The Resurrecting Duckling — what next?

Frank Stajano

AT&T Laboratories Cambridge
http://www.uk.research.att.com/~fms/
and
University of Cambridge Computer Laboratory
http://www.cl.cam.ac.uk/~fms27/

Abstract. In the context of the security of wireless ad hoc networks, we previously explored the problem of secure transient association between a master and a slave device in the absence of an online authentication server. We introduced the *Resurrecting Duckling* security policy model to address this problem.
Master-slave relationships, however, do not exhaust the range of interesting interactions. We therefore extend the Duckling model to also cover relationships between peers.

1 The Duckling: why, what, and what's missing

The range of devices that contain a microprocessor is continually expanding in every field — from consumer goods to office equipment, "white goods", vehicles and medical and scientific instrumentation. Looking ahead, the next development after endowing every device with a processor is going to be to allow all these computing nodes to communicate with each other, enabling them to co-operate and take advantage of each other's services. For convenience, in many cases this connectivity will be wireless: devices will be able to talk to each other as required by forming short-lived *ad hoc wireless networks*.

One respect in which such networks are fundamentally different from their well studied more traditional cousins is the absence of online servers for functions such as authentication. Your digital camera and your electronic organiser may spontaneously decide to communicate at any time, for example while you are taking pictures in the middle of the desert, and the ad hoc network they establish will be completely local, with no backbone infrastructure to connect it to the Internet or to anything else. This means that the problem of authentication can no longer be solved in the traditional way. The symmetric cryptography solutions in the tradition of Needham-Schroeder, Otway-Rees, Kerberos etc. explicitly require an online ticket-granting server; and even the solutions based on public key cryptography and signed certificates eventually fail if the certification authority is not online, due to the difficulty of performing timely revocation[1].

[1] Certificates may certainly be marked with an expiration date, and the interval between renewals may be made sufficiently short that timely revocation becomes pos-

B. Christianson, B. Crispo and M. Roe (Eds.). *Security Protocols, 8th International Workshop Proceedings*, Lecture Notes in Computer Science, 2000.

In a previous work [7,8] we highlighted *secure transient association* as the fundamental authentication problem in this scenario: one principal, for example the universal remote control of your electronic house, needs an association with another principal, for example your garage door or your hi-fi (or indeed both, and more). This association needs to be *secure*, in the sense that you don't want anybody else with the same type of controller to be able to open your garage door or turn on your hi-fi from the street in the middle of the night, but it also needs to be *transient*, in the sense that you want to be able to undo it when you decide to resell your hi-fi to buy a better one, without for that being also forced to resell your garage door, your television and your refrigerator.

The solution we proposed is formalised in the *Resurrecting Duckling* security policy model, which we shall now summarise. The slave device is the duckling, while the master controller acts as its mother duck. The duckling may be in one of two states, imprinted or imprintable, depending on whether it contains a soul or not; it starts (pre-birth) as imprintable, becomes imprinted at birth when a mother duck[2] gives it a soul, and it becomes imprintable again on death, when the soul dissolves. The soul is a shared secret that binds the duckling to its mother: as long as the soul is in the body, the duckling will stay faithful to the mother and obey no one else. Resurrection is allowed, as the name of the policy suggests, but the duckling's metempsychosis works in reverse: instead of one soul inhabiting successive bodies, here we have one body hosting a succession of souls. The soul is originally transferred from mother to duckling over a non-wireless channel[3] (e.g. electrical contact) in order to bootstrap the rest of the protocol. Death, which makes the duckling imprintable by a new mother, may be triggered by the conclusion of the current transaction or by a deliberate order from the mother duck ("commit suicide now!"), but not by one from an outside principal[4]. The mother duck should backup the soul with local escrow parties since, if the soul is lost (for example because your dog chews on the remote control), the duckling will be unresponsive to any other principal and it will be impossible to reset it to the imprintable state.

