

Tentamen Security, 19 Januari 2007, 15:30 - 17:30

Er zijn totaal 100 punten te verdienen; bij elk onderdeel staat aangegeven hoeveel punten het waard is.

Multiple Choice Vragen

Er zijn soms meerdere antwoorden redelijk; geef steeds het beste antwoord. Voorzie je antwoord van een korte en bondige motivatie.

1. **(Securitydoelen, 7 punten)** Na het tentamen lukt het een college student om jouw tentamen uit de stapel ingeleverde tentamens te halen en zijn naam in plaats van de jouwe erboven te zetten. Welk veiligheidsdoel is hiermee gecompromitteerd?
 - (a) authenticiteit
 - (b) beschikbaarheid (availability)
 - (c) confidentialiteit
 - (d) correctheid
 - (e) integriteit
 - (f) onloochenbaarheid (non-repudiation)
2. **(Securitydoelen, 7 punten)** Terwijl de docent even luiers is verschonen, waait je tentamen toevallig van de stapel in het haardvuur. De docent weet van niets, en beoordeelt je dan als 'niet deelgenomen'. Welk veiligheidsdoel is hier in het geding?
 - (a) authenticiteit
 - (b) beschikbaarheid (availability)
 - (c) confidentialiteit
 - (d) correctheid
 - (e) integriteit
 - (f) onloochenbaarheid (non-repudiation)
3. **(Securitydoelen, 7 punten)** Tijdens een tentamen kijkt een medestudent bij je af en schrijft een deel van je antwoord over. Welk veiligheidsdoel wordt hier geschonden? (Denk goed na!)
 - (a) authenticiteit
 - (b) beschikbaarheid (availability)
 - (c) confidentialiteit
 - (d) correctheid
 - (e) integriteit
 - (f) onloochenbaarheid (non-repudiation)

4. **(Herhaalde Caesar-ciphers, 7 punten)** Een slimmerik wil de Caesar-cipher die in zijn geheime spionnenclubje gebruikt wordt verbeteren door hem tweemaal achter elkaar toe te passen met twee verschillende sleutels. Bij een gegeven plaintekst, hoeveel mogelijke verschillende ciphertekst zijn er?

- (a) 26
- (b) $2 \cdot 26$
- (c) $\binom{26}{2}$
- (d) 26^2
- (e) 26!
- (f) 26^{26}

Open Vragen

5. (**Gedistribueerd munt werpen**) Alice en Bob willen op afstand (bijvoorbeeld telefonisch of via email) samen een random bit kiezen. Ze vertrouwen elkaar echter niet, dus kan niet zomaar één van beiden het bit kiezen en het naar de ander sturen. Ze verzinnen het volgende protocol:

$$A \rightarrow B: h(n_A)$$

$$B \rightarrow A: h(n_B)$$

$$A \rightarrow B: n_A$$

$$B \rightarrow A: n_B.$$

D.w.z., Alice kiest een nonce n_A en stuurt eerst de hash van deze nonce naar Bob. Bob doet hetzelfde. Zodra Alice Bob's hash waarde heeft ontvangen, stuurt ze de nonce n_A zelf en Bob doet weer hetzelfde. Alice en Bob berekenen nu $n_A + n_B$; het gekozen bit is het minst significante bit van n_A en n_B . (Voor alle duidelijkheid: we beschouwen in deze situatie niet een externe aanvaller die berichtjes onderscheept of verandert. Het gaat erom dat Alice en Bob elkaar niet vertrouwen, maar toch samen een random bit willen kiezen.)

- (a) (**8 punten**) Leg kort uit waarom het voor dit protocol belangrijk is dat de hash functie h collision-resistant is. (Hint: neem aan dat de hash functie h *niet* collision-resistant is en verzin nu een manier waarop Alice of Bob kan valsspelen.)
- (b) (**8 punten**) Leg kort uit waarom het voor dit protocol belangrijk is dat de hash functie h one-way is. (Hint: neem nu aan dat de hash functie h *niet* one-way is en verzin een manier waarop Alice of Bob kan valsspelen.)
- (c) (**8 punten**) Je zou kunnen proberen dit protocol te vereenvoudigen door Alice en Bob geen nonces te laten kiezen, maar bits (b_A resp. b_B). Aan het eind van het protocol is het gekozen bit dan $b_A \text{ xor } b_B$. Leg kort uit waarom dit niet werkt.
6. (**Needham-Schröder-Lowe**) Alice is klant van Bol.nl. Om het boek Applied Cryptography van Bruce Schneier te kunnen kopen, moeten Alice (A) en Bol.nl (B) zich eerst aan elkaar authenticeren. We nemen aan dat iedereen de beschikking heeft over voldoende-gecertificeerde publieke sleutels van de andere deelnemers (bijvoorbeeld via een public key server). Voor de authenticatie gebruiken ze het volgende protocol,

$$A \rightarrow B: \{A, n\}_{pk_B}$$

$$B \rightarrow A: \{n, n'\}_{pk_A}$$

$$A \rightarrow B: \{n'\}_{pk_B}.$$

Hierbij zijn n en n' nonces (verzonden door A , resp. B).

