Multi-channel protocols

Ford-Long Wong and Frank Stajano

University of Cambridge Computer Laboratory

Abstract. We examine several ad-hoc pairing protocols that strengthen their radio exchanges with additional transmissions over another channel, for example a screen showing graphically encoded information to a camera. Additional channels may have limited capacity and may still be subject to eavesdropping, but they may offer specific advantages over radio such as data origin authenticity. A single protocol may profitably use more than one channel, each with its own specific security properties, for different messages in its trace. Making this option explicit allows for further advances in protocol design.
We also present an intriguing asymmetric protocol that achieves results comparable to mutual authentication even though the verification happens only in one direction.

1 Introduction

The problem of authentication between two devices that meet for the first time has been extensively studied and is particularly significant in ubiquitous computing. In this paper we revisit the problem subject to the following constraints. We first assume that the devices cannot rely on a certification authority, on a Kerberos-style authentication server or on any other kind of third-party introducer. We then assume that their main communication channel is radio, meaning that they can never be sure whether they are talking to the expected device or to any other compatible device within range. We also consider that attackers may eavesdrop on the radio channel (obvious) and even insert or modify messages (non-trivial but still possible[1]). Under these constraints, reliable authentication is hard if not impossible.

We make the situation more manageable by assuming the presence of an extra channel between the devices: if, for example, one device has a screen and the other a camera, a suitable graphical encoding can be used to transfer bits from the first to the second over a visual channel. This possibility has obvious practical relevance given the growing popularity of camera phones in recent years. A slight variation on this theme is for the second device to have a keypad, and for the human operator to enter into the second device a sequence of characters displayed on the screen of the first. There is a great variety of options for extra

[1] The simplest case in which this might happen is one in which devices A and B are out of range of each other but both in range of the attacker-in-the-middle M. A more elaborate setting is described by Kügler [7] in the context of Bluetooth: M establishes connections with both A and B but on different hopping sequences.

To appear in
B. Christianson et al. (Eds.): Security Protocols 2005, LNCS.

channels including direct electrical contact, infrared, typing the same PIN into both devices, modulated audio beeps, voice and so on. Several of these, and among them the visual channel available to camera phones, enjoy the useful property of "data origin authenticity" that the radio channel does not possess: the user of the receiving device knows for sure that the received data was sent by the intended source device.

1.1 Contributions

We offer two main contributions in this paper.

The use of more than one channel, each with different security properties, in the context of a single security protocol, is not in itself new: in the context of ad-hoc authentication and pairing the Resurrecting Duckling model by Stajano and Anderson [11] specified that, in contrast with any other messages later exchanged between Mother and Duckling over the wireless channel, imprinting had to be carried out using direct electrical contact. The problem of pairing, and the idea of supplementing a radio interaction with communication over another medium, was revisited by several researchers. Among the most significant recent contributions we recognize at least Gehrmann and Nyberg [3], who use the "manual transfer" channel of typing a code on a keypad; Hoepman [6], who discusses various types of "authentic" and "private" channels; and McCune et al. [8], who apply a visual channel based on cameras and 2D barcodes to the protocols originally proposed by Balfanz et al. [1]. Outside the context of authentication, Stajano and Anderson built the Cocaine Auction Protocol [10] on the primitive of "anonymous broadcast" and then explored channels that would provide it, either natively or through the intermediation of a sub-protocol.

However, with hindsight, there are much older examples of multi-channel protocols: one is the geek practice of printing the hash of one's PGP key on one's business card (which offers higher integrity than the web page with the full key); another, as old as cryptography itself, is the fundamental idea of transmitting the symmetric key using a more secure channel than the one used for the ciphertext. Multi-channel protocols have therefore implicitly existed for millennia. We argue, however, that recognizing them as such is a significant change of perspective.

One of the original contributions of this paper is to explicitly introduce multi-channel protocol modelling as a new powerful design and analysis tool. The other is a surprising multi-channel pairing protocol for the asymmetric situation in which only one of the two devices can acquire a code from the other over the extra channel.

1.2 Structure of this paper

We review and compare a number of pairing protocols that make use of multiple channels, discussing advantages and limitations of the channels used. Our original contributions mentioned above are in sections 8 and 7 respectively.