This model expressively describes a great variety of interesting situations, not just the relationship between a remote control and an array of household appliances. It describes, for example, the bond between a wireless thermometer and the doctor's PDA that records graphs of the temperatures of the various patients; there, death of the thermometer duckling occurs at the end of the

sible, but the cost of this strategy is indeed in the necessity for frequent renewals of the certificates, which must be propagated to all the devices in the field — an expensive proposition if, as we assumed, the devices have no online connection to the server. And we haven't even mentioned the subtle issues to do with the requirement for secure clocks.

[2] *Any* mother duck, actually. That's the point of imprinting: any entity the duckling sees at birth is taken as being the mother duck, even if it looks like Konrad Lorenz.

[3] This is an informal way of saying "over a channel whose confidentiality and integrity are axiomatically guaranteed".

[4] This is to say that attempts to kill the duckling should damage its body or otherwise be uneconomical compared to buying a new, imprintable duckling.

transaction, when the thermometer is returned to the bowl of disinfectant. It even describes a possible mode of interaction between an e-wallet and an e-ATM: as the banking customer gets near the e-ATM, she wants the machine to imprint itself to her e-wallet and no one else's; but at the end of the transaction, the e-ATM duckling dies and is again imprintable to any other customer. It is also a representation of the relationship between a computer and its superuser, the soul being here the superuser password: when the computer ships, it is imprintable, in that anyone can become its superuser by installing the operating system and supplying a master password; but, once that is done, nobody can become mother duck unless the current superuser voluntarily relinquishes control[5].

There are however a number of other equally interesting situations that the model so far described does not adequately cover. All the above cases involved a definite master-slave relationship between the mother and the duckling, but we can envisage cases of ad hoc networks between devices that it would be more natural to consider as peers. If the components of your hi-fi and video system talk to each other, for example because the timer wants to start the satellite TV tuner and the DVD writer in order to record something off air, or because the DVD player wants to tell the TV that it should set the aspect ratio to widescreen for this programme, does it make any sense for the DVD player to become the mother duck of the television?

The new work presented here extends the Resurrecting Duckling model to cope with such peer-to-peer cases.

2 The many ways of being a master

2.1 Human or machine?

The first interesting remark concerns the nature of the principals. The master-slave model so far presented seems to make sense primarily when the mother duck master is a person and the duckling slave a peripheral that the person wishes to use. The master initiates imprinting of the slave (including the physical contact step) and then starts giving it orders. This pattern tends to suggest that the master, even though it is a physical device (the remote control), is actually only the cyber-representative of a sentient being (the person who owns the remote control, the television, the fridge and all the rest of the equipment).

Blurring the distinction between the person and the computer that represents it is a common sin, which may sometimes have unexpected consequences.

[5] There is a noteworthy subtlety here: if we consider the duckling to be just the hardware of the computer, then this system does not properly follow our security policy model, because it is trivial for a thief to kill the duckling by taking out the hard disc and reformatting it in another computer, thereby returning the duckling to the imprintable state. If however we consider the installed software to be part of the duckling, i.e. if the value of the computer is more in its software, configuration and data than in its hardware, then a computer with encrypted file system does follow the policy, because reformatting the disc will damage the duckling's body — here taken to include the data and installed software.

We shall remedy by being more precise and identifying two separate interactions: the one between the remote control as master and the DVD as slave is one where the principals are both computing devices; but on top of that there is another relationship between the owner as master and the remote control as slave, in which one principal is a human and the other is machine. The interesting point is that this second relationship, too, can be modelled with the Resurrecting Duckling: the virgin remote control gets imprinted to its owner on purchase when the owner types in a PIN. We thus have a hierarchy of master-slave duckling relationships: the human is mother duck to this remote control and possibly other cyber-representatives (for example the e-wallet, to use another one of our previous examples), while the remote control is in turn mother duck to a number of devices (DVD, hi-fi, garage door etc.). Each principal (whether man or machine) has control over all the principals in the subtree of which it is root[6] — but such control can only be exerted with the co-operation of all the principals down the relevant chain of command: I may be the mother duck of my remote control, in turn mother duck of my DVD player, but if I break the remote control I will not be able to play any DVDs despite being the grandmother duck of the player (unless I restore the relevant imprinting keys from the local backups).

2.2 Smart dust

Before going any further we should introduce the application scenario that originally inspired me to extend the Duckling model.

