全球定位系统
GNSS World of China
• 中国科学引文数据库（CSCD）
• 中文科技期刊数据库
• 中国核心期刊（遴选）数据库
• 日本科学技术振兴机构数据库（JST）
• 中国学术期刊（网络版）（CNKI）
• 中国学术期刊综合评价数据库（CAJCED）
• 中国超星期刊域出版平台

基于消息摘要加密的网络时间协议安全时间同步方法研究

陈　曦，臧文驰，马　明，龚　航，孙广富

Research on secure NTP method based on message digest encryption

CHEN Xi, ZANG Wenchi, MA Ming, GONG Hang, and SUN Guangfu

引用本文:
陈曦，臧文驰，马明，等. 基于消息摘要加密的网络时间协议安全时间同步方法研究[J]. 全球定位系统，2021，46(5)：84-91. DOI: 10.12265/j.gnss.2021072701
CHEN Xi, ZANG Wenchi, MA Ming, et al. Research on secure NTP method based on message digest encryption[J]. Gnss World of China, 2021, 46(5): 84-91. DOI: 10.12265/j.gnss.2021072701

在线阅读 View online: https://doi.org/10.12265/j.gnss.2021072701

DOI: 10.12265/j.gnss.2021072701

基于消息摘要加密的网络时间协议安全时间同步方法研究

陈曦, 臧文驰, 马明, 龚航, 孙广富
(国防科技大学 电子科学学院, 长沙 410073)

摘 要: 目前, 以网络时间协议 (NTP) 为主要的时间协议应用于有线网络中的时间同步, 其在广域网中可以实现十几毫秒、局域网中实现几毫秒的同步精度. 然而, 由于协议的开放性, 其在无安全防护的情况下极易受到网络攻击, 这给需要高安全的客户带来潜在的风险. NTP 可以增加安全策略来应对可能的安全风险, 将消息摘要 (MD) 中的 MD5 和安全散列算法 (SHA) 中的 SHA-1 引入 NTP 算法, 有效地验证了数据完整性, 防止数据包被篡改, 以保证时间同步的安全性. 进一步, 针对这两类算法提出对 NTP 包关键数据帧 Hash 加密, 在保持良好同步精度的同时可进一步提高时间同步的安全性. 通过实验对比了 MD5 和 SHA-1 算法加入所带来同步效果的影响. 结果表明: 在 MD5 和 SHA-1 算法加入后, NTP 依然能保持毫秒级的同步性能, 这对于实现 NTP 安全时间同步方法具有重要意义.

关键词: 消息摘要 (MD); 时间戳; 网络时间协议 (NTP); 时差; 时延; 防篡改
中图分类号: P185.18 **文献标志码:** A **文章编号:** 1008-9268(2021)05-0084-08

0 引 言

网络中各个节点的时钟都不相同, 因此若需要实现时间同步, 需要有一个标准参考时钟. 网络时间协议 (NTP) 广泛应用于各类领域, 如电力部门各子系统间的时间同步、金融证券行业需要高精度的时间信息、军事作战平台需要高精度的安全时间溯源 [1-2]. NTP 将时钟同步至协调世界时 (UTC), 对维持各系统间的时间同步具有重要作用.

在 NTP 广泛应用的同时, 其协议的安全性也逐渐受到人们的关注. 由于 NTP 是一种基于用户数据报协议 (UDP) 的网络时间协议, 客户端通过向服务器发送时间同步请求, 利用接收的数据包时间戳解算获得时间同步. 因此, 其无连接的协议特征使得在没有防护措施的网络中有着较高的安全风险 [3-4]. 目前, 对于无防护的 NTP, 攻击者可以伪造服务器欺骗客户端, 通过中间人攻击截取数据包后修改时间戳, 收到该数据包的客户端会同步至错误的时间. 文献 [5] 分析了利用地址解析协议 (ARP) 欺骗伪造服务器, 攻击者可以将客户端拉偏至任意时间. 因此客户端需要对收到的响应包进行校验, 确保数据信息未被篡改. 目前, 国内外已经对 NTP 下的认证手段进行了相应的研究, 利用对称或非对称的加密手段提高协议的安全性. 文献 [6] 提出将 RSA(Rivest-Shamir-Adleman Scheme) 加密算法用于 NTP, 其算法的加入增加了路径时延, 同时降低了同步精度. 文献 [7] 分析对比了最新互联网草案网络时间安全协议 (NTS) 和 NTP 的同步和安全性能. 结果表明: NTS 在提高安全性的同时增加了 CPU 负担, 认为需要提高时间同步的效率. 文献 [8] 则将安全套接层 (SSL) 证书应用在 NTP 中, 并认为使用 ECDSA 算法 (Elliptic Curve Digital Signature Algorithm) 相较于 RSA 算法, 密钥长度会更短, 效率更高.