Section 2 We start with a trivial example of multi-channel protocol by showing that a Diffie-Hellman exchange over radio requires an extra channel to resist middleperson attacks.
Section 3 The limited capacity of the extra channel (how long a PIN will your users be willing to type?) opens the door to a brute-force attack by a middleperson.
Section 4 Cameras and screens in mobile phones provide a channel that, while still of limited capacity, can carry a long enough code to defeat the attack of the previous section. A protocol that uses this technique is SiB.
Section 5 With the proliferation of CCTV, many of the proposed extra channels can no longer be considered as guaranteeing confidentiality. Following this insight, we revisit a protocol built to resist the attack of section 3, MANA III, and show that it becomes vulnerable if the attacker can eavesdrop on the extra channel.
Section 6 We discuss a variant of the previous protocol, broadly equivalent to φKE, that uses short codes on the extra channel but remains secure against middleperson even in the presence of an eavesdropper on the extra channel.
Section 7 We present a new pairing protocol for the asymmetric case of a unidirectional visual channel. It appears incomplete at first sight, which is what makes it interesting.
Section 8 Finally, against the backdrop of the protocol design and analysis of the previous sections, we discuss the general applicability of multi-channel protocol modelling and some of the research challenges of this new field.

2 Diffie-Hellman key establishment over radio

An established session key may be used to authenticate subsequent requests and an encryption key may be derived from it to protect communication confidentiality. During the key establishment process we want to ensure that the key is being established with the intended correspondent. As we said, we assume we cannot rely on a third-party introducer; we therefore exclude the use of static asymmetric-key or symmetric-key certificates.

In the Resurrecting Duckling model [11], trust is bootstrapped from a secret transferred via a secure channel between the two parties. The recommended secure channel is physical contact: it gives a strong guarantee that the shared key has been established between the two chosen devices and no others, with high confidentiality and integrity. It makes cryptography redundant for key establishment. Wired contacts on personal devices, however, are surprisingly expensive in the eyes of manufacturing engineers once we take into account not just the cost of the connectors but the additional board area and the geometrical and ergonomic constraints on industrial design.

It should be clear, however, that carrying out a Diffie-Hellman exchange over RF gives no guarantees about the party with which one is establishing a key. The process is therefore vulnerable to a middleperson attack. Even if each of the two parties successfully challenges the other to prove ownership of the established key, as in steps 5–9 and 10–14 of Protocol Trace 1, the confirmation phase can never prove that the key was established with the desired party.

#	Alice		Bob
		Basic DH	
1	Chooses random a		Chooses random b
2		$-\ g^a\ \rightarrow$	
3		$\leftarrow\ g^b\ -$	
4	$K = (g^b)^a$		$K' = (g^a)^b$
		B challenges A	
5			Chooses random C_b
6		$\leftarrow\ C_b\ -$	
7	$M'_1 = H(K, C_b)$		$M_1 = H(K', C_b)$
8		$-\ M'_1\ \rightarrow$	
9			Verify $M'_1 = M_1$
		A challenges B	
10	Chooses random C_a		
11		$-\ C_a\ \rightarrow$	
12	$M_2 = H(K, C_a)$		$M'_2 = H(K', C_a)$
13		$\leftarrow\ M'_2\ -$	
14	Verify $M'_2 = M_2$		

Protocol Trace 1. Diffie-Hellman with key confirmation.

The obvious remedy is to have steps 8 and 13 take place over an extra channel, such as manual transfer [3] or visual transfer [8], that guarantees data origin authenticity. Manual transfer may be implemented by displaying on the first device a string that encodes the MAC and by having the user type this string into the other device. This channel has limited capacity because it is unpleasantly laborious for human users to transfer long strings manually without making mistakes. One may then have to transmit a truncated MAC. This leads to an attack.

3 Attack against short codes on the extra channel

As pointed out by Gehrmann et al. [4, section 2.3], if the manually transferred authentication code is too short then a middleperson attack is still possible.

The attack is shown in Protocol Trace 2. The column 'Ch' describes whether the step consists of a transfer over the radio (RF) channel or over the manual (M) channel. In the context of this protocol trace, when we say RF we mean a channel subject to eavesdropping and substitution attacks and with no data origin authenticity, but no practical limits on capacity. When we say M we mean a channel offering data origin authenticity but with very limited capacity, of the order of 10–20 bits per message.

As the MACs must be short in order to be transmitted over the M channel, it is computationally feasible for middleperson Carol to search for second pre-images. After intercepting Alice's key contribution (step 2), Carol pretends to Bob that she is Alice and establishes a key with him (steps 3–7).