Take a wireless network not of a few nodes but of several thousand; scale the nodes down in volume from 10,000 mm^3 (i.e. a few cm across) to 1 mm^3; throw in some extra science fiction such as laser-based optical communications between those microscopic gizmos; what you get is a rough approximation to what the wizards at Berkeley are developing under the heading of "smart dust".

The system [6] consists of autonomous millimetre-sized sensor nodes, the "dust motes", that can be scattered in great quantities over the area to be monitored. Each dust mote consists of battery, solar cell, sensors, micromachined catadioptric mirror (which can reflect or not reflect an incoming laser ray towards its sender, thus passively transmitting one bit) and some digital computing equipment, plus extra optionals such as an active transmitter and a receiver (in their absence, the node consumes less power and lasts longer, but it can only talk to a larger entity such as a better equipped dust mote or a base station).

In one example scenario, a cloud of dust motes is dumped on the battlefield from a military aircraft; later a base station with a laser and a high-speed video camera acquires the sensor results from a safe distance, for example to detect the passage of vehicles or the presence of toxic gases. It is also envisaged that the better endowed dust motes might talk to each other in order to route data from motes that don't have direct line of sight to the base station.

[6] Since the link from a node to its parent is indeed a representation of a duckling-to-mother relationship, this graph can be viewed as a duck family tree. So each principal has control over all its offspring.

At this early stage in the project, manufacturing the devices and devising the appropriate low-level communications and routing protocols so that they work at all are, quite reasonably, the primary concerns, and the security issues appear not to have been tackled yet. If the White general deploys his dust motes, how can he be sure that the sensor readings he gets are good ones from his own White dust motes and not fake ones from the much more numerous Black dust motes that his adversary has cunningly deployed over the same area? And, for dust motes that have the capability of talking directly to their neighbours, how is the mutual authentication problem solved?

Once we realise that this is, in fact, a low-power ad hoc wireless network, only with some of the numbers off in unexpected directions by a few orders of magnitude, it becomes plausible to think that the Resurrecting Duckling might be of help. But something is still missing. The dust motes are certainly peers, and it would not feel right for one of them to have to become master of another in order to be able to communicate securely with it, especially given that the individual dust motes are neither self-propelled nor cyber-representatives of hypothetical humans that could physically help them perform the initial contact-based bootstrapping phase of imprinting.

2.3 *Mater semper certa...*

You always know who the mother is, the Romans used to say in their wisdom, but you can never be sure about the father, where there may be several candidates. In the Duckling model we don't care about the father at all, but we may have got somewhat carried away on the subject of the uniqueness of the mother.

OK, granted: after imprinting there is one and only one very special principal that the duckling will recognise as mother, and obey to the death; but do we really need to forbid the duckling from ever interacting with anybody else? In particular, would it not be possible for the duckling to accept orders (at least *some* kinds of orders) from other principals too? An affirmative answer to these questions leads the way to the announced extension of the Resurrecting Duckling model to peer-to-peer interaction.

There are two distinct ways of being master that we have so far confused and that we shall now distinguish. Firstly, you can be master because the slave is imprinted to you and will be faithful to you for all its life; this is a long-term relationship which might last for years. Secondly, you can be master on a much more temporary basis, just for the duration of a brief transaction: you ask your dining neighbour to pour you some wine and you assume the role of master for a moment, only to become slave later when it's your turn to pass on the vegetables. So far we implied that, in order for one principal to be master of another, the second principal had to be imprinted to the first. We now repudiate this view: the two devices can establish a very temporary master-slave relationship without either being imprinted to the other.

The imprinted duckling is indeed faithful for life to its unique mother duck; but it is happy to talk to others, and even obey their requests, as long as mummy said it was OK to do so.

The germ of this idea was already in our original paper [7], where we proposed

> "to always bootstrap by establishing a shared secret and to use strong cryptography to download more specific policies into the node. The mother can always send the duckling an access control list or whatever in a message protected by the shared secret."

But at the time we had not yet realised that the mother could also delegate her control over the duckling; in fact we said that

> "an imprinted duckling may still *interact* with principals other than its mother — it just cannot be *controlled* by them."

