Transporting a Secret using Destructively-Read Memory

Bruce Christianson[*] and Alex Shafarenko
University of Hertfordshire, Hatfield, UK

February 2023

Abstract

Alice wants to send Bob a secret such as a one-time pad. Our proposal is to use a specially designed mass-produced memory chip, rather like a flash drive, called a DeRM (Destructive-Read Memory). As with other distribution methods, including tamper-evident containers and QKD, we require a side-channel that provides end-point authentication and message integrity (although not message secrecy). Advantages of the DeRM over other tamper-evident containers include that DeRMs can be clonable, and correct verification that the DeRM has not been accessed in transit is ensured by the process of extracting the secret content.

Alice wants to send Bob a secret in such a way that Bob can be sure the secret has not been read, or changed, by Moriarty while it was in transit. We assume that the secret is similar to a one-time pad [3]: the secret is lengthy, but has no value to Moriarty if Alice and Bob become aware that the secret has been compromised before they make use of it.

One option is to use a trusted courier, Charlie. The secret is placed on a storage device (such as a DVD), which Alice hands to Charlie. Charlie conveys the DVD to Bob, along with the assurance that it is the correct DVD, and that nobody but her has had access to it. On her return, the courier confirms safe delivery with Alice.

The trusted courier may be supplemented, or replaced, by a tamper-evident container. This container has the properties that (i) it shatters whenever the contents are accessed, ie it moves from a transit state into a shattered state from which the transit state cannot be restored, and (ii) it cannot be cloned, so although the contents can be moved to another instance of the container when the first is shattered, the second container cannot be passed off as the first.

Use of a tamper-evident container still requires a secure side channel between Alice and Bob, using which Alice can inform Bob of the unique fingerprint of the real container of the DVD, and Bob can assure Alice that that container has

[*] comqbc@herts.ac.uk

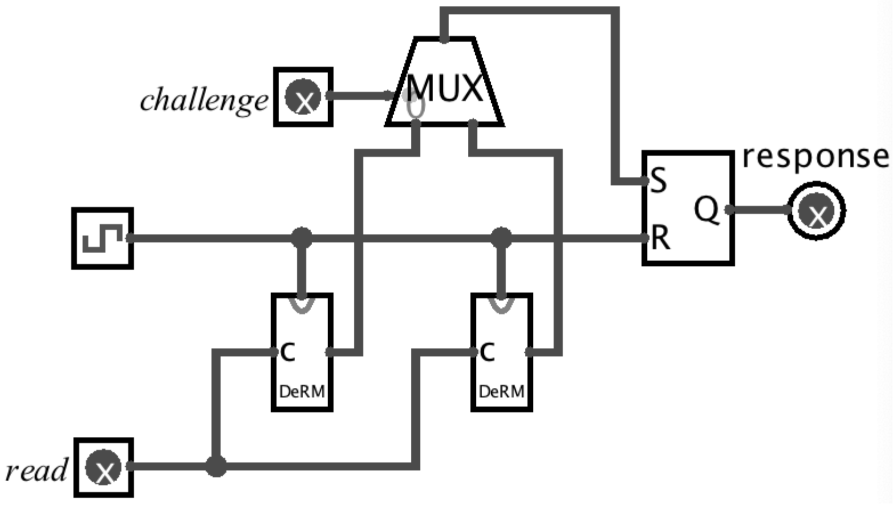
arrived intact. Here secure means that the channel end-points are authenticated, and the integrity of messages between them is guaranteed – but not their secrecy.

Another option is Quantum Key Distribution (QKD). This requires considerable expensive infrastructure to be deployed in advance, typically in the form either of fibre optic cable or satellite uplinks, connecting to Alice and Bob. QKD also requires a side-channel between Alice and Bob with the same security requirements as for the previous tamper-evident physical transport case [4].

QKD relies upon the physical fact that, for each bit, Bob has a choice of possible measurements, only one of which matches the encoding of that bit. The encoding for each bit is chosen at random by Alice. Making the wrong measurement on a bit destroys the value of the bit without revealing it. This prevents Moriarty from measuring bits and then cloning them. The secure side channel is necessary to ensure that Bob knows which measurements match the encodings.