#	Ch	Alice	Carol	Bob
1		Chooses random a		
2	RF		$-\ g^a\ \rightarrow$	
3			Chooses random a'	
4	RF			$-\ g^{a'}\ \rightarrow$
5				Chooses random b
6	RF			$\leftarrow\ g^b\ -$
7			$K_{bc} = (g^b)^{a'}$	$K_{bc} = (g^{a'})^b$
8				Chooses random C_b
9	RF			$\leftarrow\ C_b\ -$
10			$M_1 = H(K_{bc}, C_b)$	$M_1 = H(K_{bc}, C_b)$
11			Chooses random b'	
12	RF		$\leftarrow\ g^{b'}\ -$	
13		$K_{ac} = (g^{b'})^a$	$K_{ac} = (g^a)^{b'}$	
14			Finds C'_b s.t. $H(K_{ac}, C'_b) = M_1$	
15	RF		$\leftarrow\ C'_b\ -$	
16		$M'_1 = H(K_{ac}, C'_b)$		
17	M		$-\ M'_1\ \rightarrow$ $\rightarrow$	$\rightarrow$
18				Verify $M'_1 = M_1$
19		Chooses random C_a		
20	RF		$-\ C_a\ \rightarrow$	
21		$M_2 = H(K_{ac}, C_a)$	$M_2 = H(K_{ac}, C_a)$	
22			Finds C'_a s.t. $H(K_{bc}, C'_a) = M_2$	
23	RF			$-\ C'_a\ \rightarrow$
24				$M'_2 = H(K_{bc}, C'_a)$
25	M		$\leftarrow$ $\leftarrow$	$\leftarrow\ M'_2\ -$
26		Verify $M'_2 = M_2$		

Protocol Trace 2. Middleperson Attack on Short MACs.

At this point Bob wishes to challenge his RF correspondent, whom he hopes to be Alice; the verification code will be received over the extra channel (step 17) and will therefore undeniably come from Alice. What does Carol do to fool Bob?

After receiving Bob's challenge C_b in step 9 and computing the keyed hash value M_1 in step 10 from the session key shared with Bob, Carol forms a session key with Alice (steps 11–13) and performs a brute force search (step 14) to find a challenge C'_b such that the keyed hash value M'_1 derived from it equals M_1. She sends the forged challenge C'_b to Alice (step 15). Alice computes M'_1 (step 16) and shows this result over the manual transfer channel (step 17) to Bob, who verifies it (step 18) against his computed result M_1 and finds that it matches. Bob has been fooled.

Carol then performs the same forgery in the symmetrical situation of Alice challenging Bob (steps 19–26), fooling Alice as well.

The effort required by Carol to attack each challenge-response is of the order of 2^r trials, where r is the bit length of each short MAC. Assuming an adversary with powerful computing resources who is able to perform 1 billion trials a second, and a device time-out of 10 seconds, Gehrmann et al. [4] calculate that a 48-bit code is needed to defeat this attack. But manually transferring 48 bits (which correspond to 12 hexadecimal digits) is tedious and prone to error. One alternative is to use an extra channel of greater capacity.

4 SiB and the camera-phone visual channel

It is possible to acquire the code from the screen with a camera instead of typing it on the keypad. The number of bits that can be reliably transferred is slightly greater and usability improves significantly [12].

At the time of writing, the camera phones commercially available in Europe have reached resolutions of $1280 \times 1024 = 1.3$ megapixels (3.2 megapixels in Asia). In reality, the limiting factor for data transmission is not so much the camera resolution but the screen resolution, which is the lower of the two. Screen resolutions have reached 66 kilopixels, though 36 kilopixels are still more common. Based on these figures, with a suitable 2D encoding the screen-to-camera channel can reliably provide about 40 to 100 bits per message. This is still not enough for a full hash but it is sufficient for a longer code that would solve the problem described in the previous section.

We built a prototype on a Nokia Series 60 handphone, using 2-track SpotCodes [5] that provided 46 bits. We used Diffie-Hellman over elliptic curve groups. The protocol we developed, which we shall discuss in section 6, was later discovered to be basically equivalent to Hoepman's φKE [6]. In our implementation, one run of the protocol took several seconds. A good fraction of this time was taken by aligning the camera phone with the other phone displaying the code. Current phones are usable but not optimal for this task: there are problems of focusing distance, resolution and illumination.

The idea of transferring short cryptographic codes visually between camera phones was originally proposed by McCune et al. [8]. Their SiB protocol, based on earlier work by Balfanz et al. [1], closes the vulnerability pointed out in section 3 by transferring a longer code over the extra channel. The protocol would still fail if the attacker could find a preimage but, because the camera phone channel used by the authors allows 68 bits per transfer (well over twice the capacity of a manually typed PIN), the brute force search is no longer feasible in real time. In other words, SiB requires at least a "medium length" code, not a "short" code[2].

[2] Our very informal semantics for "short", "medium" and "long" codes in this context are as follows. "Short" is a code that can be brute-forced in a few seconds, during a run of the protocol, for example 10 bits. "Medium"is a code that can be brute-forced in an hour, a day or a month but not in real time during a protocol run, for example 50 bits. "Long" is a full length code that, assuming the hash function is not otherwise broken, cannot be brute-forced in hundreds of years, for example 250 bits. If we were being more formal about these terms then we would say something

5 MANA III and eavesdropping on the extra channel

There is however another threat. In protocols that implicitly or explicitly use an additional channel, as happens in many EKE variants in which a strong session key is formed from a weak PIN, there is often the assumption that the additional channel is somehow "local" and safe from eavesdropping.

