked-screen is similar to an intelligent (graphical) terminal and is connected to a worldwide network, *but* it has *no* keyboard. The notebook makes, through the networked-screen, a link to the owner's workstation. The user does not login to his workstation. Instead, the notebook sends, using a one-time-valid authentication system [37, 13], the message: "I, A, am identifying myself to B," where B is the network address of the workstation. Hereto the notebook contains A's secret key. Because the authenticated message is only valid once, replay is excluded. Now A uses his notebook as a keyboard. The notebook authenticates, using the one-time-valid authentication system, all the commands typed by A. The output of his workstation is displayed on the networked-screen. The mouse could either be connected to the screen, or be a ball on the notebook. When the user is home, his notebook is still the keyboard (so his workstation has *no* keyboard!).

If a lot of typing has to be done a networked-terminal can be used. To avoid lowering the security too much, most operating system commands cannot be executed when typed from the keyboard of a networked-terminal, but the keyboard of the electronic notebook is necessary to have the commands be authenticated. So to start editing a file the command "edit <filename>" has to be typed on the notebook; the user could type the editor commands from the networked-terminal. If the editor allows editing many files at once, each filename has to be entered from the notebook. It is clear that this is less secure than the networked-screen approach.

When more progress has been made on displays the notebook could have its own, resulting in a downgraded laptop without disc and fast CPU, but resembling a terminal more. The output of the workstation could then be authenticated too, increasing the security dramatically. (If the user wants to have the privacy of his office (home), the communication link could be protected using encryption.) Nowadays nodes on networks must be at a fixed location. Due to new research it

will be possible to have these nodes moving around. Then contemporary laptops (with disc and diskettes, etc.) will be obsolete.

To enhance the user-friendly aspect, the electronic notebook could be used to access ATMs, telephones and to replace the many plastic (chip) cards by just one. However the risk of losing the notebook becomes enormous, even when it is protected by a PIN. Therefore *one* standard chipcard (or a more secure version) should be inserted to critical applications. This standard chipcard could be the user's identity card or his electronic passport [8]. The user should however never have to type a PIN or a password (besides the *one* PIN to start up his notebook). To obtain the desired security the user should keep his standard chipcard separate from his notebook.

So the above approach advocates the elimination of the practice of giving a user accounts on multiple computers. A question is whether a user really could need those. When a user has to consult a database, there is no need to access all operating system commands. Instead his workstation should be a client and the database a server. The search commands are authenticated (or digitally signed) by the electronic notebook. The database checks if the user has this privilege and if so allows him to execute the specific read (search) command. Unfortunately some databases must be updated very frequently, *e.g.*, airline reservation systems, stock exchange information, etc. So multiple users must have write access, with all the resulting security problems (larger cities are also less safe!). *However it is not because some environments are multi-user oriented and that in the early ages of computers the cost of computers was so high that only multi-user ones were around, that therefore we have to stick with the multi-user concept.*

The security of single-user computers could be enhanced using cryptography, in particular authentication (which has often[5] been proposed to enhance computer security). Observe that the role of identification in the above has merely vanished, which it should have a long time ago. Indeed a user who is logged in on a modern computer over a network is vulnerable to the most unbelievable attacks. The computers through which his communication goes (many are very insecure ones) are able to insert all possible commands without displaying them, *e.g*, deleting half of his files, etc. By reducing multi-user computers, many (not all) covert channels are at once eliminated. Some potential subliminal channels remain, but in critical circumstances these can be eliminated as discussed in Sect. 3.2.

The single-user approach definitely does not eliminate all computer security problems. There are many remaining problems. The user has to trust the hardware of his workstation and electronic notebook. Moreover the problem that any electronic equipment radiates remains and needs to be solved using proper shielding techniques (which are not obvious). Validation by a trusted authority of hardware is required, but as we mentioned earlier, if a lot of freedom is given to the designer, it is not an easy task!

[5] Unfortunately the concept of authentication has often been confused with error-correcting codes, which are two completely different topics.

The programs the user uses may contain trapdoors that allow circumventing the security measures. To prevent this the authentication should be developed in hardware so that only the legitimate user can access his workstation. However, if conventional cryptography is used, then the workstation contains the secret key and the program may succeed in having a fraudulent order being authenticated. If digital signatures are used the key stored in the workstation to verify the authenticity is different from the one used to generate the signature. So this problem is then eliminated (assuming that there is no conspiracy between the designer of the electronic notebook, where the secret key is stored, and the designer of the workstation/program).