由于在各类安全协议算法中, 生成签名实现认证首先需要对数据包产生消息摘要. 本文采用较为广泛使用的两类消息摘要算法生成 NTP 报文摘要, 并通过不同方式融合 NTP, 从理论和实验分析其对 NTP 时间同步效果的具体影响.

1 原 理

对于 NTP 的安全同步算法, 为了减少直接通过加

收稿日期: 2021-07-27
资助项目: 国家部委资助项目(2019-JCJQ-JJ-190)
通信作者: 马明 E-mail: maming@nudt.edu.cn

密 NTP 包带来的复杂时间代价, 首先需要对报文提取摘要, 即消息指纹. 通常消息摘要长度为 100～200 bit, 填充中算法的耗时会增加客户端同步等待时间. 目前, 以下两种是较为安全且高效的消息摘要算法.

1.1　MD5

消息摘要算法 (MD) 中的 MD5 算法是一类 Hash 函数, 其以 512 bit 分组来处理输入的信息, 且每一分组又被划分为 16 个 32 bit 子分组[9]. 经过一系列的处理后, 算法的输出由 4 个 32 bit 分组组成, 将这 4 个 32 bit 分组级联后将生成一个 128 bit 散列值.

MD5 算法的第一步是附加填充数据长度. 若原始明文长度为 K, 且

$$K \bmod 512 \neq 448. \tag{1}$$

则需要在输入明文后填充 1 和 n 个 0, 填充后的数据长度为L_m, 则

$$L_m = N \times 512 + 448. \tag{2}$$

第二步为记录信息长度. 该部分为 64 bit, 用来存储填充前数据长度, 填充后整个报文的数据长度 L 是 512 的倍数.

第三步对 MD5 算法初始化缓存. 利用 4 个 32 bit 的整数: A=67 452 301, B=EFCDAB89, C=98BADCFE, D=10 325 476, 以及 4 个函数:

$$\begin{aligned} \boldsymbol{F}(b,c,d) &= (b \wedge c) \vee \left(\overline{b} \wedge d\right), \\ \boldsymbol{G}(b,c,d) &= (b \wedge d) \vee \left(c \wedge \overline{d}\right), \\ \boldsymbol{H}(b,c,d) &= b \oplus c \oplus d, \\ \boldsymbol{I}(b,c,d) &= c \oplus \left(b \vee \overline{d}\right). \end{aligned} \tag{3}$$

式中, a、b、c、d 用于缓存 A、B、C、D, 在不同步骤中有指明的顺序, 即第一分组需要将 A 到 a, B 到 b, C 到 c, D 到 d, 从第二分组开始的变量为上一分组的运算结果, 即$A = a, B = b, C = c, D = d$, 以此类推. 总共进行 4 轮向量运算, 每轮进行 16 次数据运算, 共计 64 次, 最终将输出结果级联得到消息摘要.

由于函数的单向映射性, 使得在得到消息摘要后反推原文几乎是不可行的[10]. 且对于不同的明文, 其映射结果不同, 因此可以用于验证数据报文的完整性, 若数据在传输等过程中受到恶意篡改, 即可以校验其消息摘要来防止攻击验证合法性.

1.2　SHA-1

SHA-1 由美国国家标准与技术学会 (NIST) 和美国国家安全局 (NSA) 共同研发[11]. SHA-1 输入长度在 2^{64} 以内, 输出的消息摘要长度为 160 bit, 其原理和 MD5 算法较为类似, 具体不再赘述, 下面主要描述其不同点.

SHA-1 的第一、二部分同 MD5 算法, 均为填充数据加上 64 bit 的数据长度, 最后得到 512 倍数据长度的消息.

SHA-1 后续算法同理按照 512 bit 长度进行处理, 初始化后 32 bit 的整数相较于 MD5 算法多一个 E=C3D2E1F0, 前四位相同. 算法将 512 bit 块分解为 16 个子块, 算法共 4 轮, 每轮 20 步, 因此共计 80 次, 最终将输出结果级联得到消息摘要. 表 1 对比了 MD5 和 SHA-1 算法, 由表 1 可知, MD5 算法产生的消息摘要更短, 迭代次数更少, 效率更高, 但是 SHA-1 算法的安全性相对更高[12]. 总体来说, 两类消息摘要算法都比较简单, 对于应用在 NTP 这类需要考虑时间精度的协议中比较适宜.

表 1　MD5 算法和 SHA-1 算法对比

特征	MD5算法	SHA-1算法
消息摘要长度/bit	128	160
产生相同消息摘要所需操作/次	2^{64}	2^{80}
轮数/次	4	4
迭代次数	64	80

1.3　基于消息摘要的 NTP 算法设计

将消息摘要融入 NTP, 需要考虑算法的耗时、复杂度、安全性等因素. 且由于 NTP 关键在于时间戳的准确性, 因此除了对整包提取摘要外, 本文提出对 NTP 关键信息的校验.