We argue that, for the manual (screen and keypad or keypad and keypad) and for the visual (screen and camera) channel, this assumption is no longer realistic with the current proliferation of CCTV cameras, both indoors and outdoors. In many cases such cameras even operate covertly, hidden behind opaque domes that allow them to pan and zoom without the victims knowing where the cameras are pointing.

In this section we discuss the consequences of this change in the attacker model.

The MANA III scheme by Gehrmann et al. [4], developed as a variant of Larsson's SHAKE[3], is shown in Protocol Trace 3. It aims to establish that both parties have correctly received each other's public key. It complements the radio transmissions with an exchange of short codes using manual transfer. As the authors themselves say, "Informally, the security of the scheme relies on the fact that R remains secret to the attacker (it is never sent over the air)...". In the presence of a passive attacker in the optical domain, which we believe can no longer be dismissed, this protocol can be cracked.

I_A and I_B are identifiers of Alice and Bob respectively and are publicly known. D is a data string formed from the concatenation of Alice's and Bob's public keys g^a and g^b. K_1 and K_2 are long keys. M_1 and M_2 are long MAC values formed using the function m. R is a short randomly selected string shared between Alice and Bob over the manual channel. Alice will only send K_1 after she has received M_2, and Bob will only send K_2 after he has received M_1.

Crucially, after the verification (step 14), each device must signal whether the verification succeeded (steps 15 and 16), over a channel (e.g. red/green LED) guaranteeing integrity and data origin authenticity. Although the M channel could be reused here, we indicate this channel as L (LED), rather than M, to point out that its requirements are less demanding than those of the channel used in step 6. In particular, its required capacity is only one bit per message. Note the additional subtlety that the LED is signalling to the *operator* of the device(s), not to the other device directly.

As noted in the original paper, if Bob were not told that Alice's verification failed, middleperson Carol could send Alice a random M_2 in step 10, grab K_1 from Alice in step 11, ignore the rest of the protocol run with Alice, find R by brute force and successfully impersonate Alice to Bob, who would not notice the forgery.

about the distinction between finding collisions and finding second preimages; but we aren't, so we don't.

[3] This work is cited in the bibliography of the Gehrmann paper but we have not been able to obtain a copy of the Larsson paper or even to confirm the correctness of the citation.

The reason why the attack of section 3 no longer works is essentially because the short code exchanged over the extra channel is not the MAC but the challenge: the MAC, now transmitted over radio, is full length and not subject to second preimage attacks. The protocol's security, however, relies on the challenge R being kept secret from the middleperson attacker.

#	Ch	Alice		Bob
1		Chooses random a		Chooses random b
2	RF		$- g^a \rightarrow$	
3	RF		$\leftarrow g^b -$	
4		$D = g^a \mid g^b$		$D = g^a \mid g^b$
5		Chooses random R		
6	V		$- R \rightarrow$	
7		Chooses random K_1		Chooses random K_2
8		$M_1 = m_{K_1}(I_A \mid D \mid R)$		$M_2 = m_{K_2}(I_B \mid D \mid R)$
9	RF		$- M_1 \rightarrow$	
10	RF		$\leftarrow M_2 -$	
11	RF		$- K_1 \rightarrow$	
12	RF		$\leftarrow K_2 -$	
13		$M_2' = m_{K_2}(I_B \mid D \mid R)$		$M_1' = m_{K_1}(I_A \mid D \mid R)$
14		Verifies $M_2 = M_2'$		Verifies $M_1 = M_1'$
15	L		$-$ outcome $\rightarrow$	
16	L		$\leftarrow$ outcome $-$	

Protocol Trace 3. MANA III.

If this assumption is violated, the attack is as follows. Assume that middleperson Carol is able to observe the string R being keyed into the devices. She may then send modified public keys to both Alice and Bob, such that Alice's and Bob's copies of D are different from each other's, but match the two copies held by Carol. Thereafter, Carol can individually choose different K's and generate the M's so as to authenticate successfully with both Alice and Bob. Note that this attack is independent of the length of the MACs.

6 Short codes and eavesdropper resistance

Sections 3 and 4 have shown protocols that can be cracked if the attacker can brute-force in real time the short code sent over the extra channel; they therefore require at least a "medium length" code. Section 5 has shown a protocol that resists brute force even with a short code, but which is vulnerable if the attacker can eavesdrop on the extra channel. Is it possible to come up with a protocol that transmits only short codes (rather than "medium" ones) on the extra channel but, despite that, is not broken by eavesdropping?

