Key-schedule Security for the TLS 1.3 Standard

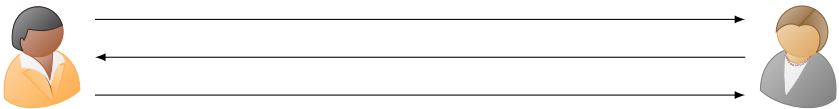
Chris Brzuska[1] Antoine Delignat-Lavaud[2] **Christoph Egger**[3]
Cédric Fournet[2] Konrad Kohbrok[1] Markulf Kohlweiss[4]

[1]Aalto University, Finland
[2]Microsoft Research Cambridge, UK
[3]Université Paris Cité, CNRS, IRIF, France
[4]University of Edinburgh, UK

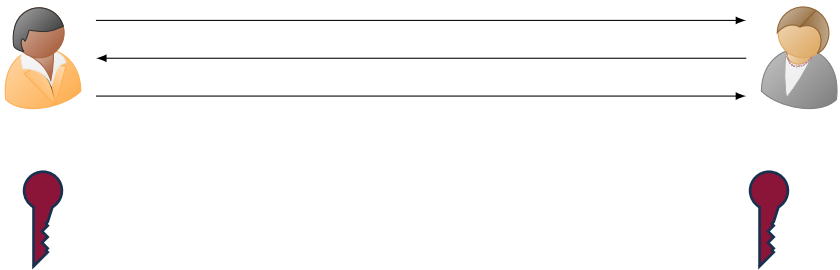
Asiacrypt 2022, Dec. 6, 2022

What is Key Exchange?

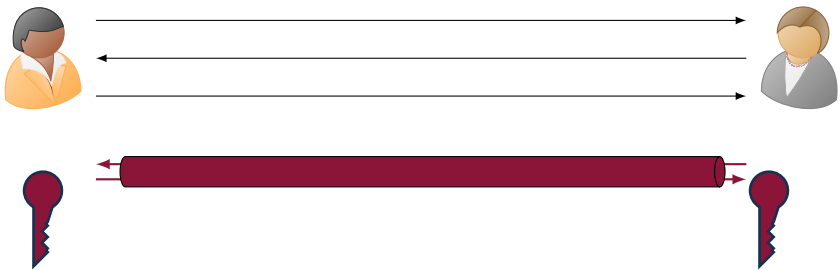
What is Key Exchange?



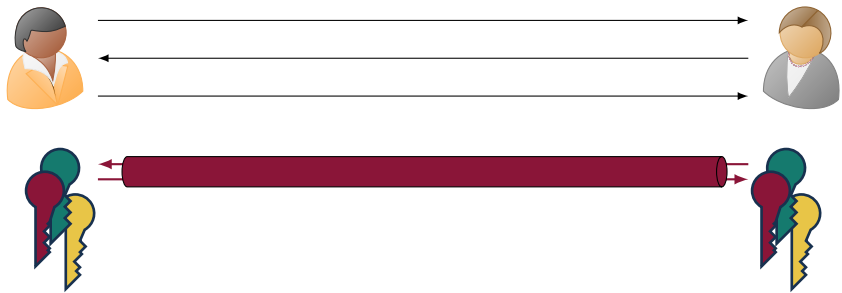
What is Key Exchange?



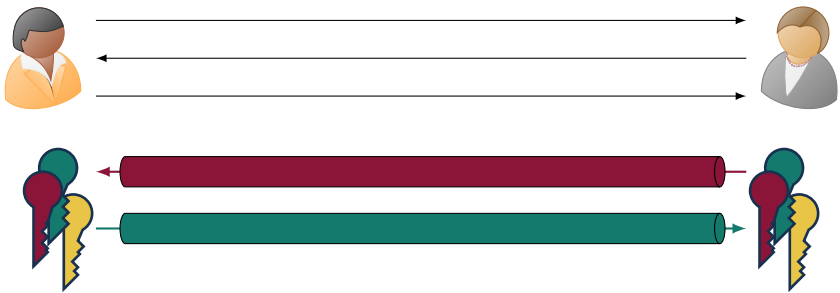
What is Key Exchange?



