Implementing Cryptography on TFT Technology for Secure Display Applications

Petros Oikonomakos[1], Jacques Fournier[1,2], and Simon Moore[1]

[1] University of Cambridge, Computer Laboratory, William Gates Building,
15 JJ Thomson Avenue, Cambridge CB3 0FD, UK
po230@cl.cam.ac.uk
[2] GEMPLUS, La Vigie, Avenue des Jujubiers, ZI Athélia IV,
13705 La Ciotat Cedex, France

Abstract. Several recent studies have underlined the need for trusted information displays in current and future personal devices. On the other hand, the display market is more and more dominated by low-cost flat-panel structures, driven by Thin-Film Transistor (TFT) circuits. Further, the quality of TFT-based electronics is constantly improving, allowing the fabrication of complicated electronic circuits on TFT technology. We have embarked on a project to implement cryptographic algorithms on polysilicon TFT technology. Our prototype designs will pave the way for secure display realisations combining cryptographic circuits and conventional pixel drivers on the same substrate. An experimental Data Encryption Standard (DES) coprocessor on polysilicon TFT technology is under development, while we are investigating a vector processor architecture to implement Elliptic Curve Cryptography (ECC).

1 Introduction

Investigations related to secure and convenient, new or improved financial transaction models are frequently published nowadays. Some of them [1–3] have identified the improvements in customer security that *trusted* displays have to offer. In this context, a display is trusted (or *secure*) if the content source can be sure that the distributed information will only be presented on the *intended* display. Alternatively, a secure display may be regarded as a means to verify that data is coming from a trusted source. When used in a customer's personal electronic device (PDA, mobile phone, "smart device" etc.), such a display would form part of a secure communication path between a user and a business. An obvious way to develop secure displays is to equip them with decryption electronics, and have the source send encrypted information to them. An unauthorised party not having the adequate key(s) would thus not be able to extract clear display data or display any unauthorised content.

On the display technology front, Organic Light Emitting Diodes (OLEDs) are emerging as a potential market substitute for Liquid Crystal Display (LCD) technology [4]. In the preferred active matrix configuration, both OLED and LCD pixel arrays are driven by Thin-Film Transistors (TFTs), fabricated on an insulating substrate (typically glass). The TFT active area is formed either traditionally by hydrogenated amorphous Silicon (a–Si:H) [5], more recently by polycrystalline Silicon (polysilicon, poly–Si) [4], or by continuous grain Silicon (CG–Si), described by Sharp as a next-generation variant of poly–Si [6]. The last two technologies demonstrate higher carrier mobility than a–Si:H, thus producing better quality transistors. It has therefore been possible to fabricate relatively complicated electronic circuits using both poly–Si and CG–Si [7, 8]. Note that the production and material costs of TFT technology are much lower than that of conventional CMOS circuits. This can be understood even from the fact that the former use very cheap materials for the substrate (glass or plastic), while the latter require Silicon. Hence TFTs are economically preferable in large area electronics applications with relatively low performance requirements.

A straightforward way to cryptographically secure an OLED display would be to use a conventional CMOS cryptographic chip for the decryption of the image information sent by the source. The decrypted information could then be suitably directed to the pixel driver array. The non-secure channel between the cryptographic chip and the driver array constitutes the weakest link in most of today's security systems. However, given the recently demonstrated improved capabilities of modern TFT technologies (mentioned in subsection 3.1 of this paper), it would be interesting to investigate whether cryptographic applications can be successfully implemented in such technologies. The motivation behind such an investigation is that consumer portable electronic devices usually occupy relatively large areas. One could therefore use as much of the area as needed for the actual display, while the rest can be occupied by TFT circuits controlling access to the display, by performing cryptographic operations. Figure 1 depicts an over-simplified configuration of a conceptual consumer smart device adhering to the above ideas. The bottom layer of the device in the figure is occupied by TFT electronics, partly driving the pixels of an OLED display, and partly performing cryptographic functions. Of course, several other components (not shown in Fig. 1) would be needed in a consumer smart device, such as a keypad, I/O functionality, a radio antenna etc. The key idea illustrated by the figure, though, is the migration of cryptographic functionality from CMOS to TFT technology, allowing for better area use, promising lower production cost, and completing the end-to-end security chain. The figure also shows three examples of parties that, depending on the application scenario, may communicate with the smart device and would therefore require use of the display; a few such applications will be explained in Section 2.

The rest of this paper is organised as follows. Section 2 establishes the need for secure displays by reviewing a few relevant works. Section 3 provides a brief up-to-date presentation of display technology and TFT drivers, as well as an overview of recent developments in TFT circuits not directly related to displays. In Section 4 we propose our idea for a cryptographic device on poly–Si TFT technology. We report our progress in the direction of a first DES coprocessor prototype, together with our investigations towards a vector processor architecture for Elliptic Curve Cryptography. Section 5 deals with low-level design