We developed such a protocol as a variant of MANA III and presented it at the workshop, although we later discovered that it was essentially equivalent to Hoepman's earlier φKE [6]. In both cases, the extra channel (whether

screen-to-camera or screen-to-keypad) is exploited for its integrity and data origin authenticity rather than for its confidentiality.

For our proposed protocol, given in Protocol Trace 4, the core objective remains the same—to assure that a session key is being established with the correct party. The pre-conditions are an RF channel having low confidentiality and low integrity, and a bandwidth-limited optical channel having low confidentiality but high integrity and high data origin authenticity.

R_a and R_b are short random nonces. K_a and K_b are long nonces. H_1, H_1', H_2 and H_2' are long hashes.

#	Ch	Alice		Bob
1		Chooses random a		Chooses random b
2	RF		$- g^a \rightarrow$	
3	RF		$\leftarrow g^b -$	
4		Chooses random R_a		Chooses random R_b
5		Chooses random K_a		Chooses random K_b
6		$H_1 = H(I_A \mid g^a \mid g^b \mid R_a \mid K_a)$		$H_2 = H(I_B \mid g^b \mid g^a \mid R_b \mid K_b)$
7	RF		$- H_1 \rightarrow$	
8	RF		$\leftarrow H_2 -$	
9	V		$- R_a \rightarrow$	
10	V		$\leftarrow R_b -$	
11	RF		$- K_a \rightarrow$	
12	RF		$\leftarrow K_b -$	
13		$H_2' = H(I_B \mid g^b \mid g^a \mid R_b \mid K_b)$		$H_1' = H(I_A \mid g^a \mid g^b \mid R_a \mid K_a)$
14		Verifies $H_2 = H_2'$		Verifies $H_1 = H_1'$
15	L		$-$ outcome $\rightarrow$	
16	L		$\leftarrow$ outcome $-$	

Protocol Trace 4. Our MANA III variant.

Each party generates an ephemeral Diffie-Hellman private value, computes the corresponding public key and sends it to the other party. Alice chooses a short random R_a (step 4), a long random K_a (step 5), and hashes the concatenation of these with her identifier I_A and the public keys, into a long hash H_1 (step 6). Bob does likewise and produces a long hash H_2. Next, Alice and Bob both send over the RF channel their computed hashes to each other, which represent their commitments (steps 7 and 8). Bob must indicate that he has received a hash, and only then, and not before, Alice may release R_a over the visual channel and K_a over the radio channel. Similarly, Alice must indicate that she has received a hash and only then, and not before, is Bob allowed to release R_b and K_b. After all the R and K have been received, both sides proceed to compute the hashes and verify that they match the copies they had received earlier in steps 7 and 8.

The length of the long hashes determines the size of the complexity theoretic problem a potential middleperson attacker would face for finding their second pre-images. The length of the visually exchanged R values determines the probability or "luck" the attacker would have in choosing coincidentally the same R

values for the commitments as Alice and Bob might later choose. (We'll get back to this at the end of this section.)

The protocol is symmetric: steps 7, 9 and 11 prove to Bob that he is communicating with Alice; conversely, steps 8, 10 and 12 prove to Alice that she is talking to Bob. If one set of steps is absent, the authentication is only unilateral. We explore this case in section 7.

Compared to MANA III, this protocol relies on the strong data origin authenticity property of the extra channel rather than on its confidentiality: when an R value is exchanged, we have high confidence that it originated from the observed party. The difference is that, here, both parties must issue their commitments H before the release of any of the R and K values. Therefore an attacker Carol who manages to observe the R values will be too late to compromise the key agreement, because she must have already committed to a fake H for which she will not be able to generate a matching K.

One may wonder why we need the K values, if the unforgeable R values are there. This is because, if there were no K_a, middleperson Carol could otherwise intercept Alice's H_1 in step 7 and try all possible values for R_a until she found the one that produced the correct hash. At that point Carol would be able to substitute her own key $g^{\tilde{a}}$, compute the hash $H(I_A \mid g^{\tilde{a}} \mid g^b \mid R_a)$ and send it to Bob. Since the R_a is the genuine one that Alice will later disclose, Bob will find that the H_1' he computes in step 13 will match this one he received from Carol in step 7 (all the inputs are the same). So the K values are there to prevent Carol from brute-forcing the R values out of the H values.

If step 14 completes with successful mutual verification of the hashes, both parties will have high confidence that the party from whom each has visually obtained the R value is the same party from whom each has received a public key. As in the original MANA III protocol, both devices must finally indicate (steps 15 and 16) whether the verification succeeded or not: each device should only consider the protocol run successful after receiving proof that the *other* device also succeeded during step 14.

