Agenda Items (and Priorities)
for the Design Team Assigned to Resolving
Working Group Last Call Discussion on NTS Drafts

Note about priority notation: The top priority [3/3] is reserved for issues which are both highly significant and very urgent. The non-priority [0/3] is reserved for items which are estimated to not require any further effort at all and should at most require the team to agree on this estimation. The other options [1/3] and [2/3] are nuances which for now have been used very subjectively.

Also see: https://trac.tools.ietf.org/wg/ntp/trac/wiki/NtsWgIcDesignTeam

1. **IP fragmentation of certificate-carrying messages during key exchange**
    - *[3/3] Perceived Priority:* Very urgent! Decision on course of action needs to be made very soon, because making substantial changes would require immediate action.
    - *Comment/Explanation:* The proposed initial key exchange protocol includes messages carrying certificate chains. If the length of said chain exceeds 1 (and possibly also if it doesn't), the size of the message will likely exceed the maximum transmission unit (MTU) above which packets are fragmented on the IP level. Externally-imposed IP fragmentation has been described as very detrimental, mainly because of two reasons: security flaws and compatibility issues with middleboxes.
    - *Discussion Points:*
        - What are the concrete security risks with IP fragmentation?
        - What is the disturbance level (and/or probability) of IP fragmentation in the presence of middleboxes?
        - How is the significance of security risks and compatibility issues weighed?
        - Is the "MUST" requirement for the self-defined KE protocol sensible?
            1. Perceived agreement: there should be a "MUST" requirement for something.
        - What are the concrete requirements for the key exchange mechanism?
            1. Simplicity (specification): The solution would ideally be easy to write down and require little specification text.
            2. Simplicity (implementation): The solution should be easy to implement.
            3. Reusability, the protocol should use existing mechanisms to make it easy for implementation and avoid NTS specific security issues
            4. Compatibility with Internet, avoid IP fragmentation
            5. Compatibility with existing firewall configurations, keep it on UDP port 123

6. No per-client state on server to avoid DoS attacks
7. Control of individual packet's transmission to spread packets and avoid congestion on server
8. Light on network/cpu resources to handle large number of clients exchanging keys at the same time, small packets, small number of crypto operations

- Here are some solutions and how they might satisfy the requirements:

		1	2	3	4	5	6	7	8
B2)	Current draft modified (split for PMTU)	-	?	-	X	X	-	?	?
~~**C)**~~	~~TLS~~	~~X~~	~~X~~	~~X~~	~~X~~	~~-~~	~~-~~	~~-~~	~~?~~
D1)	DTLS on separate port	X	X	X	X	-	-	?	?
E)	DTLS over NTP	?	?	X	X	X	-	X	?

- *Available Options:*
  - Leave NTS documents unchanged/Ignore completely. Leave specification and implementation unchanged, let IP fragmentation occur. See if this will cause problems.
    1. Advocates: Harlan, Richard
    2. Opposed: Sharon, Hal
  - Introduce self-management. Add text for managing fragmentation on an NTS or NTS-4-NTP level. Would align with the way that DTLS handles this problem.
    1. Proposed by: Sharon
    2. Disadvantages: Rate limitation becomes a way more significant problem
  - Declare certificate exchange as out-of-scope. Add language to NTS-4-NTP or perhaps NTS draft stating that certificate chains are assumed to already be exchanged initially.
    1. Mentioned as an easy Solution by: Dieter
    2. Agreement
  - Try outsourcing. Look for a channel/protocol which can handle the problem for us. Modify text in NTS-4-NTP appropriately.
    1. Proposed by: Kristof
    2. TCP might be one option.
       a. Danny has concerns about introducing TCP additionally
    3. Harlan thinks it's a "choose your poison" situation
  - One specific point of the "outsourcing" approach above would be the use of DTLS for the communication up to and including the cookie exchange. This has been discussed quite a lot recently (and Eric Rescorla has been

involved, offering some amount of guidance).
This option involves several sub-options:

1. Use DTLS over extension fields (option E).
    a. Would work without any need for network administrators to change any of their existing regulations, but:
    b. Requires state over potentially multiple NTP packet exchanges (until the DTLS message can be re-assembled).
    c. Requires interoperation of the NTP and DTLS implementations for reasons of communicating at least the "effective MTU" (data size available to DTLS in the extension field of a given NTP packet).
2. Use DTLS packets in parallel with NTP packets, both sent over UDP on port 123 (option D2).
    a. Would work without any need for network administrators to change any of their existing regulations, but:
    b. Does not circumvent any existing rate limitations for UDP 123 packets.
    c. Requires multiplexing and de-multiplexing.
3. Use DTLS packets on a separate channel (probably sent over UDP, on a port different from 123), (option D1).
    a. Circumvents rate limitations for UDP 123 packets, but:
    b. Requires a separate port to be available.

2. Key exchange protocol: fewer exchanges?

- *[3/3] Perceived Priority:* Very urgent! Decision on course of action needs to be made very soon, because making substantial changes would require immediate action.
- *Comment/Explanation:* The proposed initial key exchange protocol currently features three exchanges: "access", "association", and "cookie". It is possible and might be desirable to condense association and cookie into one exchange.
- *Available Options:*
    - Leave NTS documents unchanged. Keep all three exchanges the way they are, do not add another one.
        1. Pro Arguments: No effort; Evades potential problems around cookie exchange.
        2. Contra Arguments: Preserves potentially unnecessary computational and network load once per association.
    - Replace the two old exchanges with one new exchange in all documents.
        1. Pro Arguments: Eliminates complexity; Reduces effort of initial KE
        2. Contra Arguments: Increases effort when server seed is refreshed
    - Introduce one new combined exchange in addition to the two old exchanges.

1. Proposed by: Dieter?

- *Tied to:* IP fragmentation of certificate-carrying messages during key exchange.
  - If the “self-management” option is taken there, it would make a lot of sense to take the “replace two old exchanges with one new” option here.
  - If the “external protocol/outsourcing” option is taken there, the exchanges will possibly be heavily modified.

3. **Key exchange protocol: fewer cryptographic operations?**
   - *[2/3] Perceived Priority:* Needs to be discussed very soon, because potential changes would need to be made shortly.
   - *Comment/Explanation:* Sharon and her team suggested that even apart from reducing the number of necessary exchanges, the key exchange protocol could be performed with fewer cryptographic operations.
   - *Discussion Points:*
     - Preliminary discussion. Sharon mentioned she would need some explanations on design decisions etc. in order to flesh out any suggestions.
     - How would something like this work/what options are there?
   - *Available Options:*
     - Leave NTS documents unchanged.
     - ? (Depends on discussion points) in

4. **Key exchange protocol: two-way authentication?**
   - *[2/3] Perceived Priority:*
     This
   - Discussion Points:
     - Is having it even worth it?
       1. Is symmetric mode the main/only reason to support 2-way auth.?
          a. If so, how big are the symmetric mode meshes? If the meshes only have 2-3 peers, we may not need to scale this and preshared keys might work fine. But if they are big (hundreds) then scaling with public keys/certs might be needed.
          b. Miroslav says that mutual authentication is only needed when there is a symmetric “passive” association; if both sides of the peering exchange are symmetric active peers, then mutual authentication is not needed. (Sharon: I would like a bit more clarification about why this is the case, it would be nice if Miroslav could send more details to the list.)
       2. Do control queries need to be mutually authenticated?
       3. Is there a use case where BOTH mutual authentication/authorization AND good scaling are required?

        a. Could Peer Mode be such a use case?
    - Does NTS really scale better in use cases which require mutual authentication?
    - Would (D)TLS even support two-way authentication? If so, how?
        1. It looks like TLS does have optional mutual authentication, but I don't know much about it.
    - Miroslav: Peering association where both participants are active might be different from those where one side is passive.
- Available Options:
    - Leave NTS documents unchanged.
    - Eliminate Client-to-Server authentication/authorization.
    - Use symmetric keys to authenticate symmetric mode and passwords to authenticate control queries.