1.3.1　算法原理

图 1 是基于消息摘要的 NTP 算法原理图. 客户端首先构造 NTP 请求报文, 在利用原始报文哈希算法后产生消息摘要 A 添加至包尾, 该消息摘要和 NTP 原始报文构成 NTP 请求包发送给服务器端. NTP 服务器收到包后, 首先对原始报文进行哈希运算得到消息摘要 B. 对比 A 和 B, 若相等, 则说明请求报文在传输过程中未受到修改, 服务器则利用同样方法构造 NTP 响应包对客户端反馈. 客户端对接收的消息摘要 C 和自己计算的摘要 D 进行校验, 相等则进行时间同步, 若不等则丢弃该包, 不提供时间同步服务.

图 2 为 NTP 原始报文 48 B, 加上其他包头为 90 B, 首先, 需要将 NTP 包填充补长构成 512 bit.

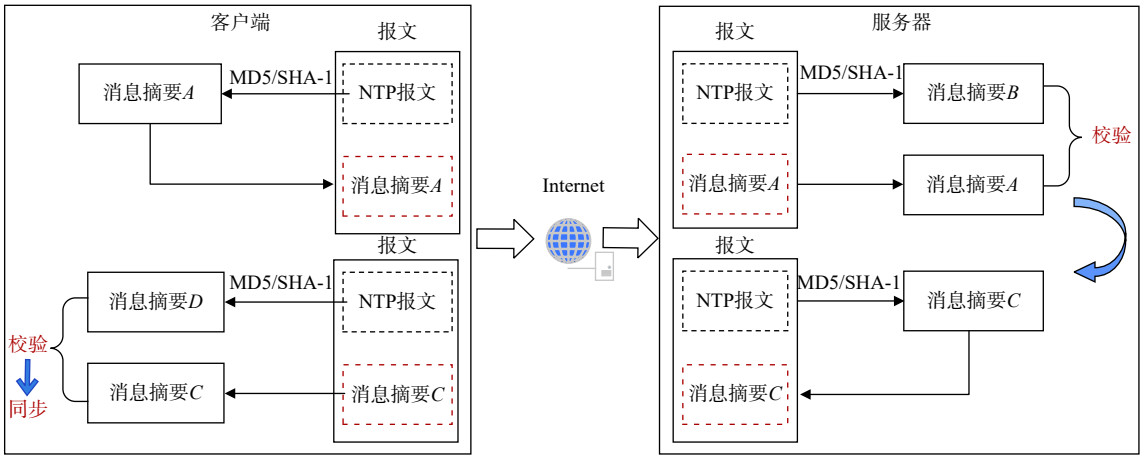


图 1 基于消息摘要的 NTP 算法原理图

图 3 为 NTP 完整数据包格式. 设定扩展字段 MD5 算法为 20 B, SHA-1 为 24 B, 其中产生的消息摘要分别占据 16 B 和 20 B. 将消息摘要填充在 NTP 包后加上消息摘要的数据包长度为 110/114 B.

通常消息摘要是针对 NTP 的 48 B 包进行处理. 但由于客户端在进行时间同步时, 重点提取包中的 T_2 和 T_3 时戳. 因此, 此处的消息摘要考虑两种方式, 分别为对完整 NTP 包和仅对 T_2 和 T_3 进行 Hash 运算.

将消息摘要运用在完整包和时间戳上进行对比, 如图 4 所示, 由于 NTP 数据包较小, 因此无论哪种方式经过填充补偿后的包均为 512 bit. 若对 48 B 的数据包补长, 需要补充 64 bit, 而若对 128 bit 时间戳提取消息摘要, 则需要补长 320 bit. 相较于另一种 Hash 方式, 该种方式的真实数据隐藏在更多填充的无效字节中. 由于 Hash 算法存在极小概率使得不同原始数据能够生成相同的消息摘要. 一旦攻击者篡改的明文包含关键信息而恰好消息摘要校验相等, 则客户端会同步至错误时间. 而通过对时间戳提取消息摘要, 攻击者更改数据比特发生在有用字节的概率会相对较小, 安全性更高.