As hinted at above, the middleperson attacker Carol has basically two options. In the first option, she guesses an R value, inserts a modified public key and a hash computed from a random K value, and then hopes that the spoofed party will coincidentally choose the same R value. The probability of this attack succeeding is 2^{-r} where r is the bit length of the R value. Carol has less than 1% chance of success for an R as short as 7 bits. In the second attack option, Carol inserts a modified public key and a random hash. After the R value is disclosed by the spoofed party, the attacker embarks on a search for a K value which can yield the hash she has already committed to. The complexity of such a search is of the order of 2^h where h is the bit length of the hash. Since we said H was "long", this is by definition infeasible.

Thus the protocol is strong even under the model of a powerful attacker who is able to eavesdrop on the extra channel and rewrite messages on the RF channel, and in a situation in which the extra channel can only carry a "short" (not even "medium") payload.

7 Asymmetric pairing

#	Ch	**Alice** (mother duck)		**Bob** (duckling)
0	PW			Start imprinting
1		Chooses random a		Chooses random b
2	RF		$- g^a \rightarrow$	
3	RF		$\leftarrow g^b -$	
4				Chooses random R_b
5				Chooses random K_b
6				$H_2 = H(I_B \mid g^b \mid g^a \mid R_b \mid K_b)$
7	RF		$\leftarrow H_2 -$	
8	PB		$-$ ack $\rightarrow$	
9	V		$\leftarrow R_b -$	
10	RF		$\leftarrow K_b -$	
11		$H'_2 = H(I_B \mid g^b \mid g^a \mid R_b \mid K_b)$		
12		Verifies $H_2 = H'_2$		
13	PB		$-$ outcome $\rightarrow$	

Protocol Trace 5. Asymmetric pairing.

Now imagine the case in which the devices are not peers and the visual channel can only be established in one direction. For example, one device is a large stand-alone screen with some local processing power; it sits in a shop window and displays a pre-programmed sequence of text and graphics. The other device is a PDA that, every week or two, uploads a new sequence into the screen over radio.

The screen needs to be imprinted to the PDA of the shopkeeper so as to prevent anyone else from uploading messages to the screen. We assume that the PDA has a camera but the screen doesn't; and that, owing to industrial design constraints, it is not possible to use a wired connection between the two. Our goal is to devise a sufficiently secure method to perform the Resurrecting Duckling's imprinting operation in the absence of a wired contact.

Taking Alice as the mother duck PDA and Bob as the duckling screen, we cannot perform all the exchanges in Protocol Trace 4 because the visual channel only works from B to A; the message in step 9 cannot be sent and this cancels out the whole subprotocol in which A acts as prover and B as verifier (steps 7, 9, 11, 16 and Bob's half of steps 13 and 14).

The bits we can still do are in Protocol Trace 5. After successful completion, Alice the PDA is assured that she has established a key with Bob the screen, but Bob receives no proof that he is being imprinted to the correct PDA. This seems incomplete, which is what makes this protocol interesting.

What we wish to avoid is for Bob to be persuaded to imprint itself to another device Carol. How can this be stopped if Bob knows nothing about the device with which it is pairing? In the Resurrecting Duckling policy, introduced in [11]

and formalized in [9, section 4.2.5], Bob the duckling imprints itself to the first mother duck he sees, whoever she is. What we want here is to prevent Carol from appearing in front of Bob for imprinting before he has a chance to see Alice. A crucial element of the solution is the presence of a human operator who wishes to imprint Bob to Alice.

Although manufacturers would love to get away with a Bob that had no other inputs than a wireless interface, we believe we also need at least the following:

1. a way to ask Bob to start imprinting;
2. a way to tell Bob whether to proceed or not, before committing to a proposed imprinting.

These two input mechanisms must be available only to a human operator Hermione having physical control of device Bob. The intention is to construct a protocol that cannot be subverted by hidden middleperson device Carol so long as human operator Hermione has physical control of duckling device Bob during the imprinting phase. Once imprinting is over, duckling device Bob may be left unattended: the Duckling policy will ensure that it can't be taken over by Carol or anyone else unless the mother duck device Alice first voluntarily relinquishes control.

Mechanism 1 could be implemented as nothing more than the act of switching on device Bob when he is still in his imprintable state. This is indicated (very poorly) as step 0 in the trace, with PW indicating the "power" channel. Mechanism 2, on the other hand, could be implemented as two mutually exclusive pushbuttons (yes/no, ok/cancel, proceed/abort...), indicated as channel PB in step 13.

The exchange presented in Protocol Trace 5, obtained from Protocol Trace 4 by removing the steps in which Alice authenticates to Bob[4], proves to Alice that she and Bob are using the same two public keys g^a and g^b. Once human operator Hermione is satisfied that device Alice completed her verification succesfully in step 12, Hermione presses the "yes" pushbutton (step 13) on duckling device Bob, thereby ordering Bob to compute and commit to the imprinting key g^{ab}. If Hermione observes that Alice's verification failed, she presses pushbutton "no" (or lets Bob abandon the protocol by timeout[5]) and Bob forgets the previous exchange and remains imprintable.

