must be regarded as a real possibility and therefore there may be an advantage in a system which gives better protection to the PIN.

6.2 DESIGN PRINCIPLES OF THE TOKEN

One way in which the PIN can be given better protection is to provide a keyboard for its entry which is under direct user control. Thus the PIN is not entered on the keyboard of the retail terminal, but upon a keyboard specially mounted on the token itself. The first design principle of the NPL intelligent token is therefore that the PIN should be presented to the system on a keyboard integral with the token. We shall later describe how the token is able to check the PIN validity and then satisfy the terminal that the PIN was valid without actually disclosing the PIN to the terminal. Clearly it is not sufficient for the token to receive the PIN on its keyboard, check it and then send a message saying 'the PIN was correct' to the terminal, since this message could be sent by a false token with an incorrect PIN. The message must be such that the terminal can rely upon it.

In a transaction processing system the user is usually dependent on displayed information on the terminal; the question is whether the user can always trust this information. It is a common experience to motorists to buy petrol from a pump with digital light emitting diode display; elements of these displays frequently fail and an amount (petrol or value) is displayed which is different from the correct value. It is not inconceivable that a retail terminal display might be deliberately altered by someone seeking to defraud the system owners or users. Again this brings us back to the customer's personal token. If it were possible to display the transaction details, particularly the amount of money to be committed, on a display under customer control, then this problem would be much reduced, at least from the customer's point of view; comparison of token and terminal displays may give added confidence. For this reason the NPL intelligent token carries its own display for communication with the customer, which is the second important design principle of the token.

We have indicated earlier the importance that must attach to the integrity of transaction messages which control the movement of funds between user and retailer accounts. It is frequent practice to protect transaction messages by encipherment techniques, often by authentication based on symmetric encipherment algorithms. An alternative and attractive method of ensuring integrity is based on the digital signature derived from an application of public key cryptography. As we shall see, the NPL token is able to satisfy a terminal that a correct PIN has been offered without disclosure of that PIN; this property depends upon the application

of a digital signature. The basic token design therefore includes the ability to calculate digital signatures using a stored secret key. Since the ability to calculate signatures is a fundamental requirement in the device, it is convenient to apply this ability to calculating signatures on transaction messages authorised by the token holder. Transaction messages signed in this way can be checked anywhere in the transaction processing system where a reliable copy of the corresponding public key is available.

6.3 REALISATION OF THE TOKEN DESIGN PRINCIPLES

We have now identified the three fundamental design principles of the NPL intelligent token – integral keyboard, integral display and ability to calculate digital signatures. We proceed to discuss the ways in which these design principles have come to be implemented in physical and logical terms. The central part of the design is an implementation of the RSA public key cryptosystem [4]; this software implementation runs on a fast signal processing chip, the Texas Instruments TMS32010.

Personal identity verification (more strictly PIN verification) begins with presentation of the token to a terminal by the user; the terminal senses the presence of the token and generates a random number which it sends as a challenge to the token; at the same time the token signals to the user (using its own display) that the PIN must be input on the token keyboard. The token is designed to check the PIN and, if the PIN is correct, to sign the random number just received from the terminal using, for this purpose, the secret RSA key contained within the token. (Should the PIN be incorrect, the signature process does not take place and the user is given a limited number of attempts to get the PIN correct, failing which the token is disabled.) Having produced a transformation of the random number by the signature process, the token returns the transformed number to the terminal. The terminal, after having generated the initial random number challenge and while the token is preparing its signed reply, will have sought the public key corresponding to the token; this can come either from a reference source of public keys or can be supplied by the token itself in the first exchange of data with the terminal, in which case the version of the public key is supplied already signed by the secret key of a superior authority (the public key of the authority must then be known to the terminal). Given the public key corresponding to the token, the terminal can check the validity of the returned signature of the random number challenge; correct signature implies correct PIN presentation on the token keyboard. Figure 6.1 illustrates the sequence of events in identity verification.

Because the token is capable of generating RSA signatures, it is a simple extension of its functionality to permit the signature of transaction messages. In a retail point-of-sale system these messages would be pre-

