Improving Onion Notation

Richard Clayton

University of Cambridge, Computer Laboratory, William Gates Building,
15 JJ Thompson Avenue, Cambridge CB3 0FD, United Kingdom
richard.clayton@cl.cam.ac.uk

Abstract. Several different notations are used in the literature of MIX networks to describe the nested encrypted structures now widely known as "onions". The shortcomings of these notations are described and a new notation is proposed, that as well as having some advantages from a typographical point of view, is also far clearer to read and to reason about. The proposed notation generated a lively debate at the PET2003 workshop and the various views, and alternative proposals, are reported upon. The workshop participants did not reach any consensus on improving onion notation, but there is now a heightened awareness of the problems that can arise with existing representations.

1 Introduction

Chaum's original paper on MIX systems [5] used a functional notation for expressing encryption. For example, a user's message T which is to be delivered to the destination A would be encrypted using A's public key K_a and the address A would be appended. This would all then be encrypted (the original paper called this "sealed") with the public key K_1 of MIX machine M_1. With the addition of random nonces R_0 and R_1 this would be expressed as a message for M_1:

$$K_1(R_1, K_a(R_0, T), A)$$

This message (along with the destination address M_1 and another nonce R_2) could be further encrypted with the public key K_2 of MIX M_2 and so on to involve as many MIX systems as might be required.

These nested structures are now widely known as "onions", a term first brought to prominence by Goldschlag et al [9]. A related notion of layered encryption was called a "telescope" by Zero Knowledge [8].

Nested encryption has been widely discussed, but Chaum's original notation is not always used, perhaps because it was so closely associated with the use of public key encryption, rather than a generic sealing operation. Other authors, such as Ohkubo and Abe [11] use an explicit notation for the encryption function, such as $\mathcal{E}$. In this notation, encryption with a key K_a becomes $\mathcal{E}_{K_a}$ and our example becomes:

$$\mathcal{E}_{K_1}(R_1, \mathcal{E}_{K_a}(R_0, T), A)$$

Another group of authors, such as Serjantov [13], have used a bracketed subscript notation, similar to BAN logic [3] and later used in the Spi calculus [1].

In this notation the encryption of message T under key K is expressed as $\{T\}_K$. Therefore the onion example becomes:

$$\{R_1, \{R_0, T\}_{K_a}, A\}_{K_1}$$

A plethora of different notations is perhaps inevitable because authors use the ones with which they are familiar. However, it is possible to set out some objective criteria for assessing notation. Designing a new notation that better meets these criteria does of course add to the existing babel, but if it is sufficient improvement then this may be acceptable.

2 Assessing Notations

The history of mathematical notation [4] is littered with notations that have failed to gain wide acceptance, or that have been swept away by later, simpler, schemes. In some cases such as Frege's *Begriffsschrift* [6] they have actively prevented contemporaries from understanding the scientific contribution. [10] A good notation will be simple to read, write and typeset. In particular:

It will fit onto a single line: Notation that uses vertical layout to convey information has seldom survived when alternatives have been developed that have the same height as normal text. Frege's notation was especially weak in this respect.

It will be easy to write: Onions are regularly written on whiteboards, in typographical systems such as LaTeX and in presentation systems such as PowerPoint. The notation should produce similar looking results no matter what medium is used. In particular, specification documents such as the RFCs produced by the Internet Engineering Task Force (IETF) are constrained to use ASCII characters and this is extremely restrictive.

It will be easy to read: Familiarity will make even the most exotic notation appear normal. However, practical issues relating to poor eyesight must be kept in mind, and multiple levels of subscript or superscript should be avoided because of the size reduction that is conventionally applied.

It will be easy to comprehend: The use of deeply nested brackets needs to be avoided. If the levels of nesting need to be joined up in pencil before they can be understood, then the notation is preventing understanding rather than enhancing it.

It will allow simple generalisation: All of the notations currently in use express onions of arbitrary complexity by means of ellipsis (...) notation. However, where more than one ellipsis is needed then it can become unclear how deeply nested constructs are built. For example:

$$K_n(R_n, K_{n-1}(...K_2(R_2, K_1(R_1, K_a(R_0, T), A), M_1), ...), M_{n-1})$$

may present some difficulties in understanding exactly how to match up the left and right hand ends of the expression.