An unattended attacking device Carol, with ability to eavesdrop on the V channel and with ability to rewrite messages on the RF channel, cannot imprint Bob to herself unless she can also *press* the "yes" pushbutton used in step 13 to commit the imprinting. Even if Carol had a mechanical finger that allowed

[4] Note that we had to introduce a "content-free" step 8 to maintain synchronization. Bob should only display R_b after being sure that Alice received a hash. In Protocol Trace 4, this was achieved implicitly by Alice having to send something useful in step 9. Here, even though she has nothing useful to send at that stage, she must still signal to Bob, over the unforgeable extra channel, that she received the hash and that he can proceed.

[5] This could also be exploited as a way to allow just one pushbutton rather than two.

her to press Bob's button, it is expected that Hermione would notice this and disallow it—that's the point of Hermione "having physical control" of Bob.

This protocol is interesting because it seems incomplete. Alice never proves herself to Bob. Bob doesn't actually know with whom he paired. Something appears to be missing. And yet, it works: Bob can only pair with the correct Alice (even if he can't recognize her) because Hermione won't let him proceed otherwise.

In the protocol of section 6, we achieved mutual authentication with a bidirectional "short" extra channel. In this protocol we show that we can achieve the same result even with a unidirectional "short" extra channel channel coupled with an even shorter "one bit only" extra channel in the opposite direction. In a sense, Bob is delegating his trust to Alice and Hermione.

Note that this core idea (tricky delegation-based mutual authentication despite asymmetric extra channel) could have been demonstrated with a much simpler protocol if the unidirectional extra channel had been allowed to be "long" rather than "short"; but this is true of most of the other protocols we discussed, which would have all basically reduced to a Diffie-Hellman augmented with unforgeable transmission of the hashes of the keys.

8 Multi-channel protocols: towards more expressive protocol modelling

We have looked at a variety of modern protocols that make use of multiple channels. As we noted in section 1.1, multi-channel protocols have been implicitly used for thousands of years but it is time to recognize that thinking explicitly in terms of multiple channels is a powerful technique for protocol design and analysis. We shall now highlight our future work plans for expressive modelling of multi-channel protocols.

To reason accurately about multi-channel protocols we need first of all a good **notation**. The protocol traces in this paper have used line-by-line listings with an explicit indication of the channel used in each step; while we believe this to be an improvement over previous practice, this notation is still not satisfactory and does not capture all the relevant details.

For example, messages 17 and 25 in Protocol Trace 2 are transmitted over the manual channel M and therefore, unlike the others, cannot be rewritten by Carol. We have shown them as "going through" Carol but this is not entirely accurate. This notation does not clearly distinguish whether Carol may or may not observe them—indeed, in section 3, where that protocol trace appears, we still assume she can't, while in section 5 we assume she can.

Stajano and Anderson [10] discussed channels whose transmission primitive was point-to-point send, point-to-domain broadcast or point-to-domain anonymous broadcast and then proposed a notation to make such distinctions explicit. Unfortunately it is not sufficiently general for analyzing all possible multi-channel protocols.

The intermediation and consent of a locally present human, featured in the last step of Protocol Traces 3, 4 and 5, should also be captured more explicitly by the notation. It might perhaps be abstracted away through a more rigorous definition of the associated one-bit channel, but the special role of Hermione in the asymmetric protocol of section 7 must also be taken into account.

The interaction with the human operator should also be described more clearly in step 0 of Protocol Trace 5: it is human Hermione who forces device Bob to start the imprinting, but this is not clear from the protocol trace which only describes the interactions between devices Alice and Bob.

There should also be a more explicit mention of temporal dependencies. In Protocol Trace 4, for example, there are pairs of messages that might be swapped without affecting the protocol (for example 2 and 3); but there are other cases in which return messages are implicitly used as ACKs, for example when Alice's reception of Bob's message 8 is significant not just for its contents but also because it signals that Bob received Alice's message 7, meaning that Alice may now release R_a over the visual channel. See what happens in Protocol Trace 5 when we remove the interleaved messages: it is necessary to reintroduce an "empty" message precisely because we need an ACK before proceeding (cfr. footnote 4). A notation that specified which dependencies really matter, perhaps inspired by PERT charts, might be more expressive and would highlight alternative linearizations, possibly leading to more efficient (e.g. shorter, when two adjacent messages may be collapsed into one) message sequences.