To increase the security, and due to progress in VLSI, the disk controller could take over some roles that the software operating system has today. The commands to format the disk, divide it in partions, etc. could then be authenticated. So the disk controller checks if these instructions have been authenticated before performing these (higher level) jobs (which are now under hardware control). On a "single-user" computer the disk space could be divided securely into four types of files:

1. "root" files which are never modified. The disk drive will never allow these to be modified; read-only optical disks could be used to achieve this, but modern implementations are often slow.
2. "root" files that are often modified. Software (operating system) based security measures would be used to protect these files.
3. manually produced user files (source files, text files, script files, etc). These files need authenticated permission from the user to be modified. The disc controller would check this authentication. A mixture of signature and authentication techniques could be used to obtain acceptable speed and security.
4. non-manually produced user files (the mail spool file, data files which are the output of a program, etc). These would be protected using software methods.

Although the potential impact of time-bombs hidden in programs and viruses has been reduced, it has not been eliminated. Validation again may be the answer, but it is again not easy. To make this more realistic, standard programs should be developed and the concept of making new versions of operating systems and updates should almost be abolished so that the work of validation does not have to be done over and over again. Many users do not need frequently updated operating systems, moreover the concept is awful. To illustrate this, try to explain the concept of updates and new versions to a mechanical engineer who comes back to earth after having been absent for a few decades. The best is to tell him that mechanical engineering has made a lot of progress: each year or so, the motor of cars is taken out and replaced by a brand new designed one. It is likely he will consider this a devolution!

5 Conclusion

Modern computers have many security problems. The approach of trying to fix those has been analyzed for — in light of the results achieved — too long a time. Many problems are either not solvable or the heuristics that have been used have failed. So, it has been clearly demonstrated that security is not an added feature, but must be taken into account before designing the hardware and software of the computer. Instead of trying to fix the problem, it may be better to throw (almost) everything away and start from scratch. This would include:

- to reduce the power of computers (*e.g.*, having "single-user" workstations instead of multi-user ones as proposed in Sect. 4),
- to design the hardware of the computer to support security as much as possible using cryptographic techniques,
- to avoid "Open Sesame" security (as identification is) but to authenticate and verify each command (and sub-commands),
- to avoid software based security,
- and finally to compromise on security only when there is no known solution or when the resulting computer would be completely impractical.

In other words, it may be better to redefine what a computer is such that it can be sufficiently protected. One could question if it is fair to change the goal of the research. Indeed by modifying the definition of a computer so that they can be made secure, it seems that this is nothing else than cheating! However it may be better to have in fifteen years time some secure hardware-software around (which one might still call a computer) than having updated versions of todays computers that became useless due to the increase in computer fraud (caused directly or indirectly by internetworking, viruses, and the fact that there will be more trained users). It is clear that many other ideas than those in Sect. 4 must be presented and that a lot of research is necessary to achieve this goal.

Useful to the above research are topics and results which have been achieved in other areas and we mentioned a few. Due to the definition of cryptography, it seems to be the most potentially helpful discipline around but one should not exclude influence from other areas. One may not forget that most research topics are influenced by other areas, *e.g.*, the research on crytography is influenced by hardware considerations, theoretical computer science, information theory, computer security, etc.

A cynical observation would be appropriate here to avoid becoming overly enthusiastic. A lot of workstations used mainly by a single-user are idle most of the time. Cryptanalysts have started to use this computer time (asking permission) to obtain massive computer time [7, 30, 38, 32] (for futuristic and overview papers consult [39, 50]). It is believed that our brain also uses the concept of massive parallel computation and so allowing multiple users to use a computer may again become attractive. But still then some of the proposed ideas are compatible with the multi-user environment if digital signatures, instead of conventional authentication schemes, are used.