Our proposal is for secret bits to be transported using a specially designed mass-produced memory chip, rather like a flash drive, called a DeRM (Destructive-Read Memory). The DeRM consists of a number of memory elements, each element contains storage for multiple bits. The elements are initialized and read autonomously. The only read operation available for a DeRM memory element is atomic and destructive: reading moves the entire element into a fixed known state, following which all the information previously stored in that element is irrecoverable.

Each memory element contains more information than the destructive read reveals. The read operation requires a challenge value as input. The information output by the read depends on the value of this challenge as well as on the content of the element: however the read destroys all the information in the element. A challenge with the wrong value therefore renders the information corresponding to the correct challenge unobtainable.



DeRM Element Design

There are many possible designs for such hardware, the diagram shows one example.

This type of DeRM memory has two cells per element. Each of the cells is an analogue structure comprising a floating-gate nMOS transistor and two charge pumps. The cell has an input c, and one output. The c input is used to perform a (destructive) read in the current clock cycle. At the rising edge of the clock the charge pumps inside the cell are fully primed, this charging happens in the preceding low-clock period. The output of one pump drives the nMOS control gate, by slowly ramping the voltage up to the level required for hot electron injection into the floating gate.

The transistor will open early if the floating gate is in a discharged state, which we define as state 0 (note that this is opposite to the usual convention in flash memory technology), and will open late if the floating gate is charged up, which we define as state 1. If the transistor opens early, this creates a pulse at the output of the cell; otherwise no pulse is generated. So the result of asserting 1 on a DeRM cell input c is that the cell transitions to state 1 no matter what state it was in. If the state was 0, a pulse will appear on the output; if it was already in state 1, no pulse appears.

The second charge pump ensures that, if the transistor opens at all, then a sufficiently strong current passes through it to ensure the required hot electron injection into the floating gate. The cell design must ensure, for example by using a suitable Zener diode, that no pulse is produced if the charge pumps are not fully charged at the beginning of the clock cycle. This prevents Moriarty from reading a cell, without destroying the content, by starving the cell of power. Once the charge pumps are fully charged, Moriarty can learn nothing by observing power draw during the read.

The SR flip-flop holds the result of the DeRM element destructive read operation, and is reset by the clock at the beginning of the clock cycle. Which of the two cell outputs is connected to the flip-flop depends on the value of the challenge. If there is no pulse from the connected cell, the flip-flop remains reset.

When the DeRM is supplied by the manufacturer, all cells are in the 0 state. A single-use initialization operation with two inputs (not diagrammed) allows Alice to set either, neither, or both of the cells to the 1 state. The subsequent destructive read operation, performed by Bob or Moriarty, causes both cells to end up in the 1 state regardless of their prior content. The DeRM element read operation is physically destructive, as it operates via a floating-gate transistor rather than a digital circuit: the destructive read is analogous to charging a capacitor.

There are thus four element states: 00, 01, 10, and 11. Each destructive read operation takes a one-bit challenge as input, and yields a one-bit output. Depending on whether the challenge is 0 or 1, the output is either the left- or right-hand bit. The other bit is destroyed without being revealed. Alice chooses, in advance, which of the two bits is the secret, and which is a dummy (sacrifice).

A DeRM Protocol

Here is a simple protocol showing how to use the DeRM to transport a secret.

1. Alice writes a secret into the DeRM choosing, at random for each element, which bit is the secret and which is the dummy.

2. Alice sends the DeRM to Bob via some untrusted transport mechanism, such as Royal Mail.

3. Using the secure side channel, Bob confirms to Alice that he has received a DeRM (which may or may not be the correct one).

4. Using the secure side channel, Alice confirms that she did send Bob a DeRM and reveals the correct challenge values.

5. Bob uses these challenges to extract the secret, and sends Alice a verification value on the secure side channel. For the verification value, Bob might encrypt the secret with a publicly known key under CBC mode, and send Alice the final block of the cryptotext. In this case Alice and Bob will destroy the final block of the plaintext secret without using it.