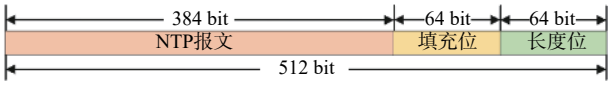


图 2 NTP 数据包补位

v4	IHL=20	TOS	Total Length=76				
IPID			x	DF	MF	Frag Offset	
TTL		Protocol=17	Ip Header Checksum				
Source IP							
Destination IP							
Source Port=123			Destination Port=123				
Length=56			UDP Checksum				
LI	v4	Mode	Stratum	Poll	Precision		
Root Delay							
Root Dispersion							
Reference ID							
Reference Timestamp							
Origin Timestamp							
Receive Timestamp							
Transmit Timestamp							
Message Digest							

图 3 NTP 数据包格式

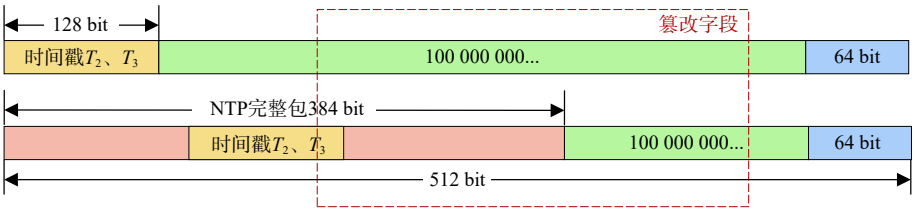


图 4 两种消息摘要计算方式对比

1.3.2 理论分析

图 5 为 NTP 同步流程图, 其中客户端发送和接收时间为 T_1 和 T_4, 服务器接收和发送时间为 T_2 和 T_3, 传输路径产生的时延分别为 δ_1 和 δ_2, 客户端与服务器间的时差为 θ, 客户端与服务器 Hash 算法所耗时间为 τ_1 和 τ_2, 校验消息摘要所耗时间分别为 σ_1 和 σ_2.

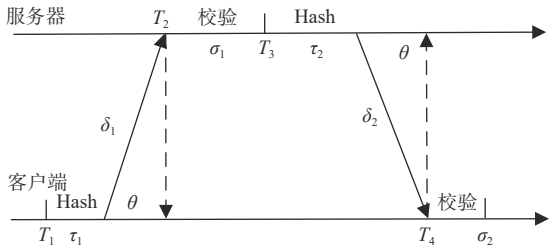


图 5　NTP 同步流程

由图 5 可知, 由于在客户端和服务器端使用相同的 Hash 算法, 因此可以认为

$$\tau_1 = \tau_2 = \tau. \tag{4}$$

对报文进行校验的时间并不影响客户端和服务器发送、接收报文的时间戳. 因此可以得到:

$$T_2 = T_1 + \theta + \delta_1 + \tau, \tag{5}$$

$$T_4 = T_3 - \theta + \delta_2 + \tau. \tag{6}$$

由于 NTP 的对称性, 双向路径时延可以视为相等

$$\delta_1 = \delta_2 = \delta. \tag{7}$$

因此, 最终得到单向时延 δ'和时差 θ 分别为:

$$\delta' = \delta + \tau = \frac{(T_2 - T_1) + (T_4 - T_3)}{2}, \tag{8}$$

$$\theta = \frac{(T_2 - T_1) - (T_4 - T_3)}{2}. \tag{9}$$

由结果可知, 理论上 MD5 和 SHA-1 算法的复杂度并不会影响 NTP 时间同步的精度, 这是由于算法相同使得路径的对称性消除了算法耗时对于精度的影响. 但是, 算法越复杂, 路径时延越长. 所以, SHA-1 算法复杂度比 MD5 算法复杂度高, 理论上时延更长, 但是同步精度并不会受到劣化. 所以若是为了更高的安全性, 性能似乎更为优越. 但是, 整个算法的复杂度会影响到客户接收时间戳 T_4 的值, 即

$$T_4 = T_1 + 2\delta + 2\tau + \sigma_1. \tag{10}$$

因此, 客户端等待服务器响应数据包的时间会增长. 由于实际网络的波动性以及不同算法对于设备硬件带来的影响, 实际的 NTP 同步性能可能会随着算法复杂度的上升而劣化. 下面将分析 MD5 和 SHA-1 算法在 NTP 同步过程中造成的实际影响.

2　算法流程

2.1　客户端算法流程

算法均在 Windows 平台下执行, 语言为 C++. 客户端算法如图 6 所示, 首先客户端构造同步请求包, 并利用 MD5 和 SHA-1 算法对原数据包 Hash 后得到报文摘要. 这里的 Hash 包含对前 48 B 的 NTP 数据包处理以及对两个时戳共 128 bit 进行处理, 这是由于该部分是防止恶意篡改的关键数据. 客户端计算摘要后, 与原文共同组帧发送给服务器, 等待服务器响应. 设定等待时间为 T_w, 若等待时间大于 T_w, 则客户端输出超时处理, 若小于则接收该数据包, 并校验包中的报文摘要. 若相等, 则进行时间同步, 同时保存时差、时延等数据至本地文件.