This limitation is unnecessary, so we now remove it. Of course the mother duck is still special, and she does still enjoy some extra control over her duckling, as we shall see.

Let's model the duckling as an object (in the OO sense) with a series of methods, i.e. actions that the duckling can perform on itself, possibly changing its own state. A *policy*[7] for the duckling shall be an arbitrarily complex statement specifying, for each of the available actions, which credentials the principal should exhibit in order to persuade the duckling to perform it. The policy can grant or deny any privileges it wants over the possible actions for the duckling; the only fixed rule, which is in some sense a bootstrapping base, is that if a principal can demonstrate knowledge of the imprinting key of a duckling, then it can upload a new policy into it.

Note that this implication is only one way: we see no reason to also dictate that one can only upload a new policy if one knows that imprinting key. As a matter of fact, for the duckling "downloading a new policy" (and even "committing suicide") are just two of the many possible actions: whether any given principal is allowed to invoke them is something that depends on the specific policy that currently resides in the duckling.

It is conceivable for the original mother duck to upload a policy that would allow other principals to upload a new policy or even kill the duckling. This may effectively be a functional alternative to backing up the imprinting key: it amounts to designating a "godfather" (godmother?) that may at any time take over the role of mother duck.

It should be clear that this power of delegation should be exercised with care, since the designated godmother(s) will be able to kick out the original

[7] We appear to be guilty of semantic overloading here, since we previously described the whole Resurrecting Duckling construction as a *security policy model*. We do in fact distinguish the two uses. A "security policy model" is a general security specification that gives overall guidelines for the behaviour of a certain class of systems: a typical example would be Bell-LaPadula [2]. Actual policies (sometimes referred to as "security targets") may be derived from it by specialisation to a particular application and implementation. The reason why we decided to reuse the word "policy" here is to emphasize that this is the same type of entity as those mentioned in trust management systems such as PolicyMaker [5] and KeyNote [4].

mother at will: anyone who can upload a new policy can also kill the duckling (by making that action possible for herself), then re-imprint it and ensure that the old mother is no longer recognised.

Without pursuing the matter in great detail, we hint at the fact that the above problem might be kept under control using a multilevel integrity system, à la Biba [3] — again something that we suggested in the original paper to address a slightly different issue. The various parts of the policy would be ranked at different integrity levels, so that one could allow the low integrity items to be rewritten but not the high integrity ones, which would include the most sensitive actions such as killing the duckling and, recursively, rewriting the high-level portions of the policy.

To sum up the important extension to our model, being mother duck allows one to perform the special action of uploading a new policy in the duckling; but, apart from that, any action can be invoked by any principal who presents the required credentials, as required by the duckling's then-current policy.

This enables peer-to-peer interaction. The remote control will give all the components of the hi-fi system the necessary credentials so that they can ask each other to perform the appropriate operations. The White general will be mother duck to all his dust motes (probably via a cyber-intermediary) and will give them the credentials that allow them to talk to each other — credentials that the dust motes from the Black army won't have, even if they come from the same manufacturer.

2.4 Further indirection issues

Interoperability *If you take me for a ride in your GPS-equipped car, where the GPS is imprinted to you, can my camera obtain the current geographical position from your equipment to stamp the pictures I take while you are driving? More generally, is a duckling limited to only talk to its siblings? If so, there would be no interoperability.*

The interoperability problem is solved by appropriate clauses in the policy.

Firstly, there may be innocuous actions (e.g. giving out the current position for a GPS unit) that a duckling is happy to perform for anyone[8]. This is obtained by not requiring any credentials for the initiators of such actions in the policy of the GPS duckling.

Secondly, my camera still must have some assurance that the positions given out by your GPS unit are trustworthy, otherwise anyone could fool it into stamping the pictures with bogus geographical coordinates. This is obtained by defining your GPS as a valid source of geographical coordinates in the policy of my camera duckling. At the implementation level this may be performed in many ways whose relative advantages will have to be assessed. For example the GPS might be given a "this device can be trusted to give out valid position information" certificate by some standards body, and the camera might recognise and

[8] But note the denial of service problem, such as the *sleep deprivation torture* introduced in the original Duckling paper.

accept this[9]. Alternatively, the grandmother duck of the camera might issue such a credential herself for that GPS ("I tell you, my son, that you can believe the coordinates sent to you by this specific GPS unit") and store it in the camera duckling.