6. Alice confirms to Bob on the secure side channel that the verification value is correct.

At this point, Alice and Bob can both be sure that they share the secret with each other and with no one else. If Moriarty has physically intercepted the DeRM, he won't know the correct challenges, and his attempts to extract the secret will destroy at least some of it.

All DeRMs are the same (they are clonable), but if Moriarty replaces the DeRM Alice sent with a new one, the secret values won't match the correct challenges, and Bob's extracted bitstring won't verify. We can easily ensure that Moriarty must extract the entire contents of the secret – many Gigabytes - in order to obtain any useful bits at all. Simply encrypt the secret with a known key before using it as a one-time pad, in such a way as to ensure (via diffusion and confusion) that every bit of the secret affects every bit of the pad.

One significant advantage of the DeRM over other tamper-evident containers is that Bob cannot shirk the step of correctly checking that the DeRM has not been accessed in transit – the check is implicit in the process of extracting the secret content.

It's possible that Moriarty will spend a lot of money on a lab with an electron microscope and lasers, and examine the memory cells one at a time. Nothing is completely secure against an adversary with unlimited resources. However our primary aim with the DeRM is to provide high leverage for low cost security – we wish to ensure that Moriarty must spend several orders of magnitude more than the cost of the DeRM, which is comparable to the cost of a flash drive, and that Moriarty's attacks have no economy of scale.

Epilogue: Down the Wormhole, and what Alice and Bob find there

What if Moriarty captures the DeRM while it is in transit and replaces it with a different piece of hardware, called a WoRM, of his own infernal construction? If Bob can't tell the difference between a genuine DeRM and a WoRM, then Moriarty can conduct a wormhole attack as follows:

In step 5 Moriarty users Alice's challenges to extract the secret from the DeRM. He transmits the DeRM's responses to the WoRM via a hidden side channel. When Bob interrogates the WoRM, it gives the correct responses. The hidden side channel does not have to involve electromagnetic radiation: for example the WoRM might detect subsonic vibrations.

You might argue that if Alice and Bob can't tell the difference between a genuine DeRM and something manufactured by Moriarty then the game is over before it begins, but the situation is not symmetrical. Perhaps Alice buys her DeRMs in bulk from a trusted supplier, and has them delivered via trusted courier, well in advance of needing to use them, and Bob is a new occasional user who could be fooled into accepting the WoRM.

We can make things harder for Moriarty by altering the protocol so that Bob picks the challenges. In the modified protocol Bob chooses the challenges that he uses in step 5 and sends them to Alice along with the verification value. This requires both Alice and Bob to have access to good sources of secret randomness. Another consequence of this change is that neither of them can choose the shared secret in advance. In what follows, we shall assume that Alice simply loads the DeRM with random bits in step 1.

A wormhole attack now requires the WoRM to be able to transmit to Moriarty as well as to receive. When the WoRM receives the challenge from Bob it sends it to Moriarty, and Moriarty replies with the response from the DeRM. The WoRM then passes this response to Bob.

This attack is easier for Bob to prevent than the previous one, because Moriarty must mount it in real time. If we assume (non-trivially) that the design of the genuine DeRM is sufficiently minimalist that Moriarty cannot cut the DeRM down to fit inside the WoRM without breaking the DeRM, then it suffices for Bob to create an information-theoretic faraday cage [2] around the device that reads the DeRM.

There are various ways of doing this [1, 5]. One standard wormhole countermeasure is distance-bounding. This relies on the fact that the speed of light is finite. For example, if Moriarty cannot place the real DeRM within 15 metres of the WoRM, then there will be a round-trip delay on the read operation of at least 100 ns, which Bob can measure.

What matters isn't so much how long reading the genuine DeRM takes, which could be in the region of 20-40 ns[1], so much as how accurately Bob can measure any delay. As we did for Alice we can assume, if we need to, that

[1] www.smxrtos.com/articles/whiteppr/flashperformance.htm

Bob obtains a specialized DeRM-reader from a trusted supplier, via a trusted courier, well in advance of using it.