It will allow errors to be easily detected: Related parts of the onion should be presented together whenever possible. If keys and addresses are at opposite ends of an expression then it is hard to spot a mistake such as:

$$K_3(R_3, K_2(R_2, K_1(R_1, K_a(R_0, T), A), M_3), M_2)$$

where the address M_3 should be M_1.

Etcetera: It is possible to go on to identify many other issues worthy of consideration. For example, the Cognitive Dimensions of Notations framework [2] lists a total of 13 dimensions that should be considered, especially when the whole life-cycle of the notation is considered and not just its appearance in academic papers and conference presentations.

3 Proposed Onion Notation

Clearly, from the list of requirements, an improved notation for onions will fit on a single line, will avoid multiple levels of subscripts, and will not nest brackets. An entirely straightforward scheme for achieving this would be to write down the onion in the temporal order in which it is constructed:

$$|R_0, T\#K_a|R_1, A\#K_1$$

where one reads | as starting a section and # as meaning "use the following key to encode the preceding section".

This notation may be adequate for theoreticians who are seldom concerned about the exact order of the concatenated components within a particular onion layer. However to satisfy those who consider this ordering to be important we need to add a symbol to express "the result of the last encryption operation". Here, in a typeset paper the LaTeX `\star` symbol $\star$ is used for this purpose, though substituting an asterisk ($*$) would not cause confusion in other contexts. Our onion expression for the packet sent to MIX M_1 now becomes:

$$|R_0, T\#K_a|R_1, \star, A\#K_1$$

which we should read as being a packet encrypted under the public key K_1 that contains the nonce R_1, then the packet encrypted under the key K_a and then the address of A; where the packet for A contains the nonce R_0 and the user's message T.

3.1 Advantages of the $\star$ Notation

Some advantages of this notation will be immediately apparent. One can generalise it to an onion of arbitrary complexity without any need to count brackets, match up MIX identifiers with keys or any of the other tedious proof-reading that traditional notations require. For example it is far easier to construct and check the generalised onion:

$$|R_0, T\#K_a|R_1, \star, A\#K_1|R_2, \star, M_1\#K_2|...\#K_{n-1}|R_n, \star, M_{n-1}\#K_n$$

than the traditional:

$$K_n(R_n, K_{n-1}(...K_2(R_2, K_1(R_1, K_a(R_0, T), A), M_1), ...), M_{n-1})$$

The small advantage this gives the author pales into insignificance when compared to the significant advantage that it gives to every future reader.

A more subtle advantage is that it is now possible to reason about the various MIXs directly from the notation. Assuming reasonable security precautions have been taken, only MIX M_i will be able to decrypt material encoded under its public key K_i. Hence it is immediately apparent that it will only have access to that part of the onion which lies in the section to the left of any $\#K_i$ within the onion. In the running example this will mean that it can only "see" the value $R_i, \star, M_{i-1}$ and from that, it will be possible to derive (informal) correctness results.

3.2 A Further Complication

One further piece of notation is needed, to cover onions that are encrypted to more than one key per MIX, such as in Pfitzmann & Waidner's 1986 proposal [12] for avoiding end-to-end retransmissions in MIX networks. Using the traditional notation of the original paper (but labelling consistently with this one), the encrypted message X_i that is delivered to MIX M_i is recursively defined as:

$$\begin{aligned} X_a &: K_a(T) \\ X_n &: K_n(k_n, A), k_n(X_a) \\ X_i &: K_i(k_i, M_{i+1}, k_{i+1}, M_{i+2}), k_i(X_{i+1}) \end{aligned}$$

where key k_i is a second key known to MIX M_i in addition to the usual K_i value.