Thirdly, there may even be cases where we *want* the duckling to be able to talk only to its siblings, as with the White vs. Black dust motes.

Control interface *If I go abroad and forget at home my PDA, which is mother duck to all my other gadgets, is it now impossible for me to control them until I get back?*

No. One should not make the mistake (induced by the primary example of the universal remote control) of identifying the mother duck with the control interface for the duckling. As soon as I buy a new gadget, I imprint it to my cyber-representative (which might well be my PDA for illustration purposes), but the policy I upload into it may specify that any other gadget of mine is allowed to control it, as long as it has a user interface that is suitable for issuing the appropriate commands. I may then use any available gadget for controlling any other, and I could conceivably imprint my MP3 player to my PDA but control it from my wristwatch. As a matter of fact I might even keep my cyber-representative in a safe and only ever take it out to imprint my other gadgets.

Tamper resistance *What happens if the Black general captures a White dust mote, dissects it à la Markus Kuhn [1] and steals its credentials? Can it now impersonate it with all the other White dust motes?*

Yes, unfortunately. If we decide to put credentials inside the ducklings, we must rely on the ducklings being tamper resistant to some extent. The original policy model already stated that breaking the tamper resistance ought to cost more than legitimately acquiring an imprintable duckling. We now add that it also ought to cost more than the value obtained by stealing the duckling's credentials.

The cost to the White general, as well as that of the direct loss of any valuable secrets, would have to include that of revoking those compromised credentials and replacing them by new ones in all the dust motes — a costly operation if they cannot all be easily contacted once deployed.

This in fact highlights a non-trivial conceptual problem: once we introduce delegation like we just did we also reintroduce, in its full glory, the problem of revocation in the absence of an online server. Since we make no a priori guarantees about the connectivity status of the duckling, there may be circumstances where not even the mother duck is contactable. From a theoretical point of view, this is probably just as bad as the original starting point. In practice the problem is somewhat mitigated by the fact that the authority issuing those credentials is now more decentralised.

[9] The validity of such a certificate is linked to the tamper resistance of the device, as we discussed in the original paper.

Trust management *How shall the duckling decide whether the credentials exhibited by another principal are sufficient to grant the principal permission to execute the requested action?*

This is a general problem for which, fortunately, a general solution has already been developed. Ducklings may embed a generic trust management engine such as KeyNote [4]. Policies and credentials shall be expressed in a common language and any duckling will be able to just feed its own policy, the external request and the supplied credentials to its engine which will return a boolean answer as to whether the requested action is allowed or not.

Policy specification *How will the owner of a device be able to specify a sensible policy for it? It looks as if doing this properly will be a job for a security expert.*

Writing a policy will indeed require competence in security and will be no less complicated than programming. End users will not be expected to write their own policies; instead, devices will come with a portfolio of "sensible policies" (hopefully with explanations), that the user will be able to parameterise. Power users will be able to write their own policies if they wish, or edit the supplied ones, and probably web sites will appear that archive the best of those homebrew variants.

Family feelings *Wouldn't it be possible to exploit the fraternal love among sibling ducklings as an additional security feature?*

Sure, neat idea! The policy for the ducklings in your home might say that they should stop working when they feel lonely, because in normal operation it is reasonable for them to expect that they will be surrounded by at least n siblings. This is a case in which we make explicit use of the *short range* of our wireless communications, inferring proximity from connectivity. If they are not in range of their siblings, it may be because they were stolen, so they should refuse to work. (Of course this heuristic fails if the thieves steal the whole lot...)

3 Conclusions

The Resurrecting Duckling security policy model regulates secure transient association between devices in an ad hoc wireless network where authentication servers may not be available. In this paper we have extended this model from a strict master-slave situation to a more general case that includes peer-to-peer relationships.