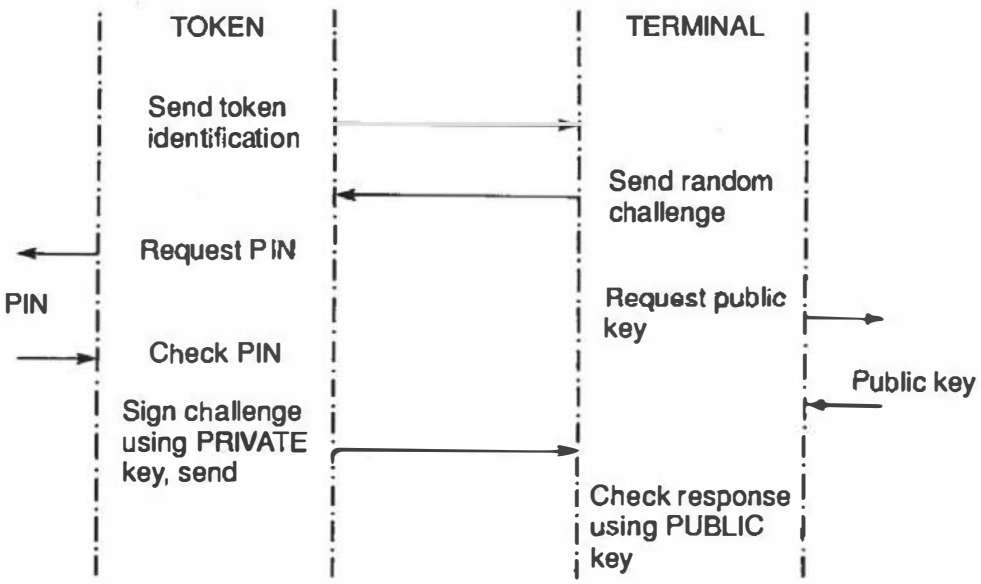

Fig. 6.1 PIN checking, token challenge and response.

pared on the retailer terminal and sent to the token for approval by the token holder (inspection in the token window by the user) and, if approved, signed by the token and returned to the terminal. The terminal can check the validity of the signature and then send the signed message to a transaction processing centre. The signature validity can be checked by any entity in the system having access to the public key corresponding to the signature token. To avoid replays of transactions, it is necessary to include a time and date field in the message. Transaction numbering does not lend itself conveniently to prevention of replay for token originated transactions; tokens accessing multiple services would require a serial number for each and hosts offering services would need a number for each token in valid issue. Figure 6.2 illustrates the sequence of events in transaction signature.

It is an interesting extension of the design that the initial random number challenge may be omitted and replaced by the transaction message. In this case the protocol is shortened by arranging that the identity verification is checked by the signature on the transaction message.

The ability of the token to sign messages can be extended to cover messages in general; the application of the token is not restricted to value transactions.

6.4 THE PROTOTYPE TOKEN

The NPL intelligent token was created in prototype form in a unit measuring 36 cm × 15 cm × 2 cm; this device contained 21 discrete

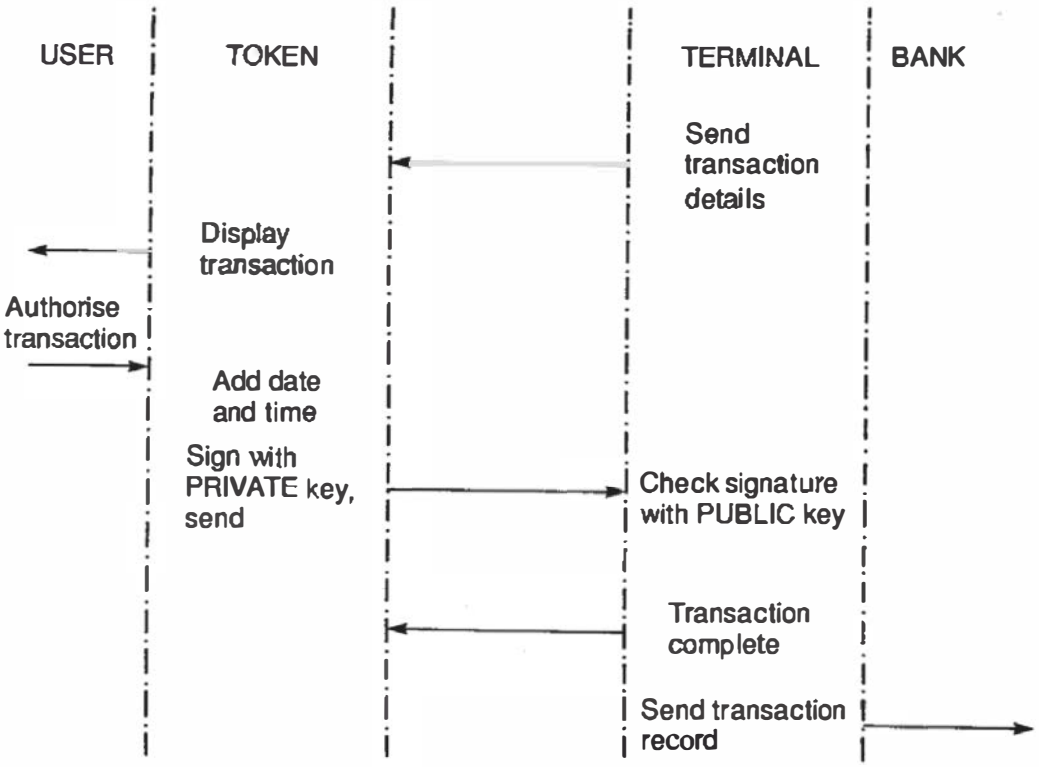

Fig. 6.2 Transaction authorisation and signature.

integrated circuits, including the TMS32010 and an Intel 8085 to act as controller. Battery maintained RAM was provided for storage of parameters such as keys and a record of transactions.