In the case of accuracy to within 100 ns, the distance-bounding approach ensures that the genuine DeRM from Alice is either directly attached to the DeRM reader, or is within 15 metres of it. Bob's task is to eliminate the second possibility, or at least to push the cost of it high enough to make alternative, potentially scaleable, attacks more attractive to Moriarty than attempting to intercept each one-time pad on its journey from Alice to Bob : for example bribing a sysadmin, inserting a Trojan horse into Bob's operating system, or subverting a chip manufacturer.

Enforcing a 15 meter cordon around the DeRM reader is more intricate than might appear : the 15 meters includes up (drones) and down (drains); and the real DeRM, attached to one of Moriarty's infernal devices, may be inside a deliberately mis-addressed padded envelope sitting quietly in the outbound mail tray near Bob's workstation. We can certainly decrease the radius with tighter timings, but probably don't need to bother: if Moriaty can place his devices so near Bob, there is plenty of lower hanging fruit for him than DeRM, for example EMF from computers.

References

[1] Bruce Christianson and Alex Shafarenko, 2006, Vintage Bit Cryptography, Security Protocols 14, Springer LNCS 5087, 261-275. doi.org/10.1007/978-3-642-04904-0_34 10.1007/978-3-642-04904-0_35

[2] Bruce Christianson, Alex Shafarenko, Frank Stajano and Ford Long Wong, 2010, Relay-Proof Channels Using UWB Lasers, Security Protocols 18, Springer LNCS 7061, 45-53. doi.org/10.1007/978-3-662-45921-8_8 10.1007/978-3-662-45921-8_9

[3] Frank Miller, 1882, Telegraphic Code to Insure Privacy and Secrecy in the Transmission of Telegrams, Charles M. Cornwell, New York. books.google.com/books?id=jNf2GwAACAAJ

[4] Peter Ryan and Bruce Christianson, 2013, Enhancements to Prepare-and-Measure Based QKD Protocols, Security Protocols 21, Springer LNCS 8263, 123-142. doi.org/10.1007/978-3-642-41717-7_14 10.1007/978-3-642-41717-7_15

[5] Frank Stajano, Ford Long Wong and Bruce Christianson, 2010, Multi-channel Protocols to Prevent Relay Attacks, Financial Cryptography 2010, Springer LNCS, 6052, pp. 4–19. doi.org/10.1007/978-3-642-14577-3_4

Transporting a Secret using Destructively-Read Memory (Transcript of Discussion)

Alex Shafarenko

University of Hertfordshire

Now, Alice and Bob must authenticate the secret, but there's no authentication of the chip. If the chip has been replaced or reprogrammed or tampered with any other way, then because the reads are destructive, Alice and Bob won't have a match of the hashes. So if that happens Bob throws away the chip, acknowledges failure to Alice, and Alice sends a new chip.

First attack, delayering: let's use some chemical solvent and strip that chip bare.

Ross Anderson: Sergei Skorobogatov and colleagues have shown that you can read out the state of chips using a scanning electron microscope. It's fiddly and you have to go through the back, but it is in many cases doable.

Reply: How long does it take?

Ross Anderson: Well they've used this to reverse engineer chips that are used to authenticate medical devices, for example. Not just topology, getting out the contents of the flash memory. Even twenty years ago we were reading the state of flip-flops just by scanning a chip with a laser and observing increased photon counts. That was at micron scale. Nowadays, with the smart card technology in use at about 90 nanometers, you have to use infrared. But believe me, reverse engineering attacks on such chips are possible.

Bruce Christianson: Yes, I've exchanged emails with Sergei about this. One market that we see for this is the dispiriting world of low-cost security, where you're buying one of these chips for ten quid and sticking it in a padded envelope with a frank on the front. If you spent five million pounds on a security system, and Moriarty can spend ten thousand pounds and break it, that's `#epicfail`. If you spent ten pounds on your security and Moriarty can spend ten thousand pounds and break it with an attack that doesn't scale up, because he's very rate limited about how fast he can use the microscope, that's as close to success as you're likely to get.