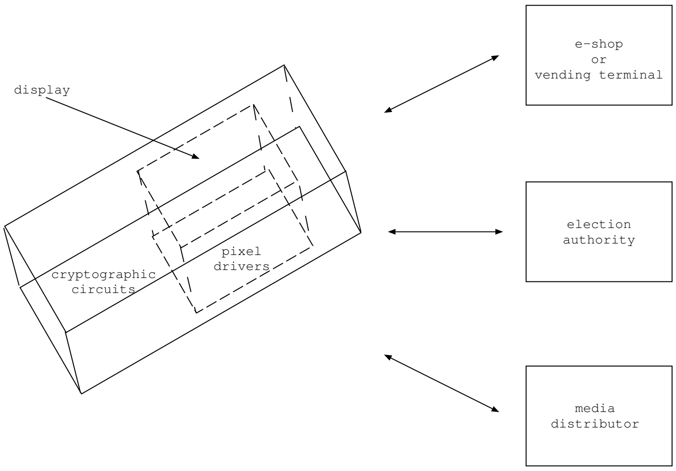

Fig. 1. A smart consumer device with a secure TFT display

considerations, particularly by presenting a Programmable Logic Array (PLA) configuration and detailing its operation. Finally, section 6 concludes the paper.

2 The Need for Secure Displays

In 1995, Yee and Tygar proposed the use of secure coprocessors in electronic commerce [1]. When used in a point-of-sale terminal scenario, the coprocessor (e.g. a smart-card) communicates with the terminal, the customer reviews the transaction on the terminal display and authorises it using the terminal interface. However, there can be no guarantee about the integrity of the terminal display. It is possible that the customer may be reviewing a transaction of a certain amount, and yet the terminal may be charging the card a different amount. This can be either due to a violation of the terminal security by criminal activity, or even due to merchant fraud. As the authors of [1] mention, the solution to that would be a private visual communication path between the smartcard processor and the end user. An information display on the user side is therefore needed, that would only present data received from the card. Such a trusted display would ensure that the user indeed authorises the same transaction that his or her own card is about to implement.

In a recent patent application [3], a cashless payment method is advocated, using a remote customer terminal (mobile phone, PDA or related apparatus encompassing user interfaces) to communicate with a trader station (e.g. supermarket till) and a central station (e.g. bank). The desired amount of money is first read into the trader station, in a conventional way (keyed-in or scanned), and then transmitted to the central station through a data line. The customer reviews the transaction on the trader station display. If the amount is correct, he or she uses the mobile terminal to wirelessly send a unique identifier both to the trader station and to the central station. In response to that, the central station again wirelessly transmits the transaction details to the customer, to be confirmed on the mobile terminal display. Payment proceeds following user authorisation from the mobile terminal to the central station. It can be argued that this model is more secure than that of the previous paragraph, since the user effectively authorises the transaction twice, reviewing payment details submitted from two different sources. Again, however, it is important that the display on the mobile terminal can be trusted to only present information received from the central station. Otherwise the whole model would be vulnerable to “man-in-the-middle” attacks [2], should an attacker interfere between the central station and the customer. Therefore, the mobile terminal will need to be a “smart device” with a secure display.

In 1998 a group of European companies formed the FINancial Transactional IC Card READer (FINREAD) Consortium [9]. Their objective ever since has been to reinforce the level of smartcard security through the specification of a smartcard reader connectable to personal computers, to facilitate home-based e-commerce. Interestingly, the first FINREAD technical specifications mandate, among others, a secure display for the reader. Thus, the consortium of business experts recognise the importance of the integrity of data presented to the user. In a recently accepted paper, Hiltgen et al. [2] describe how such a reader could be used for secure internet banking. In effect, all communication between the card and the bank takes place through the reader and its secure interfaces. The PC plays no active role in manipulating (displaying, encrypting) the card’s details. Further, simple knowledge of the card number and details are no longer enough for a malevolent party to access customer accounts, since the bank server only authenticates users by exchanging encrypted card information with the card through the reader.

In another application area, Hortmann published a short tutorial on long distance e-voting [10]. He identifies the problem of potential communication spoofing between a voter’s PC monitor and the election authority by online attackers, which is very similar in nature to equivalent scenarios in e-banking. Furthermore, in e-voting it is important not only that the voter sees the correct information on his or her display, but also that nobody else can see the information (for reasons of vote anonymity). Once again, the use of secure displays for end-user voting components is advocated. The author of [10] envisions future PDAs armed with trusted displays functioning as “Personal Security Devices” (PSDs) to realise secure e-voting.

Since 2003, the Open Mobile Alliance (OMA), an industry forum dealing with mobile services, has been working on Digital Rights Management (DRM) schemes to securely distribute and protect data on mobile terminals [11]. Their DRM specification details functional models for the distribution of purchased media applications to mobile consumer devices. In order for such a business model to operate profitably, it is imperative that the distributor be certain that the application can only be enjoyed by the customer, and not widely distributed further. If the application contains images, then encrypting them such that only the buyer's secure display can show them will provide a good solution to this problem.

Through the examples reviewed in the above, this section has demonstrated how secure displays on smart devices can enable trusted communication of private, sensitive data through public networks, in a variety of applications, including e-commerce, e-voting, and wireless distribution of media applications.

3 Display Technology and TFTs

Liquid Crystal (LC) based components currently dominate the flat-panel display market. LCDs operate by modulating light generated by a back-light source. In recent years, an alternative emissive technology has been rapidly developing: Organic Light Emitting Diode (OLED) displays. Compared to LCDs, OLEDs demonstrate higher luminous efficiency, brightness, lower production costs and lower operating voltage requirements, in addition to a larger viewing angle. An OLED is a multi-layered electronic structure. One layer is fabricated from an electron transporting material; another from a hole transporting material. In between there exists another layer where the carriers recombine and the excess energy is released as light. The whole structure is often sandwiched between a hole injecting electrode and an electron injecting electrode. Current passing through the OLED causes the emission of light [4].