- (a) (8 punten) Dit protocol is echter niet veilig. Als een aanvaller Eve (E) Alice er toe kan bewegen een sessie met haar te beginnen, kan ze zich bij Bol.nl voordoen als Alice. De aanval begint als volgt:

$$A \rightarrow E: \{A, n\}_{pk_E}.$$

Maak deze aanval af.

- (b) (8 punten) Het eerste berichtje in deze aanval wordt gestuurd door de klant Alice en niet door de aanvaller Eve. Beschrijf kort een manier waarop Eve dit in de praktijk voor elkaar zou kunnen krijgen.
- (c) (8 punten) Om bovenstaande aanval te voorkomen, kun je het tweede berichtje, $\{n, n'\}_{pk_A}$, in het oorspronkelijke protocol veranderen in $\{?, n, n'\}_{pk_A}$. Wat kun je op de plaats van '?' zetten om het protocol te repareren? Leg je antwoord kort uit.

7. (Rivest-Shamir-Adleman)

- (a) (8 punten) Op de complexiteit van welk wiskundig probleem is de *veiligheid* van het RSA crypto systeem gebaseerd?
- (b) (8 punten) In de theorie maak je als volgt een RSA sleutelpaar: kies twee willekeurige grote priemgetallen p en q en definieer $n = pq$ en $m = (p - 1)(q - 1)$; kies een willekeurige e met de eigenschap dat $\text{ggd}(e, m) = 1$; bereken met het Euclidisch algoritme een d zodanig dat $de \equiv 1 \pmod{m}$. De geheime sleutel is dan (p, q, d) en de publieke sleutel (n, e) .

In de praktijk werkt het iets anders. Men kiest $e = 3$ (altijd) en zoekt vervolgens twee willekeurige grote priemgetallen p en q zodanig dat $\text{ggd}(e, m) = 1$. De rest gaat hetzelfde. Het is mogelijk te bewijzen dat deze praktijk methode net zo veilig is als de theoretische methode.

De praktijk methode heeft het volgende voordeel: om te versleutelen moet je modulo n tot de macht e verheffen; aangezien e klein is gaat dit snel.

In plaats van deze twee methoden zou je ook het volgende kunnen proberen. Kies $d = 3$ (altijd); zoek twee willekeurige grote priemgetallen p en q met de eigenschap dat $\text{ggd}(d, m) = 1$; bereken met het Euclidisch algoritme e zodanig dat $de \equiv 1 \pmod{m}$.

Deze methode heeft een vergelijkbaar voordeel: om te ontsleutelen moet je modulo n tot de macht d verheffen; aangezien d klein is gaat dit snel.

Desondanks is deze laatste methode bijzonder onverstandig. Leg uit waarom deze laatste methode helemaal *niet veilig* is.

- (c) (8 punten) De publieke RSA sleutel van Alice is $(85, 3)$, i.e., $n = 85$ en $e = 3$. Wat is haar geheime RSA sleutel? Geef je berekening.

Antwoorden (zonder uitleg)

- 1) (a) authenticiteit
- 2) (f) onloochenbaarheid
- 3) (a) authenticiteit
[niet: confidentialiteit - dat is in deze context geen security doel]
- 4) (a) 26
- 5a) Bob zoekt n_B en $n_{B'}$ met $h(n_B) = h(n_{B'})$ en $\text{lsd}(n_B) \neq \text{lsd}(n_{B'})$. Na n_A gezien te hebben kan Bob helemaal bepalen wat het 'random' bit wordt.
- 5b) Bob berekent een n zodat $h(n_A) = h(n)$. Hij hoopt nu dat $n = n_A$. Met deze kennis kan hij het 'random' bit helemaal bepalen.
- 5c) Bob kan zien of Alice een 0 of 1 stuurde door $h(0)$ en $h(1)$ te berekenen en daardoor het 'random' bit helemaal bepalen.
- 6a) $A \rightarrow E: \{A, n\} E \rightarrow B: \{A, n\} B \rightarrow E: \{n, n'\} A \rightarrow A: \{n, n'\} A \rightarrow E: \{n'\} E \rightarrow B: \{n'\} B$
- 6b) Eve begint zelf een service met hetzelfde authenticatieprotocol en hoopt dat Alice klant wordt.
[niet: phishing - het is niet nodig dat Alice denkt met Bob te communiceren]
- 6c) B
- 7a) priemfactorisatie
- 7b) met d , deel van de geheime sleutel, kun je alles ontsleutelen
[je hebt de priemfactorisatie niet nodig]
- 7c) $p=5, q=17, d=43$