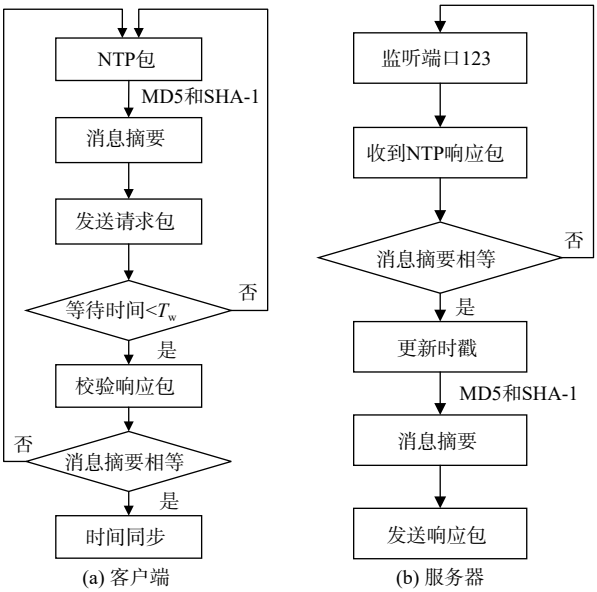


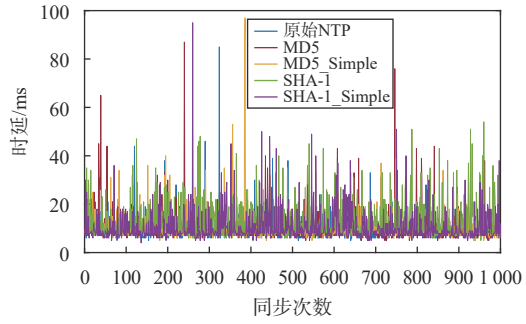
图 6　算法流程

2.2　服务器算法流程

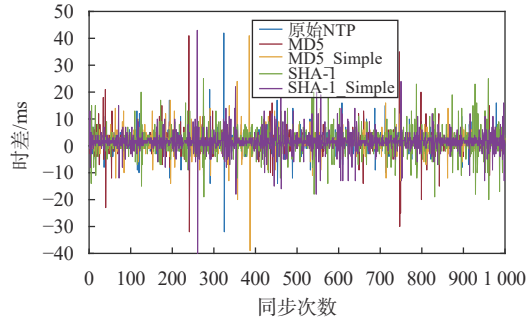
服务器端首先建立 Socket 监听 UDP 端口 123, 接收到请求包后校验包中的消息摘要, 若相等则更新本地时间, 打上时戳 T_2 和 T_3. 接着通过 MD5 和 SHA-1 算法对数据包或者时戳进行 Hash. 得到的消息摘要附在 NTP 包尾后构造 NTP 响应包反馈给客户端.

3　实验结果

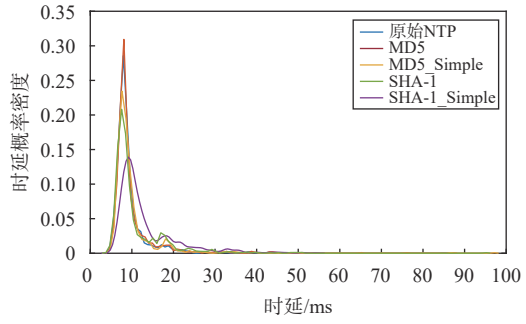
客户端和服务器由两台 PC 机执行, 其中服务器 CPU 为 i5 八代系列, 主频 1.60 GHz, 客户端 CPU 为 i5 五代系列, 主频 2.20 GHz, 其均在 Windows 下同一个局域网内, 连接速度为 72 Mibit/s. 实验设置客户端等待响应时间 T_w 为 100 ms, 并间隔 1 s 一次向服务器发送同步请求. 将未添加 Hash 的 NTP 算法称为 Original, 对完整 NTP 包 Hash 的算法分别称为 MD5 算法和 SHA-1 算法, 而对时间戳提取的 Hash 算法称为 Simple 类. 实验共对比了在 1 000 次客户端的时间同步中原始 NTP、MD5、MD5_Simple、SHA-1、SHA-1_Simple 5 种算法下的实验数据, 实验结果如图 7 所示.



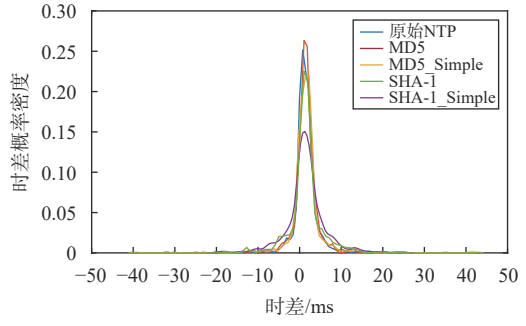
(a) 时延



(b) 时差



(c) 时延概率密度函数



(d) 时差概率密度函数

图 7 时延结果

从实验结果来看, 5 种算法的时延普遍小于 50 ms, 时差波动范围在 3σ 范围内, 且由概率密度函数可以看出, 时延主要集中在 10 ms, 时差则主要集中小于 5 ms. 同时可以看到, 采用的 SHA-1 的实验结果其时延和时差值要更大. 下面对时延和时差的统计数据进行分析, 如图 8～9 和表 2 所示.