The pixels of an LC or OLED display can be driven by either a passive or an active matrix (PM or AM respectively), formed by a horizontal address line and a vertical data line. In PM driving, the LCD elements or OLEDs are directly connected to the lines, while AM displays employ actual driving circuits. LCD pixels are voltage-driven; therefore PM driving is a valid low-cost option. In contrast, OLEDs are current-driven. Further, all pixels require a uniform current flow, in order for the OLEDs to provide uniform brightness. It is very difficult to achieve uniform current unless some transistor-based driver circuit is used. It is therefore in practice mandatory to apply active matrix driving of OLED displays. Given the physical dimensions of displays, in most cases it would be economically unwise to use CMOS driving circuits for the active matrix. This is the application area where TFTs on insulating and cheap substrates are useful.

From the above description, it is evident that good quality current sources to be used as AM pixel drivers are the most obviously needed TFT circuits. It is desirable that the drivers not only provide constant current initially, but also continue to do so throughout the expected display lifetime, regardless of any TFT threshold voltage shift over time. A number of designs have been proposed in the literature for this purpose. References [5, 12] deal with a–Si:H TFTs. These TFTs do not demonstrate very good electronic properties. They suffer in particular from low carrier mobility (lower than $1\text{cm}^2/\text{V-s}$), thus requiring very wide channels to allow sufficient current flow (e.g. Nathan et al. [5] report TFTs with channels as wide as $1000\mu\text{m}$). Further, p-channel TFTs are not available in a–Si:H technology [4]. Nevertheless a–Si:H TFT technology is mature and still draws significant research attention. In this context, reference [12] proposes a constant current source composed of 4 TFTs and a storage capacitor. Reference [5] shows an improved version, requiring no storage capacitor.

The driver designs mentioned in the previous paragraph can equally well be used in poly–Si TFT based displays. Poly–Si TFTs demonstrate much greater mobility values than their a–Si:H counterparts (typically by more than an order of magnitude). This allows for much narrower transistors (W/L=2 is achievable). Many poly–Si TFT processes are also able to produce p-channel devices. In [4], Stewart et al. describe a number of refinements to conventional poly–Si TFT fabrication processes that were shown to lead to more uniform TFT characteristics. This way, brightness uniformity can be improved even for driver circuits consisting of only 1 or 2 TFTs.

It is noteworthy that an interesting family of low-temperature poly–Si TFT processes has recently been developed (termed LTPS–TFT) [6]. These processes enable the relatively easy fabrication of TFT circuits on non-conventional substrates, e.g. plastic or various flexible substrates.

3.1 Recent TFT Applications

TFT electronics unrelated to information displays are not widespread. However, the availability of p-channel devices, the continuous improvement in electronic properties and the reduced fabrication costs, in addition to unique characteristics such as manufacturability on flexible substrates have recently triggered a certain degree of research activity on other potential uses of poly–Si TFTs, LTPS–TFTs and CG–Si TFTs. Some characteristic examples are presented in this section.

Hashido et al. [13] developed a capacitive fingerprint sensor using LTPS–TFT technology. They initially observe that conventional optical fingerprint authentication systems are very expensive and not portable. Direct-contact fingerprint sensors are a portable alternative; implementing such sensors on TFT technology additionally lowers the production cost. Their sensor is based on the assumption that the capacitance between a given area of the human finger and a sensor plate that the finger touches depends on the morphology of the area (i.e. whether it is a “valley” or a “ridge”). A simple 1-TFT sensor cell is configured that, together with a read-out TFT, converts this capacitance to voltage. The overall sensor chip comprises a matrix of such sensor cells, as well as buffers and shift registers that control the continuous scanning of all rows and columns of the matrix. Their experimental results undoubtedly support their sensing method.

Estrela et al. [14] experiment with poly–Si TFTs for biosensor applications. They observe consistent and repeatable threshold voltage shifts in the current-voltage (I-V) characteristics of TFTs when they come in contact with certain biochemical agents. Based on this, they demonstrate the potential usefulness of poly–Si TFTs as inexpensive disposable pH sensors, penicillin sensors, as well as DNA hybridization sensors.

In a more conventional application, Lee et al. [8] present a full Z80 CPU (8-bit) developed using CG–TFT technology on a glass substrate. CG–TFTs typically demonstrate three times the carrier mobility of LTPS–TFTs [6]. The presented chip comprises 13000 TFTs and runs at 3 MHz when powered at 5

Volts. The authors of [8] report it as the first publicly-announced successful step in the direction of realising full-scale electronic systems on glass substrates ("Systems on Panels").

Finally, Karaki et al. [7] announced the fabrication of an 8-bit LTPS asynchronous microprocessor, named ACT11. Operating asynchronously provides robustness against variations in TFT I-V characteristics as well as power savings. The chip nominally operates at 5 Volts.