The objective in this case is to remove the one-time pad distribution from Moriarty's attack surface. What's the purpose of a bicycle lock? In Cambridge the purpose is to get somebody else's bicycle stolen. In the low-end case, we're trying to make sure that Moriarty isn't going to attack the DeRM, because it's cheaper and more convenient for him to bribe a sys admin or, if it's near the end of the tax year and he's got to spend a lot of money to keep his charitable status, he'll subvert a chip manufacturer.

Ross Anderson: Well the competitor in the low-end case will be the big OEMs who use the same chips that are used for EMV, which tend to be repurposed for many different applications. There, you might be able to get a man in China to break the chip for five thousand pounds per instance. For a custom chip like this, you're into design and fabrication costs of six to seven figures[1], against the cost of getting something off a production line that just works: at the very low-end case Alice could send a smart card with the key materials to Bob, and when Bob replies Alice then sends Bob a single message with a PIN number to unlock the chip. That's easily doable now and it works.

Reply: I expected you to stop me in the very beginning and suggest this. Yes, you can have normal memory inside, and then on the perimeter some encipherment, and then you cannot get into the middle of the container from the pins without the PIN. However, the DeRM approach has the advantage that the protection is per bit, every bit is protected and you have to defeat all of them to obtain even one bit of key, provided you have sufficient diffusion. So you need a physical process that's so reliable that it can read all of the bits.

I should also make the point here that we can time-limit the untrusted courier and still not have to trust them. If you limit the delivery time to hours, then I don't think this attack is doable, even with a man in China, because you need to ship it there and do the things.

You can engineer the DeRM chip so the connections are random, which makes it more difficult to know the physical locations of the relevant cells. Also, you can put it in a 3D layer after layer. If you produce physical protection, between the layers, the electron microscope is not a magic bullet. It's been used primarily against chips that are not engineered to withstand that sort of thing. Also, you are aligned with a microelectronics industry that has a vested interest to protect their designs from scanning, and things like that.

So measures have already been developed to protect the design. The same physical measures will protect the challenge, especially if it's time limited to deliver the container to Bob.

Ross Anderson: I think this is a neat idea, re-implementing the basic idea of quantum key distribution but using semiconductor technology rather than quantum optics. If we get some real tamper-resistance experts in like Sergei, the implementation details would probably be different, but I have no doubt that they're doable.

However, I've got a deeper objection to this as a general way of doing stuff. It's the same as my general objections to quantum key distribution. It's this: with quantum key distribution, you can re-key a line encryptor that will do AES encryption between a bank's data centre and its backup data centre. But what is the market for that? If you were part of the AES competition, as I was, then you will probably believe that any AES line encryptor is good for at least 2^{60} blocks without rekeying. If you don't believe that, and you want to re-key it more frequently, there are Kerberos-type protocols that will enable you to do this, and have mathematical proofs of soundness.

[1] Admittedly, this works out at only a pound or so per DeRM chip.

The only way that the quantum optics people can create a market for their quantum key distribution stuff is by denying the existence of this mathematics, so they can create a market for the physics, so they can re-key the line encryptor every few seconds. So it is a solution to a very emphatic non-problem.

Now, the real problems of key management, as people from this protocols workshop have debated for 30 years now, are to do with more complex scenarios. It's not just that Alice wants to speak to Bob, it's that Alice wants Sam to introduce her to Bob with Xavier and Yves and Zara as well. It's less clear that a mechanism for physical key transport, whether it is a content-type or of a tamper-resistant type, gives you any useful leverage there.

Bruce Christianson: This is absolutely right. I agree with all these points, and it's certainly the quantum key distribution market that we are attempting to undercut, not the market for AES. Ross's point is that QKD is, to some extent, a bogus market, but a lot of people are using it.

One of the interesting IoT cases that could not afford to use QKD is where you're not actually concerned with who it really is you're talking to, you just want to be sure that it's the same person that sold you the kit that you've installed. So when I'm getting keys for software updates, or I've got some industrial process and I've gone around sticking microprocessors on all my storage tank managers, I want to know that the stuff that I'm blowing into them really is from the people that I've bought them from in the first place.