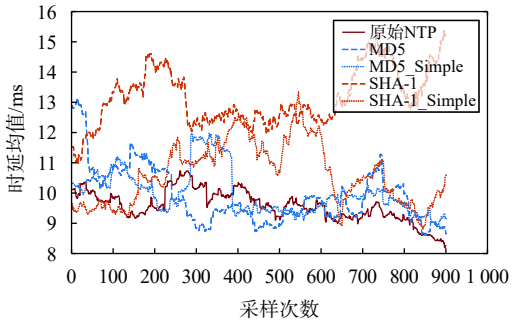


图 8 时延均值实时采样

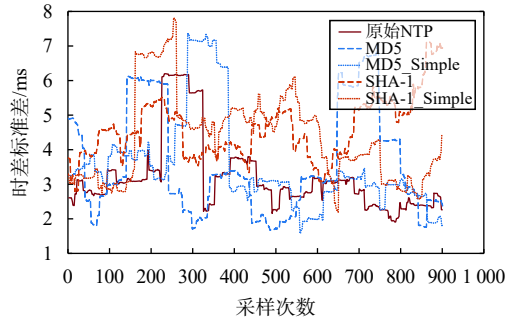


图 9 时差标准差实时采样

表 2 不同算法统计数据对比 ms

算法	时延均值	时差标准差
原始NTP	9.538 0	3.295 1
MD5	10.004 0	3.926 2
MD5_Simple	9.985 0	3.639 7
SHA-1	13.192 0	4.789 0
SHA-1_Simple	10.631 0	4.462 7

由图 8～9 和表 2 中对 1 000 次同步结果计算值可知, 未添加 Hash 算法的 NTP 其时延均值和时差标准差更小, 时延均值基本在 10 ms 以内, 时差标准差即同步精度在 3～6 ms. 而添加了 MD5 算法的 NTP 的时延均值在 8～13 ms, 时差标准差在 2～7 ms 间波动. 添加了 SHA-1 算法的 NTP 时延均值在 11～16 ms, 时差标准差在 3～8 ms. 由此看出, Hash 算法在实际应用中会部分影响 NTP 的同步精度, 且算法的复杂性上升, 同步精度略微下降. 但是, NTP 依然能够保持优于 5 ms 的同步精度, 因此, 在同步精度要求不是过于高的场合下, 两种算法均具有较好的适应性. 同时, 对比在同样 Hash 算法的条件下两种 Hash 方式所带来的影响. 可以看到, 无论是 MD5 还是 SHA-1 算法, 对关键时间戳 Hash 后其同步精度和时延并没有较为明显的改变, 其性能与直接对数据

包 Hash 的结果相当. 这是由于 NTP 数据包本身字节数较少, 在算法第一步填充补位时, 并不需要扩展到 512 的更大倍数, 均只填充至 512 bit, 所以算法整体的迭代次数并没有改变. 但是, 该方法将时间戳隐藏于大量无效比特中, 提高了算法的安全性. 同时, 处理数据长度的减小为后期实现 NTP 对称或非对称加密认证减小运算的复杂性, 提高算法的效率提供了有效的思路.

如图 10 和表 3 为下面对 5 类算法中客户端一次 Hash 运算到请求数据包所耗的时间进行分析.

