Ежегодная международная научно-практическая конференция
«РусКрипто’2021»

Криптографические аспекты протокола TSP

Алексеев Е. К., к.ф.-м.н., начальник отдела криптографических исследований, КриптоПро
Бабуева А. А., инженер-аналитик, КриптоПро
Никифорова Л. О., инженер-аналитик, КриптоПро
Смирнов П. В., директор по развитию, КриптоПро

TSP (Time Stamp Protocol)

Участники:

- запрашивающая сторона
- служба штампов времени (Time Stamp Authority – TSA)

Назначение:

- установление факта существования данных до определенного времени

Является частью инфраструктуры PKI (наряду с OCSP, DVCS и т.п.)

Примеры применения

- Электронный документооборот
- Патентование
- Ставки

Статус протокола

- IETF: RFC 3161 (2001 г)
- ТК26:
  - подготовлен проект методических рекомендаций «Использование российских криптографических алгоритмов в протоколе штампов времени (TSP)» (январь 2021 г)
  - в настоящее время идет экспертиза
- Аналоги:
  - ISO – ISO/IEC 18014 (+ связанные штампы времени)
  - ANSI – ANSI ASC X9.95 standard (+ связанные штампы времени, + возможность получать штамп для аутентифицированных данных)

Описание протокола

C		TSA $(d_{TSA}, Q_{TSA}), Cert_{TSA}$
• $H = \mathrm{Hash}(data)$	TimeStampReq → $hashID, H$	
		• проверка $hashID$ и H
	TimeStampResp ← $H, SN, time, sgn, Cert_{TSA}$	• $time$ – время • SN – уникальное целое число • $sgn = Sign_{d_{TSA}}(H \parallel SN \parallel time)$
• проверка		

– сообщение типа SignedData формата CMS

Описание протокола

C		TSA $(d_{TSA}, Q_{TSA}), Cert_{TSA}$
• $H = \mathrm{Hash}(data)$	TimeStampReq → $hashID, H$	
		• проверка $hashID$ и H
штамп времени	TimeStampResp ← $H, SN, time, sgn, Cert_{TSA}$	• $time$ – время • SN – уникальное целое число • $sgn = Sign_{d_{TSA}}(H \parallel SN \parallel time)$
• проверка		

– сообщение типа SignedData формата CMS

Основная задача

Подтвердить, что данные существовали в информационной системе на момент времени, указанный в штампе

Основная задача

Подтвердить, что данные существовали в информационной системе на момент времени, указанный в штампе

За счет чего она решается?

Подпись

стойкость к построению подделки

⇩

сложно сформировать поддельный штамп времени

C		TSA $(d_{TSA}, Q_{TSA}), Cert_{TSA}$
• $H = \mathrm{Hash}(data)$	TimeStampReq → $hashID, H$	
		• проверка $hashID$ и H
	TimeStampResp ← $H, SN, time, sgn, Cert_{TSA}$	• $time$ – время • SN – уникальное целое число • $sgn = Sign_{d_{TSA}}(H \parallel SN \parallel time)$
• проверка		

– сообщение типа SignedData формата CMS

Хэш-функция

стойкость хэш-функции

сложно подобрать новые данные для старого штампа

C		TSA $(d_{TSA}, Q_{TSA}), Cert_{TSA}$
• $H = \mathrm{Hash}(data)$	TimeStampReq → $hashID, H$	
		• проверка $hashID$ и H
	TimeStampResp ← $H, SN, time, sgn, Cert_{TSA}$	• $time$ – время • SN – уникальное целое число • $sgn = Sign_{d_{TSA}}(H \parallel SN \parallel time)$
• проверка		

– сообщение типа SignedData формата CMS

Дополнительное свойство

журналирование + SN

защита при компрометации ключа подписи службы штампов времени

C		TSA $(d_{TSA}, Q_{TSA}), Cert_{TSA}$
• $H = \mathrm{Hash}(data)$	TimeStampReq $hashID, H$	
		• проверка $hashID$ и H
	TimeStampResp $H, SN, time, sgn, Cert_{TSA}$	• $time$ – время • SN – уникальное целое число • $sgn = Sign_{d_{TSA}}(H \parallel SN \parallel time)$ • запись в журнале
• проверка		

– сообщение типа SignedData формата CMS

Не требуется защищенный канал

целостность обеспечивается за счет проверки пользователем:

- соответствие H
- подпись sgn

C		TSA $(d_{TSA}, Q_{TSA}), Cert_{TSA}$
• $H = \mathrm{Hash}(data)$	TimeStampReq $hashID, H$	
		• проверка $hashID$ и H
	TimeStampResp $H, SN, time, sgn, Cert_{TSA}$	• $time$ – время • SN – уникальное целое число • $sgn = Sign_{d_{TSA}}(H \parallel SN \parallel time)$
• проверка		

– сообщение типа SignedData формата CMS

Не требуется защищенный канал

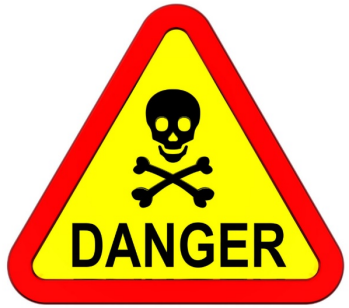
соответствие запроса и ответа при получении нескольких штампов для одних и тех же данных обеспечивается за счет:

- локального источника времени
- $nonce$ (случайный!)

C		TSA $(d_{TSA}, Q_{TSA}), Cert_{TSA}$
• $H = \mathrm{Hash}(data)$ • $nonce$ – случайное целое число	TimeStampReq → $hashID, H, nonce$	
		• проверка $hashID$ и H
	TimeStampResp ← $H, SN, time, nonce, sgn, Cert_{TSA}$	• $time$ – время • SN – уникальное целое число • $sgn = Sign_{d_{TSA}}(H \parallel SN \parallel time \parallel nonce)$
• проверка		

– сообщение типа SignedData формата CMS

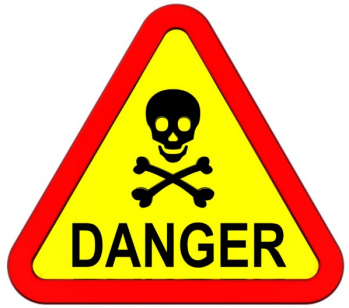
Не требуется защищенный канал



конфиденциальность данных не обеспечивается

Малоэнтропийные данные могут быть восстановлены путем перебора соответствующих хэш-значений

От чего еще не защищает протокол TSP?



Протокол TSP не обеспечивает, что данные, возникшие в системе, появились у конкретного пользователя и в соответствии с «правилами игры», установленными данной системой

У противника есть возможность:

- «угадать» данные до момента их публикации и получить штамп времени
- получить штамп времени на данные, появившиеся у другого пользователя

Задача обеспечения расширенных свойств безопасности должна решаться на более высоком уровне!

Спасибо за внимание!

Контактная информация:

babueva@cryptopro.ru