The Man-in-the-Middle Defence

Ross Anderson and Mike Bond

Computer Laboratory, University of Cambridge
{Ross.Anderson, Mike.Bond}@cl.cam.ac.uk

Abstract. Eliminating middlemen from security protocols helps less than one would think. EMV electronic payments, for example, can be made fairer by adding an electronic attorney – a middleman which mediates access to a customer's card. We compare middlemen in crypto protocols and APIs with those in the real world, and show that a man-in-the-middle defence is helpful in many circumstances. We suggest that the middleman has been unfairly demonised.

1 Introduction

The man-in-the-middle is much maligned. The security protocol literature abounds with middleman attacks, and designs for security architectures commonly assume that if we could just cut out any possibility of interception, so that endpoints talk directly and securely, then everything will be OK. This could not be further from the truth. More often than not, the party that cheats you *is* the very one you thought you wanted to talk to in the first place, rather than some large-eared villain in the shadows.

In real life the middleman is often an ally who defends your interests; he is an essential part of going about normal business. We have our estate agents, our lawyers and accountants – even our priests – all acting as middlemen and representing our interests to those who might otherwise harm us. Resentment of the middleman usually only arises when he serves more than one master, or acquires too much independent power.

In this paper we argue that computer security should restore the middleman to his proper status. We describe several protocols where participants would benefit from being shielded from the actions of other participants. In fact there is already a body of literature in computer security covering composition problems, and if we can apply these ideas more broadly, we might learn how to engineer security protocols for multiple middlemen.

2 Electronic Commerce

Since 2005, British bank payment cards use the EMV protocols – a development known to the public as "Chip and PIN". Instead of reading a static number from a magnetic strip, a payment terminal supplies a customer PIN to a smartcard which verifies it and computes a MAC on the transaction, using a key it shares with the issuing bank. On casual inspection, this appears to be an end-to-end protocol; but the customer does not have a trustworthy means of entering his PIN into his card, or of checking the transaction details (payee and amount).

Consider a concrete scenario: You go for lunch at a small London restaurant and pay using your chipcard, unware that the restaurant is corrupt. You ask for the bill, and the waiter brings a handheld terminal to your table. Meanwhile, on the other side of town, his accomplice is loitering in a jeweller's store. The waiter sends an SMS message to his accomplice, who goes up to make a purchase. As you insert your own card into the waiter's terminal, the accomplice inserts a fake card into the jewellers. The waiter's sabotaged reader simply forwards the traffic from your card wirelessly to the card at the jewellers. You enter the PIN, thinking you're paying for lunch, but in fact you're buying the crooks a diamond!

We investigated the EMV specifications to determine whether such a 'middleman attack' was possible and practical. It is, and there seems to be no easy way to extend the EMV protocol to sort it out. It then occurred to us that if the merchant (or a corrupt merchant employee) could insert a relay device to monitor and forward the EMV protocol, maybe the customer could add her own middleman to do the same job, but with her interests in mind. An economic analysis of the problem is that the chipcard defends the bank's interests; the terminal defends the merchant's interests; but no party to the protocol defends the customer. What is the electronic equivalent of taking your lawyer along with you to the shop?

3 The Electronic Attorney

Our solution is to create an electronic attorney – a device that participates in the protocol whether the merchant and bank like it or not, which is paid for solely by the customer, and which acts only in her interests.

In the case of EMV, the protocol requires clear PIN entry by the customer and clear transaction entry by the merchant, supplies both to the chipcard, and if the PIN is correct the chipcard computes a MAC on the transaction data for transmission to the bank. The path from the terminal to the card can thus be mediated by a gadget that gives the customer a trustworthy display of the payee and amount, and can supply or withold the PIN to the card. It can also keep an independent audit trail.

A real-world implementation would be a small device about the size of a credit card, with chipcard contacts at one end and a chipcard reader at the other, as well as an LCD display and several buttons. The customer would place her chip card into it and then insert both devices into the merchant terminal. If the displayed payee and amount meet her expectations, she presses an approval button, releasing the correct PIN to the card, and writing an audit entry. Matters could be arranged so that the customer does not know the true PIN at all, and thus all transactions must be made via the attorney: this protects the customer against attacks by crooked merchants who skim the mag stripe and use that in an overseas ATM with the observed PIN. It also strengthens her hand in the case of a dispute: if the bank says ‘You must have done it, because our system says so and is secure’, she can retort ‘Not at all – my device is even more secure, as it’s evaluated to EAL4 unlike your 20 million lines of crufty old COBOL.’

