ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Beyond eCK: Security against Stronger Adversaries

Michèle Feltz

Joint work with Cas Cremers

Authenticated Key Exchange (AKE) Protocols

- An **AKE protocol establishes a shared session-key between two agents** using asymmetric (public key) cryptography
  $\Longrightarrow$ further communication protected using session-key
- Security analysis in game-based security models:
  - **Adversary:** full control of the network, may learn long-term secret keys or session-specific values
  - **Security goal:** Adversary should not be able to distinguish the real session-key from a random one

Perfect Forward Secrecy (PFS)

We are interested in the following security property:

Perfect Forward Secrecy: secrecy of *past* session-keys even if long-term secret keys are compromised

Challenge: Can 2-message AKE protocols achieve PFS even under disclosure of session-specific values and the actor's long-term secret keys?

Diffie-Hellman type AKE protocol

$G = \langle g \rangle$ cyclic group of prime order q

$$
\begin{array}{ccc}
A & & B \\
 & \xrightarrow{\quad g^x \quad} & \\
 & \xleftarrow{\quad g^y \quad} & \\
\end{array}
$$

$K_{AB} = F(g^y, x, PK_B, SK_A)$ $K_{BA} = F(g^x, y, PK_A, SK_B)$

Perfect Forward Secrecy Attack [Krawczyk05]

1. The adversary E impersonates A to B:

E | B

$E \xrightarrow{g^x} B$

$E \xleftarrow{g^y} B$

$K_{AB} = F(g^y, x, PK_B, SK_A)$ | $K_{BA} = F(g^x, y, PK_A, SK_B)$

2. E corrupts A, hence learning SK_A
3. E can compute $K_{AB} = F(g^y, x, PK_B, SK_A)$ ↯

⇝ Motivated the introduction of **weak-PFS**!

Can we achieve PFS in 2-message AKE protocols?

- "No 2-message protocol, and in particular HMQV, can provide full perfect forward secrecy." [Krawczyk05]
- "No 2-round AKE protocol can achieve perfect forward secrecy." [LaMaccia-Lauter-Mityagin06]
- No "..., the eCK model is currently regarded as the strongest security model." (weak-PFS) [Lee-Park08]

Can we achieve PFS in 2-message AKE protocols?

- “No 2-message protocol, and in particular HMQV, can provide full perfect forward secrecy.” [Krawczyk05]
- “No 2-round AKE protocol can achieve perfect forward secrecy.” [LaMaccia-Lauter-Mityagin06]
- No “..., the eCK model is currently regarded as the strongest security model.” (weak-PFS) [Lee-Park08]
- **Yes, we can! [F-Cremers12]**

Contributions of our work

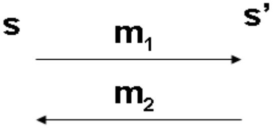
1. Formalization of two new game-based security models:
   - eCK^w: precisely modeling weak PFS
   - eCK-PFS: integrating PFS into eCK^w
     $\rightarrow$ strongest security model so far!
2. SIG: Generic transformation from eCK^w to eCK-PFS
3. Application of SIG to the NAXOS protocol

   $\rightsquigarrow$ Goal reached! There is a 2-message KE protocol that achieves PFS in the presence of a strong active adversary!

Concepts for Relating Sessions

Origin-session:	• session where message originates from • message not modified or injected by adversary • weak-PFS and PFS

Matching sessions:	• intended communication partners • based on matching conversations



Our New eCK-like Models: eCK^w and eCK-PFS

How We Capture weak-PFS and PFS

weak-PFS: compromise of long-term secret keys *after* the end of the test session under the condition that an origin-session for the test session exists

- passivity of adversary $\leftrightarrow$ existence of **origin-session**

PFS: compromise of long-term secret keys *after* the end of the test session

- irrespective of the existence of an origin-session

Our New eCK-like Models: eCK^w and eCK-PFS

Queries:

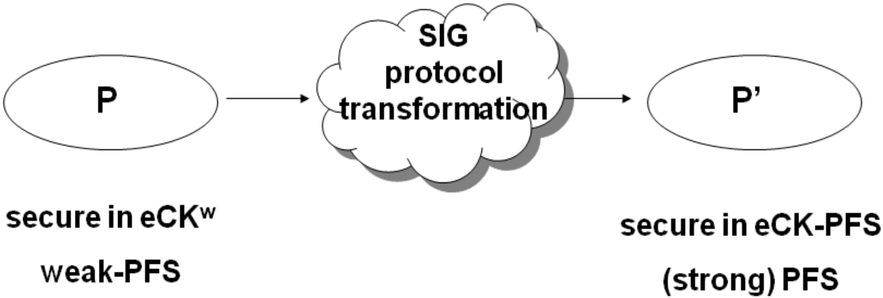
- Send(m, s): sends message m to session s
- LtkRev(A): learns long-term secrets of A
- SesskRev(s): learns session-key of s
- RandRev(s): learns random values of s