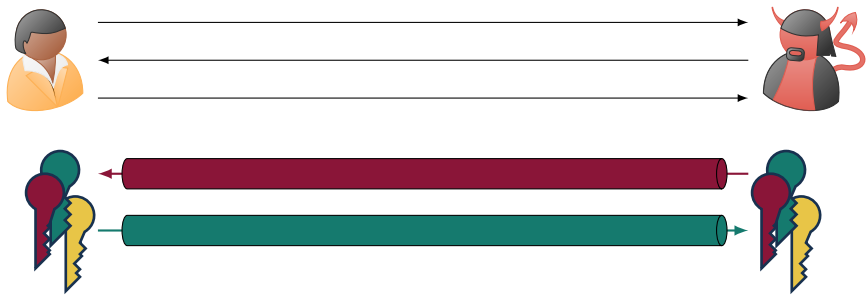
What is Key Exchange?



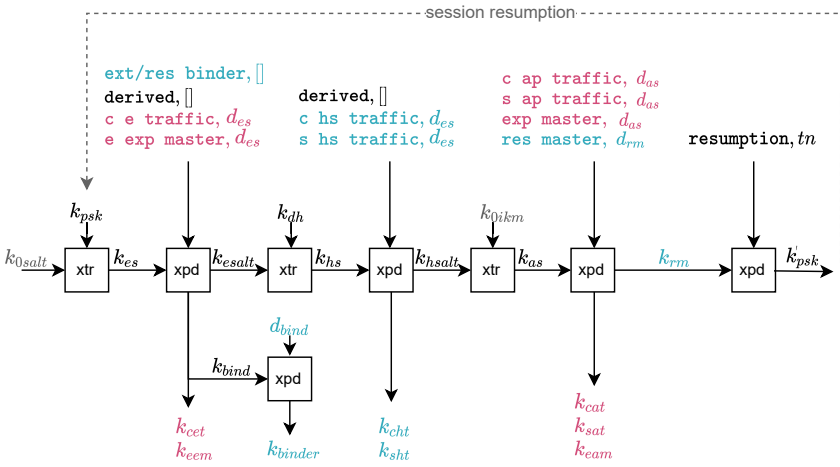
What is Key Exchange?



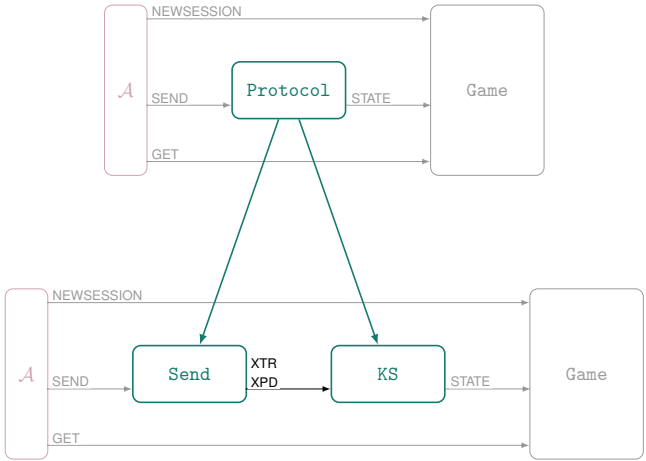
What is Key Exchange?