4 Developing Cryptography on Poly–Si TFTs

The discussion so far established that as TFT technologies mature, they can accommodate more and more complicated digital electronics applications. The integration of substantial functional circuits and display drivers on the same substrate appears to be a matter of time. The state-of-the-art rapidly approaches a stage where high-volume production will demand serious CAD tool support for TFT chip production lines. Motivated by these observations, we have embarked on a research project to implement cryptographic functionality on poly–Si TFT technology. We expect this concept to be particularly useful for the development of secure displays, to be used in financial and other future applications such as these described in Section 2 of this paper. The current capabilities of TFTs cannot cope with clock frequency values above a few MHz (or equivalent asynchronous throughput). However, most of these applications could easily be accommodated by static and slow displays without seriously impairing customer satisfaction. In addition, TFT characteristics are improving rapidly. Therefore it is expected that the commercial relevance of TFT electronics applications will increase continuously in the future. In other words, cryptographic TFT chips may in the future be used even in scenarios requiring fast displays.

To investigate about the implementation of cryptographic functions on poly–Si TFTs, we chose to focus on the simple concept of displaying information encrypted using DES. In order to securely distribute and refresh the DES keys we include ECC capabilities in our scheme as illustrated in Fig. 2. Our chip will work in the following fashion.

- The data received in the input buffer is assumed to be encrypted according to the Data Encryption Standard (DES) [15].
- The 56-bit DES keys are transmitted encrypted using an asymmetric public-key scheme, in our case Elliptic Curve Cryptography. In the field of public-key cryptography, Elliptic Curves (ECs) have performance and key-size advantages over the RSA scheme [16]. We therefore choose them for our design.
- The environment first provides a number of encrypted DES keys to the input buffer.
- The input controller routes these keys to the ECC processor `VeMICry` shown in the figure.
- Processor `VeMICry` is being designed to include special hardware to implement modular arithmetic needed for ECC. Its overall architecture accords

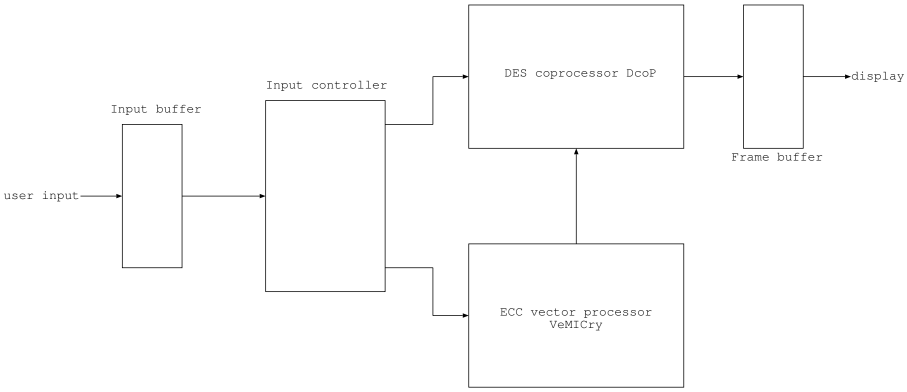

Fig. 2. Cryptographic chip architecture

to the *vector* processor model [17]. More details are provided in subsection 4.2 of this paper.
- While VeMICry is decrypting the keys using its private key, the environment provides the actual DES-encrypted data to the input buffer.
- The input controller then makes sure the data is routed to the DES coprocessor (DcoP in the figure).
- Coprocessor DcoP is a pure hardware module and makes heavy use of PLAs as building blocks. More details on its architecture are given in subsection 4.1 of this paper.
- When VeMICry finishes decrypting the keys, it sends them to the DES coprocessor, to be used for data decryption.
- Coprocessor DcoP then performs DES decryption and writes the decrypted data to the output buffer. In the figure the buffer is termed frame buffer, since the chip is intended to feed display drivers.
- While DcoP performs decryption, the environment provides new keys to the chip input. The keys are again sent to VeMICry and a new cycle of operation begins.

56-bit DES keys are no longer considered to be completely secure [18]. However, one could envisage to refresh the keys frequently enough to discourage any attack on the DES. In a real application, one could use 3-DES or AES; our chip is simply a proof-of-concept of cryptography using TFTs. The security of the overall scheme will also depend on the security of the ECC processor (and its resistance to side-channel attacks) and how the device's private key is stored. If implemented and combined with pixel drivers, the architecture will provide cryptographic protection to the display. With those cryptographic capabilities, we could for example make sure that only "authenticated" users can access the display or that distributed images are only visible on that particular display.

Of the blocks shown in Fig. 2, VeMICry and DcoP are currently under development. The input controller is expected to be nothing more complicated than a state machine, routing a fixed number of input packets to VeMICry, followed by another fixed number of packets to DcoP. The following subsections 4.1 and 4.2 provide architectural details on the design of the two processors.

4.1 The DES Coprocessor

The DES coprocessor is being designed purely as a hardware module, comprising three blocks, namely the key schedule, round block, and the controller. It is a straightforward implementation, shown in the block diagram of Fig. 3. The coprocessor receives a 64-bit encrypted data input, directed to the round block, and a 56-bit key, directed to the key schedule block. The environment (ultimately the input controller of Fig. 2) also raises two flags – I and K – as soon as valid data and a valid key have been fed to DcoP. As soon as I and K are raised, the controller state machine orders the key schedule to compute a subkey, again by raising a suitable flag. The key schedule block computes the subkey and feeds it to the round block, while informing the controller about computation completion. The controller further signals to the round block that the subkey is ready. Upon receiving the signal, the round block responds by using the subkey to produce the partial result, and subsequently informs the controller. The same process is repeated sixteen times for all DES rounds [15]. After all rounds, the decrypted output is available at the round block output. The controller informs the environment and waits for new input and key values. Throughout the process, the controller asserts or deasserts suitable signals to make sure the key schedule performs single or double shifts depending on the current round.