5. Discussion about Chicken-and-Egg Problem (Decided, To Be Written)

- ~~*[2/3] Perceived Priority:* The degree of treatment in the NTS documents should be decided soon. The matter is complex enough for a detailed treatment to take quite long.~~
- ~~*Comment/Explanation:* The Chicken-and-Egg problem references the fact that on the one hand, all common cryptographic primitives demand some degree of commonly agreed to time (the required degree varies between different primitives), and on the other hand, reliable (secure) digital time synchronization techniques require the use of such cryptographic primitives.~~
- ~~*Available Options:*~~
    - ~~Leave NTS documents unchanged. In this case the documents would barely make any mention of the topic apart from it existing but being out of scope.~~
    - ~~Give an in-depth discussion on the matter in one of the NTS documents.~~
        1. ~~Pro Arguments: This would make sure that the discussion (with contributions by authors with some good expertise) is written down in a publicly accessible place.~~
        2. ~~Contra Arguments: The issue is important in a more general context than NTS-4-NTP or even NTS. Moreover, it is highly complex and an in-depth treatment would require a significant amount of effort. Including it in the NTS documents would seriously delay their completion.~~
    - Supply some discussion on the matter and specifically write down the assumptions about initial trust that need to be fulfilled in order for NTS to work properly. Ideally, an in-depth discussion would be encouraged or even tasked in another document.
        1. ~~Pro Arguments: This would clear out those aspects that are directly relevant for NTS, therefore the issue would not be neglected and the NTS documents would be self-contained. If an~~

~~in-depth discussion of the more general problem is written down somewhere else, this would be a better overall treatment of the matter.~~

2. ~~Contra Arguments: This would not make certain that a well-written in-depth discussion is documented and available. It would also probably be impossible to link to the in-depth discussion from the NTS documents (due to temporal constraints: NTS documents would hopefully be completed before the potential additional document).~~

6. **Cipher Suites**
    - *[2/3] Perceived Priority:* This issue needs to be addressed, but not necessarily as soon as other issues on this list (would not have as significant secondary effects).
    - *Comment/Explanation:* Aanchal Malhotra pointed out that the current language (in the NTS-4-NTP document) used for regulating which cryptographic algorithms are to be used is problematic ("X or weaker MUST NOT be supported", "Y or stronger MUST be supported"). She advised treating the question in more detail, e.g. in a table considering all possible algorithms. This issue also begs the question if there should be a similar treatment of encryption/signature algorithms (as currently, the section in question deals only with MAC algorithms).
    - *Available Options:*
        - Leave NTS documents unchanged. This would leave very problematic language in place: not only does the current language assume a defined ordering of strength for crypto algorithms, it is also logically flawed.
        - Try to find an external document which can be quoted on the matter of which algorithms to employ and which to deprecate. The IETF's CFRG working group might be a good candidate for a place to find such a document.
        - Give a listing of all possible (MAC) algorithms, attach one of the options "MUST NOT", "MAY", or "MUST" (be supported) to each of them. Declare how to treat algorithms that are not on the list.

7. **Peer Mode**
    - *[2/3] Perceived Priority:*
    - *Comment/Explanation:* The peer mode has not been fully understood and it is not clear that the current treatment of it is suitable.
    - *Available Options:*
        - Leave NTS documents unchanged.
        - Include text warning that NTS-4-NTP might not protect the peer mode as desired due to lack of clear language/understanding.
        - Acquire the necessary expertise to assert that the treatment of the peer mode is sufficient and functional, possibly make further required changes.