4 Other Man-in-the Middle Defences

While there is some literature about shared-control processes (such as the privacy guard in CAFE [5]), the protocol community has not yet recognised the middleman as useful. There are of course middlemen in other security spheres: think of the firewalls and virus checkers that mediate between your PC operating system and the outside world. These middlemen’s job is to stay up-to-date with the latest threats and check incoming bits for signs of hostility.

Security APIs – the big brothers of security protocols – could well gain from middleman defences. The Hardware Security Modules (HSMs) at banks that perform PIN processing must conform to dated APIs with fundamental weaknesses in the encrypted data formats (described in detail in [1, 3]). When an HSM must operate in a hostile environment (such as a semi-trusted facilities management firm, or a data centre in a unstable foreign country), an additional middleman is a logical solution to the problem. There apparently isn’t the economic incentive for first-world banks operating their own data centres to push for replacement of the API. But, where needed, a further device can act as a gateway to the banking HSM, observe and filter the transaction stream, stall transactions that look suspicious, and keep an independent audit trail of all activity.

Finally, we considered software middlemen for banking APIs in order to deal with short-term defence against the decimalisation table attack [2]. A hardware equivalent might involve mounting an HSM such as a 4758 inside a PC, and then mounting the PC in a steel box with a tamper-sensing barrier.

5 Composing Middlemen

In their influential paper “Cascade Ciphers: The Importance of Being First”, Ueli Maurer and Jim Massey showed that when ciphers are composed, the resulting cipher is as strong as the first, except in the case where the ciphers commute in which case the composition is as strong as the best. Defensive middlemen work similarly. If both my electronic attorney and an electronic attorney belonging to the Mafia are plugged between my chipcard and a merchant’s terminal, then so long as it’s my attorney that is next to my card, I will be all right; however, if a Mafia-owned attorney is in direct contact with my card, then it can perform arbitrary middleman attacks.

The case of commuting defences is more subtle. In the cipher case, this refers to stream ciphers; in the case of electronic attorneys, one can imagine a number of devices communicating with both card and terminal using a dependable broadcast protocol, so that any bad advice could be countered by denunciation. Something similar may be found in business, where a company thinking of a takeover might have a conference with multiple complementary specialists (bankers, lawyers, brokers, accountants) making suggestions and trying to find flaws in suggestions made by others.

6 Conclusions

The middleman has traditionally been seen as evil by security protocol designers, and attempts are made (often in vain) to exclude him. In real life, however, middlemen are ubiquitous, and we think the time has come for a rethink.

Designers like to aim at an elegant and incorruptible protocol for a broad range of tasks, but then fail for all manner of reasons, from unclear or dishonest assumptions, through shifting goalposts and featuritis to committee design. In the resulting complex, real-world protocols there is a place for a middleman. Consider the law: even a clever and articulate defendant retains a lawyer if he can afford it, since the complexity and volatility of legal protocols make it uneconomic for a nonspecialist to maintain the capability to argue a good case on his own behalf. Similarly, keeping up with computer viruses is a full-time job: no sensible security expert would maintain his own virus checker unless that were his speciality. In the case of EMV, it seems to be more by luck than by judgment that the protocols are open to middleman defenses. However, the economics suggest that they will stay that way. Tweaking the base protocols to make middleman attacks harder would be immensely expensive and take years to roll out, but keeping a middleman up-to-date is much cheaper.

To summarise, the man-in-the-middle defence is a good way to do two things. First, it is a sensible place to introduce a dynamic and upgradeable element which allows a slower but more careful evolution of an underlying protocol, or the retrofitting of protection to a protocol which is too expensive to change. Second, it gives us an opportunity to bring the human back into the protocol where there was no window for manual intervention before.

References

1. Ross Anderson, Mike Bond, Jolyon Clulow, Sergei Skorobogatov, “Cryptographic processors – a survey”, University of Cambridge Computer Laboratory Technical Report TR-641
2. Mike Bond, Piotr Zielinski, “Decimalisation Table Attacks for PIN Cracking”, University of Cambridge Computer Laboratory Technical Report TR-560
3. Jolyon Clulow, “The Design and Analysis of Cryptographic APIs for Security Devices”, MSc Thesis, University of Natal, SA, 2003
4. Ueli Maurer, James Massey, “Cascade Ciphers: The Importance of Being First”, Journal of Cryptology vol 6 no 1 pp. 55–61, 1993
5. Jean-Paul Boly et al., “The ESPRIT Project CAFE – High Security Digital Payment Systems”, in ESORICS 94, Springer LNCS 875 pp 217–230