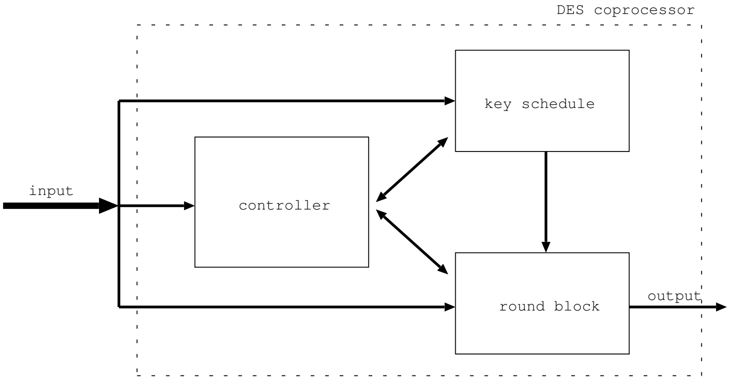

Fig. 3. DES coprocessor block diagram

It is evident that this simple model can easily be adjusted to perform encryption instead of decryption, by performing left or right shifts in the key schedule block. Further, it can also easily be amended to implement triple DES instead of standard DES.

Behavioural Verilog [19] models for the `DcoP` blocks have been developed and confirmed by simulation. The actual layout is currently under development. The resulting chip will be the first, to our knowledge, cryptographic application on poly–Si TFT technology, and at the same time the first poly–Si TFT chip to feature a 64-bit datapath. It will test the feasibility of cryptography on TFTs and build up our confidence towards full integration of the architecture of Fig. 2. Note that this design can be easily tweaked to execute stronger encryption algorithms like DESX.

4.2 A Vector Processor for Elliptic Curve Cryptography

This section provides information about the architecture and functional model of the vector processor with cryptographic support shown in Fig. 2. We have used the acronym VeMICry, for Vectorial MIPS for Cryptography [20]. In essence, VeMICry comprises a simple MIPS-I processor [21] implementing usual, "scalar" instructions, together with a vector coprocessor for the vector instructions. The simplified block diagram of Fig. 4 depicts this idea. The overall processor works very much as a standard MIPS as regards conventional instructions; when *vector* instructions are encountered in the program then the decoder directs them to the vector coprocessor. As the name suggests, vector instructions operate on vectors of registers rather than on individual registers. A total of 17 vector instructions have been defined for VeMICry; a full list is provided in [20]. A few examples – relevant to public key cryptography – are:

- Unsigned Vector Addition: adds the contents of respective elements (registers) of two vectors and writes the result to a third, while propagating carries from the ith element to the i+1st.
- Vector-Scalar Unsigned Addition: adds a scalar value – stored in a single register – to each vector element and writes the result to a target vector.
- Vector-Scalar Arithmetic Multiplication: multiplies a vector by a scalar value while propagating carries. The result is written to a target vector.
- Vector-Scalar Polynomial Multiplication: multiplies a vector by a scalar value without carry propagation. The result is written to a target vector.

Clearly, the last two instructions can be used to implement modular multiplication, based on Montgomery's reduction algorithm [22], in prime or binary Galois Fields. This multiplication is the most critical opearation of EC point multiplication required for EC decryption.

The reason why we chose a vector architecture is that cryptographic algorithms in general and ECC in particular operate on very wide datapaths and long precision numbers. Decomposing the data into vectors of registers of smaller widths and working on vectors and vector elements *in parallel* is expected to increase performance. Further, a vector processor datapath is modular and scalable, thus can easily be deployed in a variety of applications. Finally, a vector processor has a simpler control path and scheduling logic than other superscalar processors, thus reducing power dissipation [17, 23]. Figure 5 shows the "heart"

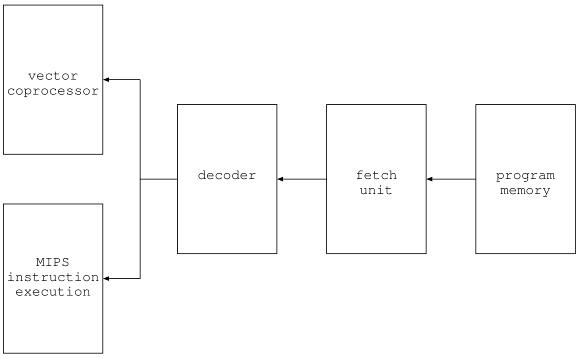

Fig. 4. ECC vector processor block diagram

of the vector coprocessor, that is the vector register file together with vector processing units (VPUs) to implement the instructions. Naturally, the coprocessor also needs peripheral control logic not shown in the figure. In essence, this logic will implement a vector instruction pipeline, separate from and communicating with the scalar MIPS pipeline (Fig. 4). Parameters in the design of the coprocessor include the number of vectors q, the number of elements per vector p, and the number of processing units r (all three shown in Fig. 5), as well as the register bit-width, currently fixed at 32. The choice of these parameters will influence the processor performance, area, and degree of parallelism. In order to explore the trade-offs between these characteristics, we have built a functional model of the VeMICry using the ArchC simulation tool [24]. Details and simulation results can be found in [20], showing significant performance improvements when the Montgomery algorithm runs on the vector processor model, compared to equivalent realisations on a purely scalar, conventional MIPS.