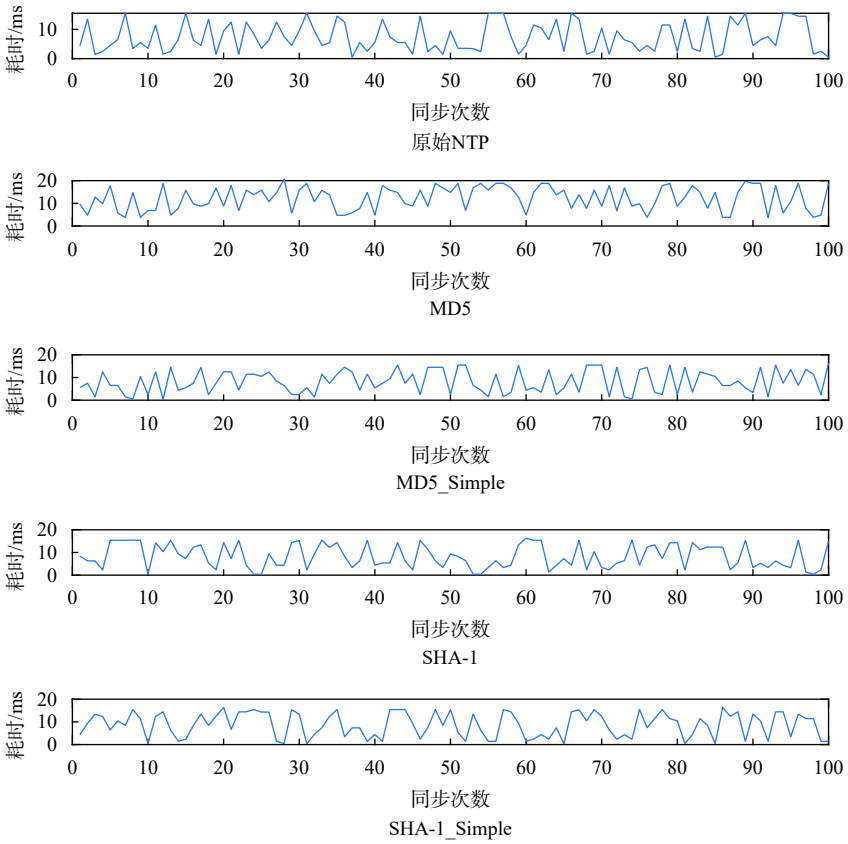


图 10　单次算法耗时对比

表 3　不同算法耗时对比　　ms

算法	波动范围	均值
原始NTP	0.3～15	4.976 3
MD5	0.4～17	5.198 0
MD5_Simple	0.4～16	4.973 9
SHA-1	0.4～16	5.224 5
SHA-1_Simple	0.4～16	5.361 9

结果表明, 无 MD5 和 SHA-1 算法添加的 NTP 耗时最少, 在 0.3～15 ms 波动, 均值为 4.976 3 ms. 而添加了 MD5 和 SHA-1 算法的 NTP, 其同步请求耗时都增长了约 1～2 ms, 时长在 0.3～17 ms 内波动. 其中, MD5 算法和 MD5_Simple 的耗时均值为 5.198 0 ms 和 4.973 9 ms, SHA-1 算法和 SHA-1_Simple 的耗时均值分别为 5.224 5 ms 和 5.361 9 ms. 所以, 由于 MD5 算法的迭代次数要比 SHA-1 算法少, 其耗时要更少. 而对关键数据进行 Hash 后的耗时与对完整数据包处理的耗时相差无异, 原因同上述分析.

对 5 种算法所占主机 CPU 进行分析, 经统计未加消息摘要算法其平均占用 CPU 约 0.26%, 而添加了 MD5 算法、SHA-1 算法等几种算法平均占用 CPU 约在 0.3%～0.4%. 因此消息摘要算法的添加会增加 CPU 的负担, 但是其消耗并不明显, 且不同算法间差距不大. 同时, 上述实验均在网络负载较良好的环境中进行, 本地流量在 0～20 Kibit/s. 通过数据包模拟网络环境在 UDP 添加无序包, 使得本地流量波动在 3～4 Mibit/s 时, 统计了该环境下 5 种算法的统计特性如表 4 所示. 结果表明: 在网络环境劣化后, 其 NTP 在 5 种算法下的同步精度优于 10 ms, 时延均值上升到 14～20 ms, 不同算法间的性能差异和网络环境良好的情况下结论无异. 另外, 本地流量的上升也影响了算法的耗时, 其普遍增加了 5～6 ms.

表 4 本地流量增加后不同算法统计数据对比 ms

算法	时延均值	时差标准差
原始NTP	14.284 0	6.121 8
MD5	17.605 0	7.054 8
MD5_Simple	16.415 0	7.075 7
SHA-1	20.332 0	7.967 9
SHA-1_Simple	18.275 0	8.006 3

4 结束语

本文针对当前 NTP 协议存在的安全风险与认证协议，分析了不同消息摘要算法对于协议的同步性能带来的影响. 理论分析得知，无论任何协议的添加均不会对 NTP 的同步精度带来影响. 然而实验表明，算法的复杂性会造成 NTP 同步精度的轻微劣化. SHA-1 算法的迭代次数比 MD5 算法要多，因此整体的算法耗时、同步精度、时延等性能均有略微下降，但是整体并不影响 NTP 的同步效果，在网络环境良好的局域网内客户端依然能够保持优于 10 ms 的同步精度，这对于大多数应用场合来说已经足够了. 因此，为了更高安全的同步效果，SHA-1 算法可以作为认证过程中的哈希函数. 同时，本文提出了对关键数据时间戳做哈希处理，将哈希算法对数据包中的 T_2 和 T_3 时戳进行计算. 实验表明：由于 NTP 数据包本身较小，其效果与对完整数据包进行消息摘要计算的结果相差无异，均能实现较高的时间同步精度. 同时，时间戳的隐藏使得攻击者不容易篡改至关键比特增加了算法的安全性，其减少数据处理长度对于利用密钥加密实现双向认证的算法也提供了有效的思路.