Consideration was given to a semi-custom designed RSA processor chip in the early days of the project. It was considered at that time that the technology was not readily available for such a device and so an alternative method of implementing the algorithm was sought. Texas Instruments had just released the first in a series of fast processor chips designed mainly for signal processing applications. This device, the TMS32010, is a 16-bit microprocessor with an instruction execution time of 200 ns. The instruction set includes a signed 16-bit multiply executed in one cycle. Although not ideal (overflow and the sign bit caused problems) this processor was programmed to perform the RSA calculations. The new design continues to use this processor.

The TMS32010 has limited program and data memory spaces of 4K words and 144 words respectively. Further, there is no high-level language compiler and the speed of the processor makes it difficult to interface to slow peripheral chips. For these reasons, all the remaining functions of the token were placed under the control of a second processor. This split has several advantages; the two processors can work in parallel, thus

reducing or even hiding the RSA calculation time, a high-level language can be used for the application software, none of the TMS32010 memory resources are wasted, and peripheral interfacing is easier and uses fewer components.

Creation of the prototype enabled the development team to demonstrate the correct functioning of the device in applications such as access control, point of sale transactions and signature of alphanumeric messages. Since the prototype was comparatively large, the next stage was to engineer a smaller version. In order to reduce the size, the functions of a number of the separate integrated circuits were absorbed into one application specific integrated circuit (ASIC). The chip count was thereby reduced to 11; further space was saved by using surface mounting technology. The result, produced in collaboration with Texas Instruments, was a device similar in size to a medium sized pocket calculator (about 14 cm $\times$ 9 cm $\times$ 1 cm).

In the new version, the Intel 8085 is replaced by a member of the TMS7000 series of 8-bit microprocessors. As before, this processor also controls the RSA processor and all peripherals, maintains the secret data and runs the application program. 32 K bytes of program memory are available, most applications to date have used only half of this amount. 8K bytes of battery backed RAM are built in for the storage of keys and other data needed for applications.

All of the decoding logic, address latching and bus de-multiplexing for both processors have been reduced to a single semicustom chip designed at the NPL and fabricated in 1.8 micron CMOS gate array technology by Texas Instruments. Figure 6.3 is a block diagram showing the important physical features of the token. The display is a 16 character by 2 line liquid crystal display and the keypad consists of 4 rows of 3 buttons. The reasons for including these on the token are given elsewhere in this chapter. The clock maintains information about the date and time of day, this is used in some applications to date stamp messages. Communication between the token and the outside world is by way of a three-wire serial interface.

Communication between the two processors takes place over an 8-bit bidirectional bus buffer constructed in the semicustom chip. A data block containing the message, exponent and modulus required for an RSA calculation is sent to the RSA processor via this interface, the result is returned in a similar way. Block transfers are completed in a few microseconds, a time not considered significant when set against the calculation time.

The token contains secret information unique only to itself. No other token contains the same secret so the compromise of one does not put the security of the whole system at risk. Nevertheless, measures must be taken to detect tampering and destroy the secret information upon detection. Possession of the secret key would enable an intruder to falsify

DOCKET ALARM

Explore Litigation Insights

Docket Alarm provides insights to develop a more informed litigation strategy and the peace of mind of knowing you're on top of things.

Real-Time Litigation Alerts

Keep your litigation team up-to-date with **real-time alerts** and advanced team management tools built for the enterprise, all while greatly reducing PACER spend.

Our comprehensive service means we can handle Federal, State, and Administrative courts across the country.

Advanced Docket Research

With over 230 million records, Docket Alarm's cloud-native docket research platform finds what other services can't. Coverage includes Federal, State, plus PTAB, TTAB, ITC and NLRB decisions, all in one place.

Identify arguments that have been successful in the past with full text, pinpoint searching. Link to case law cited within any court document via Fastcase.

Analytics At Your Fingertips

Learn what happened the last time a particular judge, opposing counsel or company faced cases similar to yours.

Advanced out-of-the-box PTAB and TTAB analytics are always at your fingertips.

API

Docket Alarm offers a powerful API (application programming interface) to developers that want to integrate case filings into their apps.

LAW FIRMS
Build custom dashboards for your attorneys and clients with live data direct from the court.

Automate many repetitive legal tasks like conflict checks, document management, and marketing.

FINANCIAL INSTITUTIONS
Litigation and bankruptcy checks for companies and debtors.

E-DISCOVERY AND LEGAL VENDORS
Sync your system to PACER to automate legal marketing.