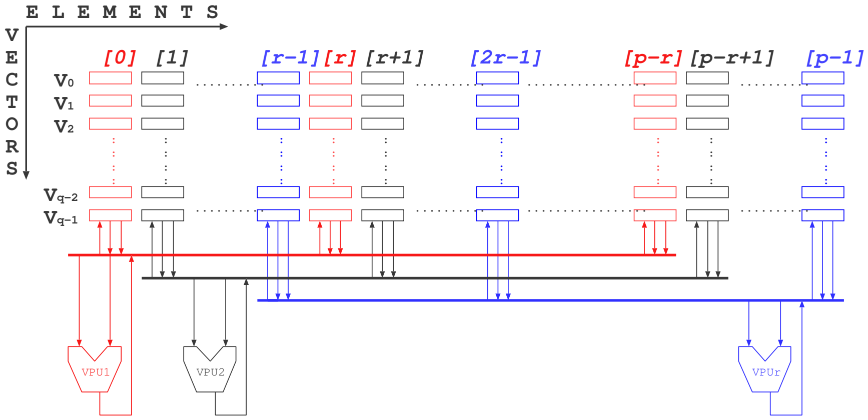

Fig. 5. Vector register file and connections to vector processing units

A Verilog model for the vector processor of Fig. 4 is under development. The architecture will allow us to work on datapaths up to 256 bits wide.

Note that the VeMICry functional model is not restricted to modular multiplication, ECCs or public key cryptography; the AES algorithm [25] has also been simulated on it and again improvements were demonstrated in [20]. While in this particular project it is employed for ECC decryption, it should be regarded as a scalable, high performance processor architecture employable in a variety of cryptographic applications.

5 Low-level Design Considerations

Instead of randomly placed logic gate realisations, in our DES design we are heavily relying on regular structures, in particular PLAs. Due to their geometrically regular layout, PLAs demonstrate timing predictability and controllability. Therefore they are often used in modern CMOS design flows to achieve quick timing closure [26]. In these dynamic-logic structures, power is dissipated only immediately after clock edges [27]. Therefore, the PLA outputs do not experience data-dependent power glitches; this can be regarded as a counter measure against side-channel attacks. In line with the recommendations of [28], we thus provide a degree of security "by design". This may not be very relevant in the case of the architecture in Fig. 2 as in practice an attacker would rather extract the ECC private key than the DES secret keys which are refreshed frequently. However, it is definitely a positive feature for our coprocessor as such, should it be used to implement DES, DESX or triple DES alone.

After reviewing the PLA configurations proposed in the literature and conducting a number of electrical simulations, we decided to use the circuit shown in Fig. 6 as our basic PLA cell. The figure depicts one "AND" and one "OR" plane term, together with an interplane buffer and control logic.

In more detail, the control logic comprises two Muller C-elements and a few inverters constituting a delay line (four inverters are shown for the sake of the illustration – more or less can be used as required). The asynchronous 4-phase single-rail handshaking protocol [29] is thus realised. In the asynchronous operation context, the PLA is treated as combinational hardware handling bundled-data coming from an asynchronous latch. The PLA output is also considered to feed the latch of the next logic stage. The PLA is for the most part an asynchronous counterpart of the synchronous design presented by Wang et al. [27]. Indeed, if a clock was applied as shown in the figure ("clock") instead of the asynchronous control signals, then we would have a perfectly working synchronous PLA. For our project, we choose an asynchronous implementation. This is firstly because it is difficult to route a clock distribution network throughout the chip, given that TFT technologies rarely use more than two metal layers [13]. Further, in line with Karaki et al. [7], we acknowledge the importance of I-V characteristic variation tolerance that asynchronous design offers.

The PLA is implemented using n- and p-channel TFTs (nTFTs and pTFTs respectively) in dynamic logic configuration and as such works in two phases,