Now the mother duck defines a lower level *policy* for her duckling on imprinting; through this policy, the power to control the duckling can be delegated to any other principal. The important conceptual step is to distinguish the long-lived master-slave relationship of imprinting from the temporary master-slave relationship of asking the duckling to perform one action.

The versatility of the extended model covers a wide range of new uses. We think we have addressed most of the practical scenarios in ad hoc wireless net-

working, but as this work is still in progress we shall gratefully receive any criticisms about exceptions and omissions.

4 Acknowledgements

Over the past year I have talked about ducklings to maybe ten different audiences on both sides of the Atlantic and I am indebted to all of them for their stimulating and often very sharp comments which have influenced and fuelled my progress. Space prevents me from acknowledging all the worthy suggestions individually, but I especially thank Jonathan Smith from the University of Pennsylvania and my colleague James Scott from the University of Cambridge for first pointing out cases in which the master-slave relationship was inadequate. Randy Katz's mind-blowing talk on smart dust was very inspirational to me, even though (or perhaps precisely because) it offered no solutions to the security problems. Many colleagues at AT&T Florham Park and Newman Springs, including at least Matt Blaze, Ed Chen, Paul Henry and Ben Lee, provided constructive criticism and encouragement. Ross Anderson, who co-authored the original Duckling paper, was always available for comments and fruitful discussions. Finally, the audience of this Cambridge Security Protocols workshop was very responsive; in particular Virgil Gligor, Markus Kuhn and Pekka Nikander offered interesting insights, some of which have been incorporated in this revision.

These grateful thanks should not however be mistaken as claims of endorsement, and I remain of course fully responsible for defending any ideas and opinions herein expressed.

References

1. Ross Anderson and Markus Kuhn. "Tamper Resistance—A Cautionary Note". In "Proc. 2^{nd} USENIX Workshop on Electronic Commerce", 1996. ISBN 1-880446-83-9. http://www.cl.cam.ac.uk/~mgk25/tamper.pdf.
2. D. Elliot Bell and Leonard J. LaPadula. "Secure Computer Systems: Mathematical Foundations". Mitre Report ESD-TR-73-278 (Vol. I–III), Mitre Corporation, Bedford, MA, Apr 1974.
3. Kenneth J. Biba. "Integrity Considerations for Secure Computer Systems". Tech. Rep. MTR-3153, MITRE Corporation, Apr 1975.
4. Matt Blaze, Joan Feigenbaum, John Ioannidis and Angelos D. Keromytis. "The KeyNote Trust-Management System". RFC 2704, Network Working Group, Sep 1999. http://www.crypto.com/papers/rfc2704.txt.
5. Matt Blaze, Joan Feigenbaum and Jack Lacy. "Decentralized Trust Management". In "Proceedings of the 17^{th} IEEE Symp. on Security and Privacy", pp. 164–173. IEEE Computer Society, 1996. ftp://ftp.research.att.com/dist/mab/policymaker.ps.
6. Joe M. Kahn, Randy H. Katz and Kris S. J. Pister. "Next Century Challenges: Mobile Networking for "Smart Dust"". In "Proceedings of International Conference on Mobile Computing and Networking (MobiCom 99)", Seattle, WA, USA, Aug 1999. http://robotics.eecs.berkeley.edu/~pister/publications/1999/mobicom_99.pdf.

7. Frank Stajano and Ross Anderson. "The Resurrecting Duckling: Security Issues in Ad-Hoc Wireless Networks". In Bruce Christianson, Bruno Crispo and Mike Roe (eds.), "Security Protocols, 7th International Workshop Proceedings", Lecture Notes in Computer Science. Springer-Verlag, 1999. http://www.cl.cam.ac.uk/~fms27/duckling/. See also [8]. Also available as AT&T Laboratories Cambridge Technical Report 1999.2.
8. Frank Stajano and Ross Anderson. "The Resurrecting Duckling: Security Issues in Ad-Hoc Wireless Networks". In "Proceedings of 3^{rd} AT&T Software Symposium", Middletown, New Jersey, USA, Oct 1999. http://www.cl.cam.ac.uk/~fms27/duckling/. Abridged and revised version of [7]. Also available as AT&T Laboratories Cambridge Technical Report 1999.2b.