        - Remove all text on how NTS should be used to secure peer mode, replace it by a statement saying that the use of “classic” symmetric key protection is recommended.

8. **Symmetry of Message Sizes “time_request” and “time_response”**
    - *[1/3] Perceived Priority:* Needs to be discussed, but seems non-urgent because it would be easy to change the specification accordingly and the changes would have very few secondary effects.
    - *Comment/Explanation:* Someone on the list (Miroslav?) suggested that it might be a good idea to have NTP packets carrying “time_request” and “time_response” messages to be of the same lengths. This would be done in order to achieve more symmetric computation delays (under certain assumptions).
    - Discussion Points:
        - How likely is it that this will ever have a positive impact (situation would have to be symmetric in terms of computational capacity etc.)?
        - What would be possible detriments to forcing the messages to have same bit lengths/same contents?
        - Would this be better located in an NTP scope than an NTS scope?
    - *Available Options:*
        - Leave NTS documents unchanged. This would effectively ignore the issues overall.
        - Include text stating that the extension fields for “time_request” and “time_response” should be padded to be of the same size.
        - Change text/message specifications so that extension fields for “time-request” and “time_response” include the exact same data types (with identical lengths).

9. **Use of Initial (Unsecured) Timestamps**
    - *[1/3] Perceived Priority:*
    - *Comment/Explanation:*
    - *Available Options:*
        - Leave NTS documents unchanged. In this case the documents do not even mention the topic at all.
        - Provide some text about the matter of using unsecured timestamps in general. Mention that backchecking plausibility after security has been enabled makes things a little better.
        - Give detailed guidelines for the validation and use of unsecured timestamps after secured ones have been exchanged and are ready for comparison.

10. **Seed Refresh: Should this Be Mentioned**

- *[1/3] Perceived Priority:* Not urgent because easy to eliminate from the text?
- *Comment/Explanation:* Sharon (and Florian Weimer before her) put up the question whether the mention of the seed refresh in its current form makes sense.
- *Discussion Points:*
  - What exactly would be gained by not describing a seed refresh?
- *Available Options:*
  - Leave NTS documents unchanged. This establishes the seed refresh as a distinct regular procedure that all participants know appropriate behavior for.
  - Discard all text about seed refresh and replace it by text about server restart (or something similar).
  - Eliminate all text concerning seed refresh without replacement.

11. Discussion about Different Security Approaches

- *[1/3] Perceived Priority:* Might not be necessary to treat this in the NTS documents. Decision about where/when to treat it should however be made soon.
- *Comment/Explanation:* Someone suggested that there should be a treatment of the advantages and disadvantages of the multiple different security approaches for NTP (namely NTS, classic symmetric approach, Autokey, IPSec, DTLS, …). Ideally, this treatment should include advice on when to use which approach.
- *Available Options:*
  - Leave NTS documents unchanged. This does not prohibit writing this discussion down at a later time in another document.
  - Include the discussion in one or more of the NTS documents. This would most likely be NTS-4-NTP, because the discussion is probably more fruitful when a specific time synchronization protocol is assumed.

12. MAC-Algorithm instead of Hash (for HMAC) Algorithm (Decided & Done)

- ~~*[0/3] Perceived Priority:* Already treated in the NTS documents (as a result of the feedback given in the WGLC), hopefully no further changes necessary for a satisfactory treatment..~~
- ~~*Comment/Explanation:* Jim Schaad suggested that it would make more sense to have identifiers for MAC algorithms as a whole instead of for a hash algorithm that would then be mandatorily used for use in an HMAC function.~~
- ~~*Available Options:*~~
  - Leave NTS documents unchanged. Was already treated in the submission for IETF 95.
  - ~~Make additional changes. Would require further feedback about the current treatment.~~

13. Make the Requirement about Use of Unauthenticated Time into "MUST NOT"

- Sharon suggests to change the language to “Client MUST NOT use unauthenticated time” (or similar).