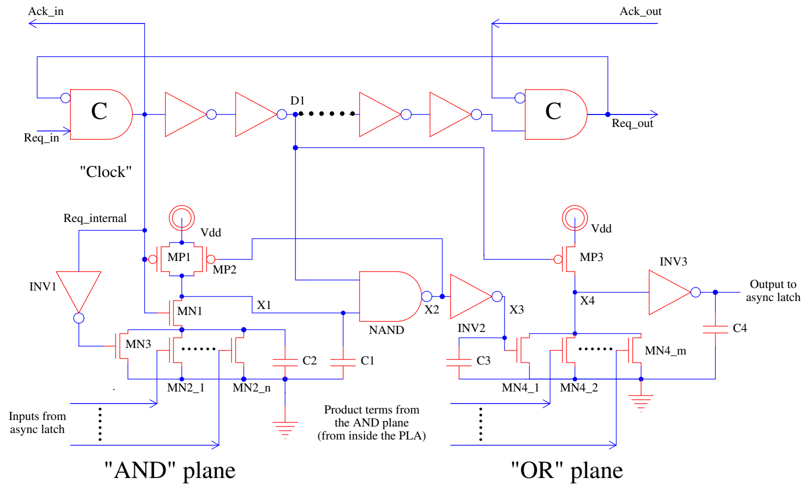

Fig. 6. Basic PLA architecture

namely "precharge" and "evaluate". When Req_internal=0, the circuit is in the precharge phase, and points X1 and X4 are driven to Vdd (the latter after two inverters' delay). In the subsequent evaluate phase (Req_internal=1), the pull-down network of nTFTs in the AND plane determines the logic value at X1. After two inverter delays, this value is allowed to propagate to the OR plane through the interplane buffer composed of the NAND gate and inverter INV2. The OR plane pull-down network then determines the ultimate PLA output. Capacitors C1 – C3 in the figure model parasitics, corresponding to long lines in the actual layout [27], while C4 signifies the output load.

While the PLA operation described above is typical of dynamic logic, the design of Fig. 6 also includes some non-standard elements. First of all, the first inverting element of the interplane buffer is not a pure inverter but a NAND gate. This ensures that the voltage at point X2 is the logic inverse of X1 only in the evaluation phase. During precharge, the voltage at X2 is kept high, therefore point X3 is kept low and the need for a ground switch in the OR plane is eliminated. This mechanism both speeds up the OR plane, and saves power, since it minimizes the switching activity in the interplane buffer. The second non-standard technique is the charge sharing phenomenon exploited in the AND plane. Notice the nTFT MN1. It is effectively the ground switch of the AND plane, but it has been moved between the precharge pTFT and the nTFTs implementing the function. As soon as Req_internal goes high, capacitor C1 transfers some of its charge to C2 through MN1, regardless of the input pattern. If any of the MN2_i nTFTs are on, then the rest of the charge in C1 will be transfered to ground and X1 will be driven low. The charge sharing effect thus speeds up the discharge process and the overall PLA evaluation phase. If all MN2_i TFTs are off, then C1 loses some charge to C2; this charge is replenished when transistor MP2 is turned on, since X2 is driven low. Thus, the design continues to operate correctly. In the subsequent precharge phase, transistor MN3 turns on and discharges C2. We owe both these ideas to [27].

The addition of two inverter delays between the activation of the AND and the OR planes in the structure of Fig. 6 is our own modification to the original structure of [27]. Indeed, in the design of [27] both planes were activated simultaneously by the system clock (equivalent to our Req_internal). Through simulation we found that this created unnecessary and data-dependant glitches on the interplane buffer, consuming power needlessly and potentially creating security hazards.

We have laid out a library of AND- and OR-plane cells and interplane buffers on poly–Si TFT technology using the Electric full-custom VLSI layout tool [30]. We subsequently wrote a relatively simple tool in the Perl language, which uses this library to automatically create full PLA layouts on the paradigm of Fig. 6 when fed by a description of their equations, in the standard PLA format exemplified and explained in the code of Fig. 7. Most logic functions of the DES standard (notably, the S-boxes) will be laid out using this tool. Together with other basic components (latches, multiplexers, barrel shifters, permutation operations – the latter manually designed simply as re-arrangements of wires), they form the building blocks of DcoP, to be connected together manually again using the layout editor of Electric.

```
.i 3       ──────→ no of inputs
.o 2       ──────→ no of outputs
.p 7       ──────→ no of product terms
001 10    | product terms
010 10    | left-hand side:
100 10    | 1: variable contributes to the term
111 10    | 0: complement of variable contributes to the term
11- 01    | -: don't care
1-1 01    | right-hand side:
-11 01    | 1: term contributes to the OR-plane sum term
          | 0: term does not contribute to the OR-plane sum term
.e         ──────→ end
```

Fig. 7. An example of the standard PLA description format

6 Conclusion

Trusted displays are needed in modern and future applications. In the 'Trusted Computing' model, they will enable content providers to identify the equipment on which protected material is displayed. They may also be used to authenticate any party wishing to present visual information on them. In this paper we have advocated *cryptographically secure displays* and presented their on-going implementation using *polysilicon Thin-Film Transistor* technology. To this end, we have proposed a general cryptographic configuration combining public and secret key cryptography. We have outlined the high-level architectures of its constituent elements, namely a hardware DES coprocessor and a vector processor tailored for cryptographic applications. Finally, we have reported on low-level design considerations, namely by describing a PLA structure and associated automatic layout generator, intended to be used for the production of the main building blocks of our chip layouts.

We are actively working towards a first cryptographic test chip featuring a DES coprocessor on TFTs, and expect to have samples available for measurements within 2006.

Acknowledgement

Thanks are due to EPSRC for funding this work under grant code GR/S05496/01. The authors would also like to thank Simon Hollis for his useful input during the planning phase of this work, and Ross Anderson for his comments on the paper.

