Security Protocols and Verification

Defense of Cryptographic Protocol

Garance Frolla
Ely Marthouret
Ewan Decima

Team: **ASKO OM8464A2**

September / November 2025

Contents

1 Attack against PROTOxyde d'alCOl

1.1 Notation

- Let $\mathcal{K}_I$ denote the set of all facts known to the intruder I.
- Let $\big(K_I\big)_{I\in\mathcal{I}}$ the set of key use by I during the *bruteforce* step.
- Let $\mathcal{M}$ denote the set of all messages sent during a communication between to agents.
- Let $M_{1,A} \in \mathcal{M}$ denote the first message sends by Alice, i.e., $M_{1,A} = \{|\langle A, N_A\rangle|\}_{K_{AB}}$
- Let $s(\cdot,\cdot)$ denote the sender function, i.e., $s(m,x)$ means message M is sent by agent C.
- Let $[\cdot]_{(\cdot)}$ denote the extract function of a tuple message, i.e., for $M = \langle m_1, m2, ..., m_n\rangle \in \mathcal{M}$, $[M]_i = m_i \quad \forall i \in [|1,n|]$.
- Let $\langle X', Y', Z', \Sigma'\rangle$ denote four random value.

1.2 The attack

- First this our understanding of your attack : the intruder I steal the first alice's message : $M_{1,A}$. At this step $K(N_A) \notin \mathcal{K}_I$ and $K\big(K_{AB}\big) \notin \mathcal{K}_I$.
- After that, I impersonates S by crafting the ticket $\{|\langle A, \tau, \lambda, K_{AB}\rangle|\}_{K_{BS}}$, replacing K_{BS} with keys K_I to form $T_I := \{|\langle X', Y', Z', \Sigma'\rangle|\}_{K_I}$ and sending $M_{1,A}$ and T_I to B.
- B gets $M_{1,A}$. At this point $K\big(S(M_{1,A}, A)\big) \notin \mathcal{K}_B$. But it's normal according to the ASKO OM8464A2 protocol. Then B gets the crafted ticket T_I. B will decipher it with K_{BS} and send back to $[dec(T_I, K_{BS})]_1$

  $$\Big\{\Big|dec\big(\{|N_A+1|\}_{KAB}, [dec(T_I, K_{BS})]_4\big)\Big|\Big\}_{[dec(T_I,K_{BS})]_4}$$

  For better understanding, let us denote $[dec(T_I, K_{BS})]_1$, $[dec(T_I, K_{BS})]_2$, $[dec(T_I, K_{BS})]_3$, $[dec(T_I, K_{BS})]_4$ as X, Y, Z, Σ respectively.
- The attack lies on the fact that identities are short bitstring, B will always decipher the ticket T_I with his symmetric key K_{BS}, and hoping that:

$$\exists\, J \in \mathcal{I}\,|J \neq BS \wedge\ dec(T_J, K_{BS}) = \langle A, Y, Z, \Sigma\rangle.$$

With such a key K_J, B will think that Σ is K_{AB}. At this point $K\big(\Sigma\big) \notin \mathcal{K}_I$ and $K\big(K_J\big) \notin \mathcal{K}_I$.

2 Refutation

2.1 Symmetric Encryption Specification

The assumption that if B uses a different key $K_{BS} \neq K_I$ that the one used for encryption of T_I by the intruder the decrypted output will be garbage (random-looking bytes) is questionable and depend a lot of the code implementation of the protocol. In fact, if the implementation uses an authenticated mode, like AES-GCM or ChaCha20-Poly135, decryption will fail completely. But if the implementation uses AES-CBC it won't throw an error and B will get nonsense plaintext.

Moreover if I uses the wrong cipher configuration such ad AES-ECB instead of AES-CB or AES-GCM, B will end up with completely unreadable results, and if the padding is incorrect (for example, PKCS7 was expected but not applied), B will likely encounter padding errors or corrupted trailing bytes.

Finally, if I uses the wrong Initial Vector (IV) or nonce, the first block (or few blocks) of decrypted data will be corrupted. The rest might still decrypt correctly depending on the mode (e.g., CBC mode propagates errors, but CTR mode only corrupts corresponding parts)

Since the protocol design project did not require us to specify the actual implementation of the protocol but only its theoretical aspect, the choice of the encryption scheme cannot be considered a point of attack; we can assume the encryption scheme to be flawless. For example in the Needham-Schroeder template, there is no mention of the encryption scheme.