References

1. Adleman, L. M.: An abstract theory of computer viruses. In Advances in Cryptology — Crypto '88, Proceedings (Lecture Notes in Computer Science 403) (1990) S. Goldwasser, Ed. Springer-Verlag pp. 354–374.
2. Bengio, S., Brassard, G., Desmedt, Y. G., Goutier, C., Quisquater, J.-J.: Secure implementations of identification systems. Journal of Cryptology **4** (1991) 175–183.
3. Beth, T., Desmedt, Y.: Identification tokens — or: Solving the chess grandmaster problem. In Advances in Cryptology — Crypto '90, Proceedings (Lecture Notes in Computer Science 537) (1991) A. J. Menezes and S. A. Vanstone, Eds. Springer-Verlag pp. 169–176.
4. Blakley, G. R.: Safeguarding cryptographic keys. In Proc. Nat. Computer Conf. AFIPS Conf. Proc. (1979) pp. 313–317.
5. Burmester, M., Desmedt, Y. G., Itoh, T., Sakurai, K., Shizuya, H.: Divertible and subliminal-free zero-knowledge proofs of languages. Submitted for outside publication.
6. Burmester, M. V. D., Desmedt, Y.: All languages in NP have divertible zero-knowledge proofs and arguments under cryptographic assumptions. In Advances in Cryptology, Proc. of Eurocrypt '90 (Lecture Notes in Computer Science 473) (1991) I. Damgård, Ed. Springer-Verlag pp. 1–10.
7. Caron, T. R., Silverman, R. D.: Parallel implementation of the quadratic sieve. J. of Supercomputing **1** (1988) 273–290.
8. Davida, G. I., Desmedt, Y. G.: Passports and visas versus IDs. Computers & Security **11** (1992) 253–258.
9. Denning, D. E. R.: Cryptography and Data Security. Addison-Wesley Reading, MA 1982.
10. Denning, D. E. R.: An intrusion-detection model. IEEE Transactions on Software Engineering **SE-13** (1987) 222–232.
11. Desmedt, Y. G.: Abuse-free cryptosystems: Particularly subliminal-free authentication and signature. Submitted to the Journal of Cryptology, under revision April 1989.
12. Desmedt, Y.: Is there an ultimate use of cryptography? In Advances in Cryptology, Proc. of Crypto '86 (Lecture Notes in Computer Science 263) (1987) A. Odlyzko, Ed. Springer-Verlag pp. 459–463.
13. Desmedt, Y.: Major security problems with the "unforgeable" (Feige-)Fiat-Shamir proofs of identity and how to overcome them. In Securicom 88, 6th worldwide congress on computer and communications security and protection (March 15–17, 1988) SEDEP Paris France pp. 147–159.
14. Desmedt, Y.: Society and group oriented cryptography : a new concept. In Advances in Cryptology, Proc. of Crypto '87 (Lecture Notes in Computer Science 293) (1988) C. Pomerance, Ed. Springer-Verlag pp. 120–127.
15. Desmedt, Y.: Subliminal-free authentication and signature. In Advances in Cryptology, Proc. of Eurocrypt '88 (Lecture Notes in Computer Science 330) (May 1988) C. G. Günther, Ed. Springer-Verlag pp. 23–33.
16. Desmedt, Y.: Abuses in cryptography and how to fight them. In Advances in Cryptology — Crypto '88, Proceedings (Lecture Notes in Computer Science 403) (1990) S. Goldwasser, Ed. Springer-Verlag pp. 375–389.
17. Desmedt, Y.: Making conditionally secure cryptosystems unconditionally abuse-free in a general context. In Advances in Cryptology — Crypto '89, Proceedings

(Lecture Notes in Computer Science 435) (1990) G. Brassard, Ed. Springer-Verlag pp. 6–16.

18. Desmedt, Y., Goutier, C., Bengio, S.: Special uses and abuses of the Fiat-Shamir passport protocol. In Advances in Cryptology, Proc. of Crypto '87 (Lecture Notes in Computer Science 293) (1988) C. Pomerance, Ed. Springer-Verlag pp. 21–39.
19. Desmedt, Y., Yung, M.: Unconditional subliminal-freeness in unconditional authentication systems. In Proceedings 1991 IEEE International Symposium on Information Theory (Budapest, Hungary, June 24–28, 1991) p. 176.
20. Fairfield, R. C., Mortenson, R. L., Coulthart, K. B.: An LSI random number generator (RNG). In Advances in Cryptology. Proc. of Crypto 84 (Lecture Notes in Computer Science 196) (1985) G. R. Blakley and D. Chaum, Eds. Springer-Verlag pp. 203–230.
21. Feige, U., Fiat, A., Shamir, A.: Zero knowledge proofs of identity. Journal of Cryptology **1** (1988) 77–94.
22. Fiat, A., Shamir, A.: How to prove yourself: Practical solutions to identification and signature problems. In Advances in Cryptology, Proc. of Crypto '86 (Lecture Notes in Computer Science 263) (1987) A. Odlyzko, Ed. Springer-Verlag pp. 186–194.
23. Goldreich, O., Micali, S., Wigderson, A.: How to play any mental game. In Proceedings of the Nineteenth annual ACM Symp. Theory of Computing, STOC (May 25–27, 1987) pp. 218–229.
24. Goldwasser, S., Micali, S., Rackoff, C.: The knowledge complexity of interactive proof systems. Siam J. Comput. **18** (1989) 186–208.
25. Jones, T. C., Seberry, J.: Authentication without secrecy. ARS Combinatoria **21** (1986) 115–121.
26. Kemmerer, R.: Shared resource matrix methodology: A practical approach to identifying covert channels. ACM Trans. Comput. Syst. **1** (1983) 256–277.
27. Kemmerer, R., McHugh, J., Haigh, J., Young, W.: Experience using two covert channel analysis techniques of a real system design. In Proc. 1986 Symp.Security and Privacy, IEEE (1986) pp. 14–24.
28. Kümmerle, K.: High bandwidth communication systems: Where do we go? May 1992. Guest Speaker at IEEE Infocom '92, Florence, Italy.
29. Lampson, B. W.: A note on the confinement problem. Comm. ACM **16** (1973) 613–615.
30. Lenstra, A. K., Manasse, M. S.: Factoring by electronic mail. In Advances in Cryptology, Proc. of Eurocrypt '89 (Lecture Notes in Computer Science 434) (1990) J.-J. Quisquater and J. Vandewalle, Eds. Springer-Verlag pp. 355–371.
31. Micali, S.: Fair public-key cryptosystems. Presented at Crypto '92, Santa Barbara, California, U.S.A., to appear in Advances in Cryptology — Crypto '92, Proceedings (Lecture Notes in Computer Science), Springer-Verlag Augustus 16–20, 1992.
32. Morain, F.: Distributed primality proving and the primality of $(2^{3539}+1)/3$. In Advances in Cryptology, Proc. of Eurocrypt '90 (Lecture Notes in Computer Science 473) (1991) I. Damgård, Ed. Springer-Verlag pp. 110–123.
33. Newberry, M.: Active intruder Detection: Some Aspects of Computer Security and User Authentification. PhD thesis University of New South Wales, ADFA, Department of Computer Science Canberra, Australia 1991.
34. Ostrovsky, R., Yung, M.: How to withstand mobile virus attacks. In Proceedings of the 10-th Annual ACM Symp. on Principles of Distributed Computing (August 19–21, 1991) pp. 51–60.