References

1. Yee, B., Tygar, J.D.: Secure Coprocessors in Electronic Commerce Applications. Proceedings of the 1st USENIX Workshop on Electronic Commerce, July 1995, 155–170
2. Hiltgen, A., Kramp, T., Weigold T.: Secure Internet Banking Authentication. Accepted for publication in IEEE Security & Privacy, available online at `http://www.ubs.com/1/ShowMedia/ubs_ch/authentication?contentId=75819&name=IEEE2.pdf`
3. Offer, G.: Method and Apparatus for Performing a Cashless Payment Transaction. United States Patent Application #20,020,161,708
4. Stewart, M., Howell, R.S., Pires, L., Hatalis, M.K.: Polysilicon TFT Technology for Active Matrix OLED Displays. IEEE Transactions on Electron Devices, Vol. 48, No. 5, May 2001, 845–851
5. Nathan, A. et al.: Amorphous Silicon Thin Film Transistor Circuit Intergration for Organic LED Displays on Glass and Plastic. IEEE Journal of Solid-State Circuits, Vol. 39, No. 9, September 2004, 1477–1486
6. Sharp Microelectronics of the Americas website: `http://www.sharpsma.com/lcd/lcdguide/Technologies/CG-Silicon.php`
7. Karaki, N. et al.: A Flexible 8b Asynchronous Microprocessor based on Low-Temperature Poly-Silicon TFT Technology. Digest of Technical Papers of the 52nd IEEE International Solid-State Circuit Conference (ISSCC) 2005
8. Lee, B. et al.: A CPU on a Glass Substrate Using CG-Silicon TFTs. Digest of Technical Papers of the 50th IEEE International Solid-State Circuit Conference (ISSCC) 2003
9. FINREAD Specification, FINREAD Consortium. `http://www.finread.com`
10. Hortmann, M.: Tutorial on E-Voting. EURESCOM mess@ge, Issue 3, 2001, page 22, available online at `http://www.eurescom.de/~pub/about-eurescom/message03_2001/message03_2001.pdf`
11. Open Mobile Alliance (OMA): DRM Specification V2.0 Candidate Version 2.0 - 26 April 2005, available online at `http://www.openmobilealliance.org/ftp/Public_documents/BAC/DLDRM/`

12. He, Y., Hattori, R., Kanicki, J.: Current-Source a-Si:H Thin-Film Transistor Circuit for Active-Matrix Organic Light-Emitting Displays. IEEE Electron Device Letters, Vol. 21, No. 12, December 2000, 590–592
13. Hashido, R. et al.: A Capacitive Fingerprint Sensor Chip Using Low-Temperature Poly–Si TFTs on Glass Substrate and a Novel and Unique Sensing Method. IEEE Journal of Solid-State Circuits, Vol. 38, No. 2, February 2003, 274–280
14. Estrela, P., Stewart, A.G., Yan, F., Migliorato, P.: Field Effect Detection of Biomolecular Interactions. Electrochimica Acta, Vol. 50, 2005, 4995–5000
15. US Department of Commerce, National Institute of Standards and Technology: Data Encryption Standard (DES). Federal Information Processing Standards Publication 46-3, October 1999
16. Batina, L., Berna Örs, S., Preneel, B., Vandewalle, J.: Hardware Architectures for Public Key Cryptography. Integration, The VLSI Journal, Vol. 34, 2003, 1–64
17. Asanović, K.: Vector Microprocessors. PhD Thesis, University of California Berkeley, 1998
18. Electronic Frontier Foundation: Cracking DES. Secrets of Encryption Research, Wiretap Politics & Chip Design. July 1998
19. Lilja, D.J., Sapatnekar, S.S.: Designing Digital Computer Systems with Verilog. Cambridge University Press, 2004
20. Fournier, J., Moore, S.: A Vectorial Approach to Cryptography Implementation. Proceedings of the 1st International Conference on Digital Rights Management: Technology, Issues, Challenges and Systems, November 2005
21. MIPS Technologies: MIPS Architecture for Programmers Volume II: The MIPS32 Instruction Set. Technical Report MD00086, Revision 0.95, March 2001
22. Montgomery, P.: Modular Multiplication without Trial Division. Mathematics of Computation, Vol. 44, 1985, 519–521.
23. Folegnani, D., González, A.: Energy Effective Issue Logic. Proceedings of the 28th Annual International Symposium on Computer Architecture, June-July 2001, 230–239
24. T.A. team: The ArchC Architecture Description Language – Reference Manual. Technical Report v.1.2, University of Campinas, December 2004
25. US Department of Commerce, National Institute of Standards and Technology: Advanced Encryption Standard (AES). Federal Information Processing Standards Publication 197, November 2001
26. Posluszny, S. et al.: "Timing Closure by Design", A High Frequency Microprocessor Design Methodology. Proceedings of the 37th ACM/IEEE Design Automation Conference, June 2000, 712–717
27. Wang, J.S., Chang, C.R., Yeh, C.: Analysis and Design of High-Speed and Low-Power CMOS PLAs. IEEE Journal of Solid-State Circuits, Vol. 36, No. 8, August 2001, 1250–1262
28. Li, H., Markettos, A.T., Moore, S.: Security Evaluation Against Electromagnetic Analysis at Design Time. Proceedings of the 7th International Workshop on Cryptographic Hardware and Embedded Systems, August-September 2005, 280–292
29. Sparsø, J., Furber, S.: Principles of Asynchronous Circuit Design: A Systems Perspective. Kluwer Academic Publishers, 2001
30. Static Free Software: Using the Electric VLSI Design System, available online at `http://www.staticfreesoft.com/manual/`