A completed session s is ***fresh*** if:

1. No SesskRev on session s or on its matching session
2. Not both LtkRev(actor) and RandRev(s)
3. Not both LtkRev(peer) and RandRev(origin-session of s)
4. If there is no origin-session, then no LtkRev(peer) **before the end of session s**

From eCK^w to eCK-PFS

P, P' **two-message** AKE protocols



Our SIG Transformation: Design Considerations

- Focus: 2-message Diffie-Hellman (DH) type key exchange protocols (e.g. TS2, HMQV, NAXOS, CMQV,...)
- SIG transformation: Sign your DH exponential g^z!
  - enforces existence of origin-session (i.e. prevents active attacks)
  - allows to achieve perfect forward secrecy (PFS)
- Flexibility: possible design trade-offs (e.g. sign identity of peer as well)

SIG: Generic Transformation from eCK^w to eCK-PFS

Let Π be the class of 2-message DH type KE protocols.

$A : (a, g^a), (sk_A, pk_A)$ $\qquad\qquad$ $B : (b, g^b), (sk_B, pk_B)$

$$\xrightarrow{g^x, Sign_A(g^x[,B])}$$

$$\xleftarrow{g^y, Sign_B(g^y[,g^x,A])}$$

e.g. $x \in_R \mathbb{Z}_p$ or $x = H(r, a)$ with $r \in_R \{0, 1\}^k$

Theorem

Assume: the signature scheme is deterministic and unforgeable.

$P \in \Pi$ **secure in eCK^w** $\Rightarrow$ $SIG(P)$ **secure in eCK-PFS**

Application of SIG to NAXOS

Proposition

NAXOS is secure in eCK^w.

Corollary

SIG(*NAXOS*) is secure in eCK-PFS.

$$A : (a, \underline{A} := g^a), (sk_A, pk_A) \qquad\qquad B : (b, \underline{B} := g^b), (sk_B, pk_B)$$

$$r_A \in_R \{0,1\}^k$$

$$X = g^{H_1(r_A, a)} \quad \xrightarrow{X, Sign_A(X[,B])}$$

$$r_B \in_R \{0,1\}^k$$

$$\xleftarrow{Y, Sign_B(Y[,X,A])} \quad X = g^{H_1(r_B, b)}$$

$$H_2(Y^a, \underline{B}^{H_1(r_A,a)}, Y^{H_1(r_A,a)}, A, B) \qquad\qquad H_2(\underline{A}^{H_1(r_B,b)}, X^b, X^{H_1(r_B,b)}, A, B)$$

Is MAC an Alternative?

The MAC transformation [Boyd-GonzalezNieto11]:

- uses a static Diffie-Hellman key as shared information between two agents
- is supposed to provide PFS independently from eCK security

SIG versus MAC transformation:

- eCK-PFS is stronger than eCK^w and PFS separately
- attack on MAC(NAXOS) in eCK-PFS

eCK-PFS stronger than eCK^w and PFS separately

Assume: No origin-session exists for the test session.

Let t denote the time when the test session ends.

eCK^w	PFS
LtkRev(actor) before or after t	LtkRev(actor) and LtkRev(peer) after t

$\rightsquigarrow$ ϕ := LtkRev(actor) before t and LtkRev(peer) after t

- ϕ neither captured in eCK^w nor in PFS
- BUT ϕ captured in eCK-PFS!

Attack on *MAC*(*NAXOS*) in eCK-PFS

Let $S_{AB} = g^{a'b'}$ denote the shared static DH key between A and B.

1. The adversary E issues the query LtkRev(B)
2. E impersonates A to B:

E $\qquad$ $B : (b, g^b), (b', g^{b'})$

$\xrightarrow{X, MAC_{S_{AB}}(A,B,X)}$

$\xleftarrow{Y, MAC_{S_{AB}}(B,A,Y)}$

3. E issues the query LtkRev(A)
4. E can compute the same session-key as B does (as in the PFS attack on NAXOS in eCK-PFS) ↯

Conclusion:

- Introduction of new security models eCK^{w} and eCK-PFS $\rightarrow$ eCK-PFS strongest security model so far!
- Generic transformation SIG from eCK^{w} to eCK-PFS
- PFS can be achieved in two-message AKE protocols even in the presence of a very strong adversary!