Ross Anderson: With QKD, you are starting off from an integrity channel that has the ability to check hashes from one end to another or for [BB84] or whatever, and you're getting confidentiality out of it by appealing to Heisenberg's uncertainty principle or in modern protocols to some notion of entanglement, but you have to start off with end-to-end integrity. Now, exactly the same thing would appear to apply here because what's to stop Moriarty simply setting up a man-in-the-middle attack? So in that sense, what you're doing here doesn't offer anything more than QKD.

Reply: True, true. That was the intention, to replace quantum technologies. But we are much cheaper.

Ross Anderson: But what's actually happened with QKD is that people like Nicolas Gisin with his company ID Quantique in Geneva, make lots of demonstrations about QKD and they claim that they actually sold a pair of line encryptors to a Swiss bank once. But where they actually make their money is by selling quantum optics equipment to lots of physics departments so that undergraduates can replicate the Bell test. I don't think you have got that kind of second string to your bow.

Reply: But their technology comes at a very high price.

Ross Anderson: Well the purpose of that technology was to win a Nobel prize, which happened last year, so John Clauser and Alain Aspect and Anton Zeilinger duly got to shake hands with the King of Sweden. So that's been done.

Reply: As well as being cheaper, we are also less ambitious. [laughter]

Adrian Perrig: One important point here is for achieving extremely long term secrecy: entities have requirements to have secrecy for seventy years, one hundred years. These quantum key distribution techniques were proposed to be able to achieve that. So I think there are some real world use cases here. I'm not a big fan of QKD of course, especially not the huge quantities of money that flow into it, so while I agree with Ross's point there that QKD and AES are separate techniques, I think this DeRM will be a really interesting way to counter the massive volumes of money flowing in to QKD, in a much simpler way, without boiling the ocean.

Ross Anderson: About ten years ago there was a meeting at the Royal Society where the UK Research Councils were announcing that twenty million of our money could be taken away and given to this quantum programme. That has since got legs on. One of the arguments made by one of the quantum evangelists was, think of the security of DNA. Your DNA will be shared with your children and grandchildren, so the cover time required for that is centuries, and only quantum keys are good enough to encrypt your DNA data.

I was sitting there and I could not trust myself to put my hand up and intervene, because I thought that if I did so and started attacking all the various ways in which this is wrong at different levels in the stack, then I would be thrown out of the room for discourtesy. So this would be counter productive. But we can have a talk over lunch if you wish about the privacy of DNA, or the cover time, or whatever. These arguments to my way of thinking are totally and utterly spurious.

Adrian Perrig: But if, with regulators, your data must remain secret for very long time periods, then as a company what else can you do?

Frank Stajano: I would like to ask Adrian, in what way does the mechanism for transferring some secret from base A to base B have anything to do with the duration of the secrecy that it protects? I don't quite understand that.

Adrian Perrig: Right now, the techniques that are being used for key exchange may be breakable by quantum computers, or the algorithms may be broken eventually. By adding QKD, I mean this is their opinion, I'm not necessarily going to buy it, but they say by adding in QKD then even if the other systems are being broken the final encryption still remains secure. That's the argument.

Ross Anderson: I think the argument is if the secret has only ever been in the public domain encyphered by a one-time pad, and you've complied with the protocol conditions on one time pad disposal, then the time to break it is infinite. Whereas, for any encryption that relies on an algorithm, in theory, somebody could come up with a wonderful solution for breaking the algorithm at some point in the next hundred years[2].

Reply: I think that one-time pad disposal is okay for us. There's an easy solution, just destroy the DeRM after use, drop it into acid while it's still inside the

[2] Such algorithmic breaks need not be retrospectively fatal for authentication and integrity, but they will be in the case of confidentiality.

Faraday cage. But we need to defeat possible relay attacks if our claim of confidentiality is to be maintained for years. By the time we start using the secret we need to be sure a wormhole relay attack has not taken place, which we can do by increasing the bandwidth.

And I think that if we increase the bandwidth, it's a hard guarantee, because it's a physical principle. At some point Moriarty just runs out of bandwidth, no matter what he does.