To handle this case, the new notation needs one further piece of syntax. The $\star$ values must be enumerated to indicate which of the preceding encrypted results is to be used. We define $\star_0$ to be the same as $\star$, i.e. the result of the immediately preceding encryption operation. $\star_1$ is then defined to be the result of the encryption operation before that (etc as needed). With this extension the scheme becomes:

$$\begin{aligned} X_a &: |T\#K_a \\ X_n &: |X_a\#k_n|k_n, A\#K_n| \star_0 \star_1 \\ X_i &: |X_{i+1}\#k_i|k_i, M_{i+1}, k_{i+1}, M_{i+2}\#K_i| \star_0 \star_1 \end{aligned}$$

Expanding the onion by even just a single level demonstrates the advantage of the new notation. X_i could also have been expressed as:

$$K_i(k_i, M_{i+1}, k_{i+1}, M_{i+2}), k_i(K_{i+1}(k_{i+1}, M_{i+2}, k_{i+2}, M_{i+3}), k_{i+1}(X_{i+2}))$$

whereas in the new notation it becomes:

$$|X_{i+2}\#k_{i+1}|k_{i+1}, M_{i+2}, k_{i+2}, M_{i+3}\#K_{i+1}|\star_0\star_1\#k_i|k_i, M_{i+1}, k_{i+1}, M_{i+2}\#K_i|\star_0\star_1$$

which is rather simpler to parse. In fact, it is even quite straightforward to avoid the recursive definition entirely, which although elegant, takes a little while to understand. Using the $\star$ notation the full onion can be written out without the risk of alienating the reader, something the original authors wisely eschewed:

$$\begin{aligned}
&| T\#k_a \\
&| k_n, A\#K_n| \star_0 \star_1\#k_{n-1} \\
&| k_{n-1}, M_n, k_n, A\#K_{n-1}| \star_0 \star_1\#k_{n-2} \\
&| k_{n-2}, M_{n-1}, k_{n-1}, M_n\#K_{n-2}| \star_0 \star_1\#k_{n-3} \\
&| k_{n-3}, M_{n-2}, k_{n-2}, M_{n-1}\#K_{n-3}| \star_0 \star_1\#k_{n-4} \\
&| k_{n-4}, M_{n-3}, k_{n-3}, M_{n-2}\#K_{n-4}| \star_0 \star_1\#k_{n-5}...
\end{aligned}$$

From this we can, by simple inspection, directly reason which other keys the possessor of K_i will have access to (i.e.: k_i and k_{i+1}). However, possession of these keys will only allow access to material encrypted under K_{i+1}, K_{i+2} or K_{i+3}, and hence the system security properties hold.

With the improved notation, it will still be possible to do this type of analysis even with the presence of otherwise distracting details. These would include the use of nonces to prevent output message identification and the addition of padding to prevent traffic analysis attacks based on the size of message headers.

4 Workshop Discussion

The presentation of the paper at the workshop was followed by a short discussion:

Adam Shostack: On your presentation slides you showed how the brackets could be linked together by different levels of underlining. I think that was very visual and appropriate usage of underlining and overlining would be an elegant way to portray exactly which elements are encrypted with particular keys.

Sandra Steinbrecher: Your notation seems to be created to assist implementers, but many researchers will be coming from a mathematical background and will cope with the existing notations.

Rachel Greenstadt: It's important to be able to come from related fields such as crypto or security and read our papers. Everyone will be used to brackets and so your notation must have brackets.

Richard Clayton: It is possible to dispose of the # notation and use brackets for the action of the encryption keys. The $\star$ notation remains the same so as to avoid all the nesting. i.e.:

$$K_a(R_0, T)|K_1(R_1, \star, A)|K_2(R_2, \star, M_1)$$

James Alexander: Perhaps language theory has something to contribute. In that field the representation and the notation of the underlying model will differ. There should be no problem with notations for different purposes if there is a clear mapping between them, perhaps using a functional notation.

Dogan Kesdogan: Adding yet another notation might not be entirely helpful.

Christian Grothoff: I'd suggest that a functional notation might meet the need for clarity without the hardships caused by the introduction of new syntactic elements. For example, defining a function and then using a standard notation for composition would look like this:

Definition:	$\mathcal{E}(a, b)(x) := K_a(R_a, x, b)$
Encryption:	$\big(\mathcal{E}(M_3, M_2) \circ \mathcal{E}(M_2, M_1) \circ \mathcal{E}(M_1, A)\big)\ (T)$

Andrei Serjantov: I made a specific point of using the notation from the community in which my papers were originally introduced.