2.2 Litterature

In cryptographic literature, and particularly in theoretical descriptions of protocols (for key exchange, authentication, etc.), there are no specifications regarding the choice of implementation or hardware aspects. For example, in *Applied Cryptography, Second Edition* by Bruce Schneier (translated by

Laurent Viennot), in the section 'Authentication and Key Exchange,' the theoretical description of the Needham–Schroeder protocol does not mention the choice of the symmetric encryption algorithm, nor the size used to encode the participants' identities. (*See 1 in Appendices A*)

2.3 Timestamp

In this attack scenario, the main objective of I is to find a symmetric key K_I such that for a fixed $M = \langle X', Y', Z', \Sigma' \rangle \in \mathcal{M}$

$$dec(enc(\langle X', Y', Z', \Sigma' \rangle, K_I), K_{BS}) = \langle A, Y, Z, \Sigma \rangle$$

i.e.

$$enc(\langle X', Y', Z', \Sigma' \rangle, K_I) = enc(\langle A, Y, Z, \Sigma \rangle, K_{BS})$$

In your attack you omit an important point: B process a check on the timestamp τ. That is why Y must be equal to a valid timestamp (coded on 64 bit to avoid the *Year 2038 problem*) and Z must be a valid lifetime. This greatly complicates the brute-force attack. The intruder must not be content with $X = A$ but must (indirectly) generate a timestamp $Y = \tau$ and a lifetime $Z = \lambda$ such that $\tau > t - \lambda$ where t is the time when B will get T_I.

Let us estimate the computational complexity of this brute-force attack. Assuming AES-128 encryption (128-bit keys), the keyspace contains 2^{128} possible keys. For a random key K_I, the probability that decrypting with K_{BS} produces values satisfying all constraints is the product of individual probabilities:

- Probability that $[dec(T_I, K_{BS})]_1 = A$: $P(\text{"}X = A\text{"}) = \frac{1}{2^1} = \frac{1}{2}$
- Probability that $[dec(T_I, K_{BS})]_2 = \tau$ is a valid timestamp within ± 1 hour window: $P(\text{"}Y = \tau\text{"}) \simeq \frac{2^{12}}{2^{64}}$
- Probability that $[dec(T_I, K_{BS})]_3 = \lambda$ is a valid lifetime (assuming 32-bit encoding with 2^{10} valid values): $P(\text{"}Z = \lambda\text{"}) \simeq \frac{2^{10}}{2^{32}}$

The combined probability is approximately:

$$\frac{1}{2} \times \frac{2^{12}}{2^{64}} \times \frac{2^{10}}{2^{32}} = 2^{-75}$$

Therefore, the expected number of candidate keys K_I that satisfy all constraints is:

$$2^{128} \times 2^{-75} = 2^{53} \text{ keys}$$

Even in the worst case where A (and this choice of yours is questionable) is encoded on only 1 bit, the intruder would need to test: 2^{53} keys at 10^9 operations per second $\simeq \frac{9\times10^{15}}{10^9} = 9 \times 10^6$ seconds $\simeq 104$ days.

2.4 Bruteforce Issues

Crucially, since $K_{BS} \notin K_I$, I cannot verify locally whether a candidate K_I produces the correct decryption. Instead, I must test each of the 2^{53} candidate keys through interaction with B:

- For each candidate K_I, craft T_I
- Send $M_{1,A}$ and T_I to B over the network
- Observe B's response to determine if the decryption was valid

Assuming an optimistic round-trip time of 10 ms per attempt, testing 2^{53} candidates would require :

$$2^{53} \times 0.01 \text{ s } \simeq 9 \times 10^{13} \text{ s } \simeq 2.9 \text{ million years}$$

This estimate assumes:

- Perfect network conditions with no rate limiting
- B processes all attempts without detection or blocking
- The intruder can maintain continuous communication for million of years

In practice, the attack is completely infeasible because:

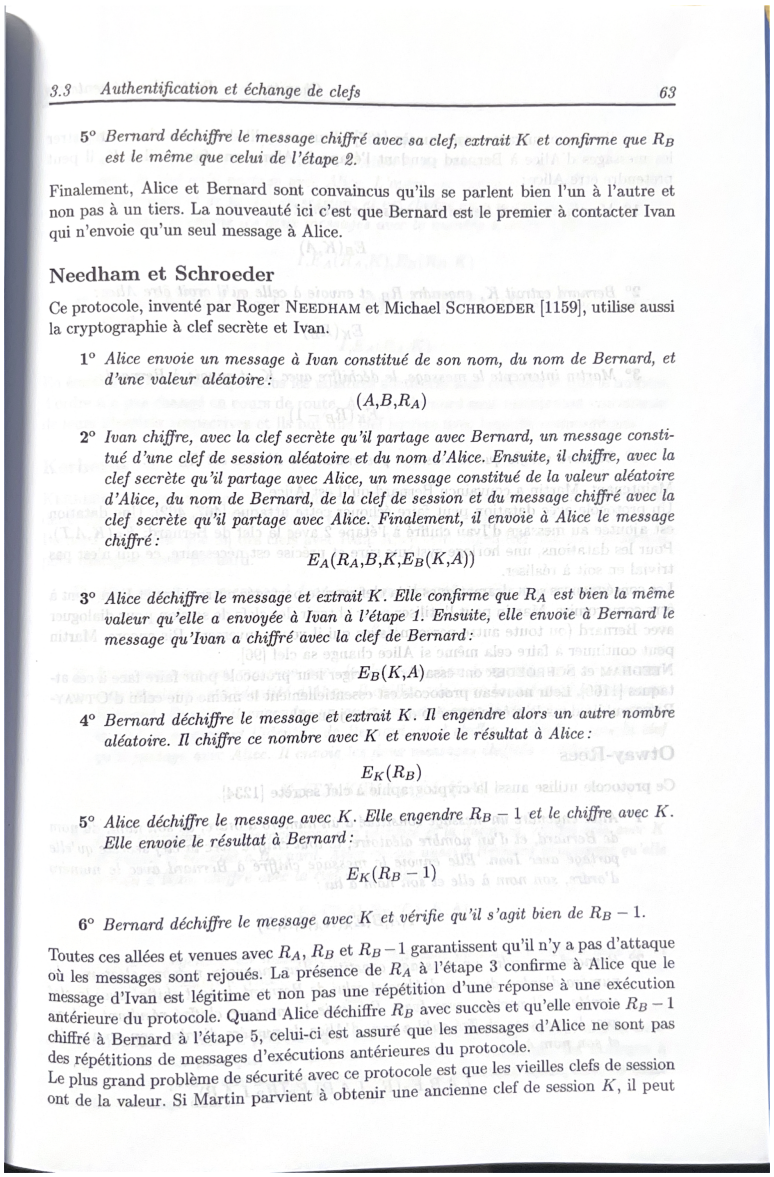
- The timestamp constraint means candidates must be retested as time progresses
- Network's rate limit would increase the time by order of magnitude
- Finally the intruder is not *Dracula*, he will not live forever ...

3 Conclusion

We refuse this attack, but we are ready to face another one.

Appendices

A Theoretical Needham-Schroeder



3.3 *Authentification et échange de clefs* 63

5° *Bernard déchiffre le message chiffré avec sa clef, extrait K et confirme que R_B est le même que celui de l'étape 2.*

Finalement, Alice et Bernard sont convaincus qu'ils se parlent bien l'un à l'autre et non pas à un tiers. La nouveauté ici c'est que Bernard est le premier à contacter Ivan qui n'envoie qu'un seul message à Alice.

Needham et Schroeder

Ce protocole, inventé par Roger NEEDHAM et Michael SCHROEDER [1159], utilise aussi la cryptographie à clef secrète et Ivan.

1° *Alice envoie un message à Ivan constitué de son nom, du nom de Bernard, et d'une valeur aléatoire :*

$$(A,B,R_A)$$

2° *Ivan chiffre, avec la clef secrète qu'il partage avec Bernard, un message constitué d'une clef de session aléatoire et du nom d'Alice. Ensuite, il chiffre, avec la clef secrète qu'il partage avec Alice, un message constitué de la valeur aléatoire d'Alice, du nom de Bernard, de la clef de session et du message chiffré avec la clef secrète qu'il partage avec Alice. Finalement, il envoie à Alice le message chiffré :*

$$E_A(R_A,B,K,E_B(K,A))$$

3° *Alice déchiffre le message et extrait K. Elle confirme que R_A est bien la même valeur qu'elle a envoyée à Ivan à l'étape 1. Ensuite, elle envoie à Bernard le message qu'Ivan a chiffré avec la clef de Bernard :*

$$E_B(K,A)$$

4° *Bernard déchiffre le message et extrait K. Il engendre alors un autre nombre aléatoire. Il chiffre ce nombre avec K et envoie le résultat à Alice :*

$$E_K(R_B)$$

5° *Alice déchiffre le message avec K. Elle engendre $R_B - 1$ et le chiffre avec K. Elle envoie le résultat à Bernard :*

$$E_K(R_B - 1)$$

6° *Bernard déchiffre le message avec K et vérifie qu'il s'agit bien de $R_B - 1$.*

Toutes ces allées et venues avec R_A, R_B et $R_B - 1$ garantissent qu'il n'y a pas d'attaque où les messages sont rejoués. La présence de R_A à l'étape 3 confirme à Alice que le message d'Ivan est légitime et non pas une répétition d'une réponse à une exécution antérieure du protocole. Quand Alice déchiffre R_B avec succès et qu'elle envoie $R_B - 1$ chiffré à Bernard à l'étape 5, celui-ci est assuré que les messages d'Alice ne sont pas des répétitions de messages d'exécutions antérieures du protocole.
Le plus grand problème de sécurité avec ce protocole est que les vieilles clefs de session ont de la valeur. Si Martin parvient à obtenir une ancienne clef de session K, il peut

Figure 1: NS in *Applied Cryptography* by Bruce Schneier