35. Poras, P. A., Kemmerer, R. A.: Covert flow trees: a technique for identifying and analyzing covert storage channels. In Proc. of the 1991 IEEE Symposium on Security and Privacy (May 1991) IEEE Computer Society Press pp. 36–51.
36. Purdy, G. B.: A high security log-in procedure. Commun. ACM **17** (1974) 442–445.
37. Quisquater, J.-J.: Signatures, identifications et controles d'accès. Lecture given at INRIA (France) December 16, 1986.
38. Quisquater, J.-J., Delescaille, J.-P.: How easy is collision search? Application to DES. In Advances in Cryptology, Proc. of Eurocrypt '89 (Lecture Notes in Computer Science 434) (1990) J.-J. Quisquater and J. Vandewalle, Eds. Springer-Verlag pp. 429–434.
39. Quisquater, J.-J., Desmedt, Y. G.: Chinese lotto as an exhaustive code-breaking machine. Computer **24** (1991) 14–22.
40. Shamir, A.: How to share a secret. Commun. ACM **22** (1979) 612–613.
41. Simmons, G. J.: Verification of treaty compliance-revisited. In Proc. of the 1983 IEEE Symposium on Security and Privacy (April 25–27, 1983) IEEE Computer Society Press pp. 61–66.
42. Simmons, G. J.: The prisoners' problem and the subliminal channel. In Advances in Cryptology. Proc. of Crypto 83 (1984) D. Chaum, Ed. Plenum Press N.Y. pp. 51–67.
43. Simmons, G. J.: The subliminal channel and digital signatures. In Advances in Cryptology. Proc. of Eurocrypt 84 (Lecture Notes in Computer Science 209) (1985) T. Beth, N. Cot, and I. Ingemarsson, Eds. Springer-Verlag, Berlin pp. 364–378.
44. Simmons, G. J.: The secure subliminal channel (?). In Advances in Cryptology: Crypto '85, Proceedings (Lecture Notes in Computer Science 218) (1986) H. C. Williams, Ed. Springer-Verlag pp. 33–41.
45. Simmons, G. J.: How to insure that data acquired to verify treaty compliance are trustworthy. Proc. IEEE **76** (1988) 621–627.
46. Simmons, G. J.: Contemporary cryptology: A foreword. In Contemporary Cryptology, G. J. Simmons, Ed. IEEE Press 1992 pp. vii–xv.
47. Simmons, G. J.: An introduction to shared secret and/or shared control schemes and their application. In Contemporary Cryptology, G. J. Simmons, Ed. IEEE Press 1992 pp. 441–497.
48. T7001 random number generator. AT&T, Data Sheet May 1985.
49. U.S. Department of Defense. Department of Defense Trusted Computer System Evaluation Criteria August 15, 1983. Also known as the Orange Book.
50. White, S. R.: Covert distributed processing with computer viruses. In Advances in Cryptology — Crypto '89, Proceedings (Lecture Notes in Computer Science 435) (1990) G. Brassard, Ed. Springer-Verlag pp. 616–619.
51. Wilkes, M. V.: Time-Sharing computer systems. American Elsevier New York 1975.