The notation should also feature a legend detailing the properties required of the various channels used. In this paper we have discussed primarily confidentiality, integrity, user-friendliness, origin authenticity and carrying capacity. There are of course other properties, including at least anonymity, covertness, range, cost (equipment costs and running costs) and latency. For some of these properties a boolean qualifier makes no sense; for most of the remaining ones it is at best an oversimplification. And then there are the trade-offs: for a channel of limited carrying capacity per message, one could increase that capacity by sending several messages in a row, probably at the expense of latency, usability and running costs. It will be hard to settle on the correct level of abstraction—not so generic that is becomes content-free but not so detailed that it becomes unusable.

Following on from this, while in this paper we have used ad-hoc descriptions in the main text, it would be useful to have a uniform and coherent **taxonomy** of channel properties and of channels so as to be able to compare the channel requirements of any two protocols. Listing such requirements explicitly will clarify whether a given protocol may be run over any other channels than the ones originally intended—perhaps new ones that had not even been invented when the protocol was designed. More importantly, it will make the protocol authors more aware of the details of their design, which in itself may help avoid some errors. The expressiveness and clarity of the notation used to express an idea is an important factor in avoiding implementation errors, as is well known from programming languages.

Note also that, once its properties have been stated, a complex channel may be implemented on top of a more primitive channel that had different properties: see the discussion of anonymous broadcast in the cited Cocaine Auction paper but also all the composite channels offered by the likes of IPSEC and SSL on top of a basic packet network.

Finally, the last and perhaps the most important tool we need is a **logic** for multi-channel protocols in the spirit of BAN [2]: a framework of simple rules (sending a message on a certain channel leads to certain consequences, properties and beliefs for the various parties involved) that either allows a protocol designer to prove that a protocol achieves its goal or, if it can't, gives clues as to why it probably doesn't.

9 Conclusions

With this paper we explicitly open up the field of multi-channel protocols. Until now, protocols were described primarily in terms of sender, recipient, payload and sequencing order. Instead, it may be advantageous to send different messages of the protocol over different channels and take into account the different properties of the various channels.

We have presented a surprising asymmetric pairing protocol: by carefully exploiting the properties of its channels, it provides mutual authentication even if the proof only happens in one direction.

Finally, we have charted in detail the plans for some future work that will enable effective modelling of multi-channel protocols. We believe that multi-channel protocols will become a fertile new field for security protocols research.

10 Acknowledgements

Thanks to Alf Zugenmaier for pointing us to Hoepman's work.

References

1. D. Balfanz, D. K. Smetters, P. Stewart and H. C. Wong. "Talking to strangers: authentication in ad-hoc wireless networks". *Network and Distributed System Security Symposium*, Feb 2002.
2. Michael Burrows, Martín Abadi and Roger Needham. "A Logic of Authentication". Tech. Rep. 39, DEC SRC, Feb 1989.
3. C. Gehrmann and K. Nyberg. "Enhancements to Bluetooth Baseband Security". *Proc. Nordsec 2001*, Nov 2001.
4. Christian Gehrmann, Chris J. Mitchell and Kaisa Nyberg. "Manual authentication for wireless devices". *Cryptobytes*, **7**(1):29–37, 2004.
5. HighEnergyMagic. "SpotCode", 2004. `http://www.highenergymagic.com/`.
6. Jaap-Henk Hoepman. "The Ephemeral Pairing Problem". In "Proc. 8th Int. Conf. Financial Cryptography", No. 3110 in LNCS, pp. 212–226. Springer, 2004.

7. Dennis Kügler. "Man in the Middle Attacks on Bluetooth". In "Proc. Financial Cryptography", No. 2742 in LNCS, pp. 149–161. Springer-Verlag, 2003.
8. Jonathan M. McCune, Adrian Perrig and Michael K. Reiter. "Seeing is Believing: Using CameraPhones for Human-Verifiable Authentication". Tech. Rep. CMU-CS-04-174, Carnegie Mellon University, 2004.
9. Frank Stajano. *Security for Ubiquitous Computing*. John Wiley and Sons, 2002.
10. Frank Stajano and Ross Anderson. "The Cocaine Auction Protocol — On The Power Of Anonymous Broadcast". In "Proc. 3rd Int. Workshop on Information Hiding", No. 1768 in LNCS. Springer-Verlag, 1999.
11. Frank Stajano and Ross Anderson. "The Resurrecting Duckling — Security issues for Ad-Hoc Wireless Networks". In "Proc. 7th International Workshop on Security Protocols", No. 1796 in LNCS. Springer-Verlag, 1999.
12. Eleanor Toye, Anil Madhavapeddy, Richard Sharp, David Scott and Alan Blackwell. "Using camera-phones to interact with context-aware mobile services". Tech. Rep. UCAM-CL-TR-609, University of Cambridge Computer Laboratory, December 2004.