Paul Syverson: I think the key point is that there is not just one community working on onions. People are coming at them from theory, from PETs and from the crypto community. Your notation may assist the people who are writing code, but it may not be very helpful to those who just want to analyse the validity of their schemes, so I am not sure that there is as big a win here as you are suggesting.

5 Conclusions

A notation for expressing the construction of onions has been presented that is as concise as existing notations. By mirroring the temporal construction of the onions the reader avoids having to count brackets backwards and forwards to establish the author's intent. The notation is also capable of dealing with complex cases with encryption with multiple keys.

However, the notation was not widely welcomed by the workshop, although people were prepared to accept that existing papers sometimes manage to obscure the mechanisms they present by poor use of notation. There seems to be a role for programme committees in ensuring some consistency of notation for accepted papers to prevent unnecessary divergence from established practice.

Finally, when pondering the utility of change, one might note the observation of Leibniz [7] (who was one of the most successful innovators of mathematical notation, several of his inventions having survived to the present day):

> *"In signs one observes an advantage in discovery which is greatest when they express the exact nature of a thing briefly and, as it were, picture it; then indeed the labour of thought is wonderfully diminished."*

Acknowledgements

I would like to thank George Danezis and Andrei Serjantov for their initial help in ensuring I did not produce an unacceptably obscure notation. George also took notes of the workshop discussion. I'd also like to thank the anonymous referees for drawing my attention to Frege's notation-induced failure and for making me face up to the significant problems that brackets were causing, so that I finally stumbled upon the simplicity that "|" could provide.

References

1. M. Abadi and A. D. Gordon: A calculus for cryptographic protocols: The spi calculus. Information and Computation, 148(1), 1999, pp. 1–70.
2. A. Blackwell and T. Green: Notational Systems - the Cognitive Dimensions of Notations framework. In J. M. Carroll (Ed.): HCI Models, Theories, and Frameworks: Towards a Multidisciplinary Science. Morgan Kaufmann Publishers, 2003.
3. M. Burrows, M. Abadi and R. Needham: A Logic of Authentication. ACM Trans. on Computer Systems 8(1), 1990, pp. 18–36.
4. F. Cajori: A history of mathematical notations. The Open Court Publishing Company, Chicago Il. 1928–9.
5. D. Chaum: Untraceable Electronic Mail, Return Addresses, and Digital Pseudonyms. Comm. ACM 24(2), 1981, pp. 84–88.
6. G. Frege: Begriffsschrift. Eine der arithmetischen nachgebildeten Formelsprache des reinen Denkens. Halle, 1879.
7. C. I. Gerhardt: Briefwechsel von G.W.Liebniz mit Mathematikern, Vol I. quoted in para. #540 of [4].
8. I. Goldberg and A. Shostack: Freedom Network 1.0 Architecture, October 10 2001. `http://www.homeport.org/~adam/zeroknowledgewhitepapers/arch-notech.pdf`
9. D. M. Goldschlag, M. G. Reed, and P. F. Syverson: Hiding Routing Information. In R. Anderson (Ed.): Information Hiding, LNCS 1174, Springer Verlag 1996, pp. 137–150.
10. R. Mendelsohn: Begriffsschrift in Modern Notation. `http://comet.lehman.cuny.edu/mendel/papers/Adobe%20Versions/AdobeNewBGForms.pdf`
11. M. Ohkubo and M. Abe: A Length-Invariant Hybrid Mix. In T. Okamoto (Ed.): ASIACRYPT 2000, LNCS 1976, Springer Verlag 2000, pp. 178–191.
12. A. Pfitzmann and M. Waidner: Networks without User Observability. Computer Security 6(2), 1987, pp. 158-166.
13. A. Serjantov: Anonymizing censorship Resistant Systems In P.Druschel et al (Ed.): First International Workshop, IPTPS 2002, Cambridge Ma. USA, March 2002, LNCS 2429, Springer Verlag 2002, pp. 111-120.