参考文献

[1] LÉVESQUE M, TIPPER D. A survey of clock synchronization over packet-switched networks[J]. Communications surveys and tutorials, 2016, 18(4): 2926-2947. DOI: 10.1109/COMST.2016.2590438

[2] 李培基, 李卫, 朱祥维, 等. 网络时间同步协议综述[J]. 计算机工程与应用, 2019, 55(3): 30-38.

[3] BISHOP M. A security analysis of the NTP protocol version 2[C]//The 6th Annual Computer Security Applications Conference, IEEE, 1990. DOI: 10.1109/CSAC.1990.143746

[4] 黄九梅, 洪锡联, 赵英. 网络时间同步及其安全性研究[J]. 中国科技信息, 2008(16): 97-98.

[5] 刁造翔, 章小宁, 王淑君, 等. 局域网条件下的NTP伪造服务器攻击技术[J]. 电子信息对抗技术, 2016, 31(6): 63-68.

[6] 彭栋, 郭伟. 安全网络授时服务技术研究[J]. 时间频率学报, 2018, 41(1): 37-45.

[7] LANGER M, TEICHEL, K, SIBOLD D, et al. Time synchronization performance using the network time security protocol[C]//European Frequency and Time Forum (EFTF), 2018. DOI: 10.1109/EFTF.2018.8409017

[8] KOGCE M, SISECI N E. A new approach to security of N TP via SSL certificates[C]// The 1st International Informatics and Software Engineering Conference (UBMYK), 2019. DOI: 10.1109/UBMYK48245.2019.8965454

[9] 周琴琴. 基于Hash函数的MD5和SHA-1加密算法研究及其硬件实现[D]. 合肥: 安徽大学, 2012.

[10] 王小云, 于红波. 密码杂凑算法综述[J]. 信息安全研究, 2015, 1(1): 19-30.

[11] 王孟钊. SHA1算法的研究及应用[J]. 信息技术, 2018(8): 152-153, 158.

[12] 吴松魁. 基于UVM的HASH类算法IP核验证[D]. 西安: 西安电子科技大学, 2020.

作者简介

陈曦 (1997—), 女, 硕士, 研究方向为时间频率系统技术.

臧文驰 (1994—), 男, 硕士, 研究方向为时间频率系统技术.

马明 (1989—), 男, 博士, 研究方向为时间频率系统技术.

Research on secure NTP method based on message digest encryption

CHEN Xi, ZANG Wenchi, MA Ming, GONG Hang, SUN Guangfu

(*College of Electronic Science and Technology*, *National University of Defense Technology*, *Changsha* 410073, *China*)

Abstract: At present, the network time protocol (NTP) is the main time protocol used for time synchronization in wired networks. It can achieve a synchronization accuracy of more than ten milliseconds

in wide area networks and a few milliseconds in local area networks. However, due to the openness of the protocol, there is no security protection. Under the circumstances, it is extremely vulnerable to network attacks, which brings potential risks to customers who need high security. The NTP protocol can increase security strategies to deal with possible security risks. The message digest algorithm 5 (MD5) and the secure hash algorithm (SHA-1) message digest algorithm is introduced into the NTP protocol algorithm, which is effective to verify data integrity and prevent data packets from being tampered with to ensure the security of time synchronization. Further, for these two types of algorithms, Hash encryption of key data frames of NTP packets is proposed, which can further improve the security of time synchronization while maintaining good synchronization accuracy. Experiments have compared the influence of the synchronization effect brought by the addition of the algorithm. The results show that after the message digest algorithm is added, NTP can still maintain millisecond-level synchronization performance, which is of great significance to the realization of the NTP secure time synchronization method.

Keywords: message digest (MD); timestamp; the network time protocol (NTP); offset; delay; anti-tampering

(上接第 64 页)

guidance application, which is an important basis for reliability. However, at present, ordinary users only pay attention to the accuracy of navigation and positioning, and do not consider the reliability of positioning results. The airborne of navigation field has high requirements for the reliability of positioning results. Therefore, this paper studies and analyzes the integrity fault monitoring performance. A clock error adjustment strategy is proposed, and the effectiveness of the method is verified by simulation, the influence of interity fault removed and non-removal on positioning accuracy and integrity is given. It is proved that integrity fault monitoring can ensure the system performance, operation ability and safety performance of GLS to meet the expected requirements.

Keywords: the GBAS landing system (GLS); precision approach; integrity; clock error; accuracy