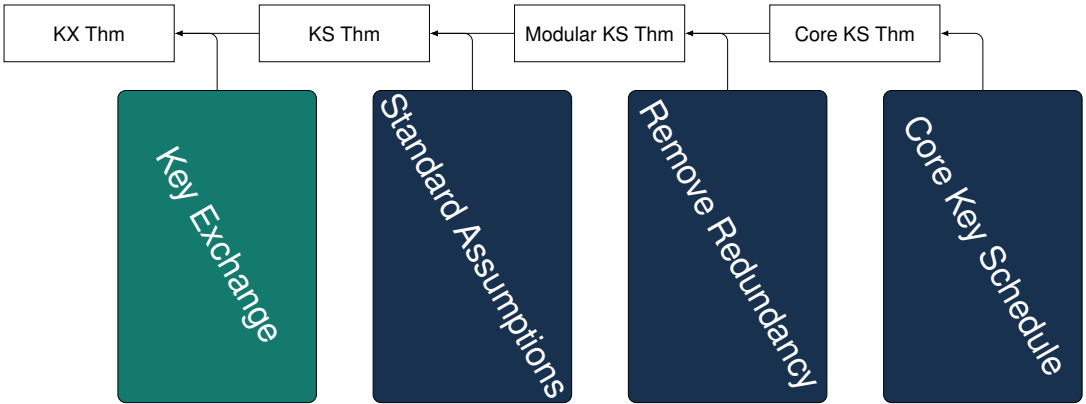
What is a Key Schedule



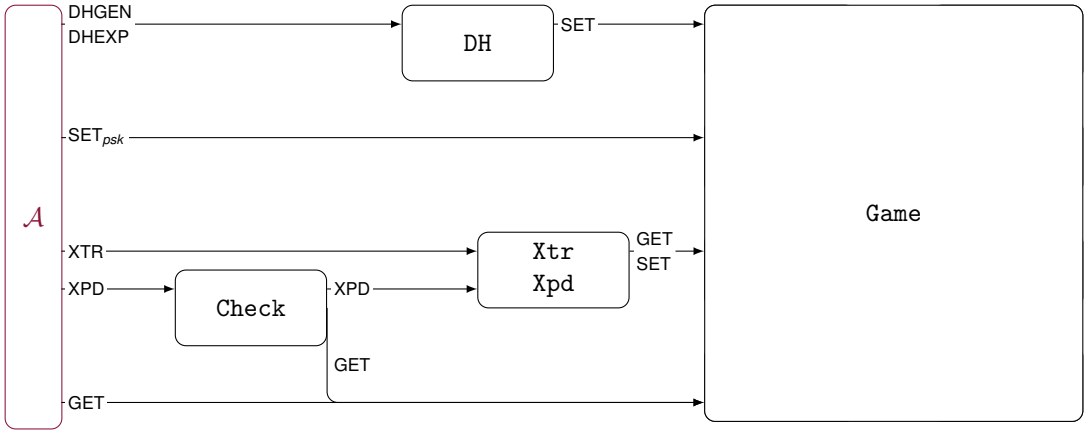
What is a Key Schedule



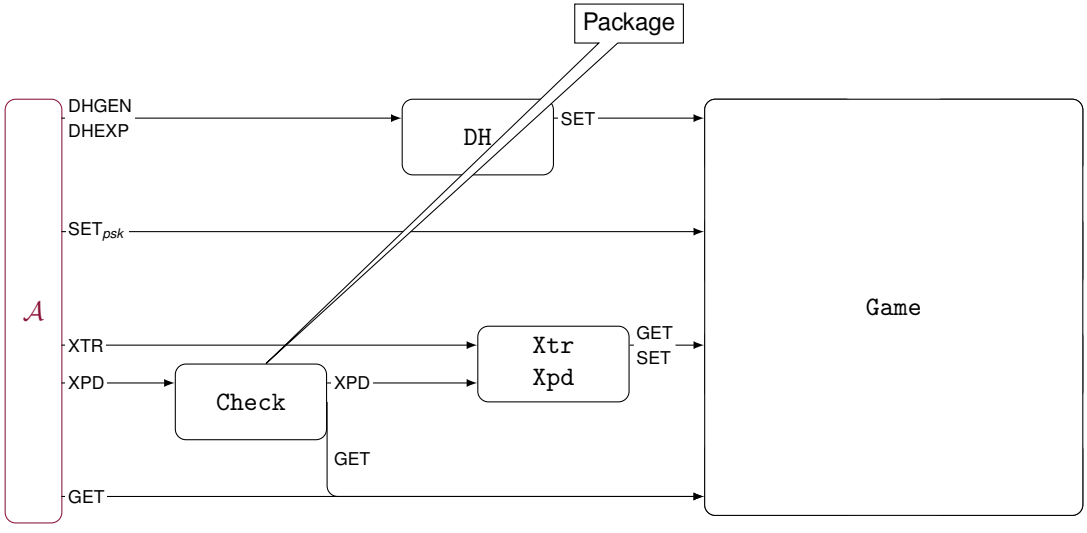
Proof Structure for TLS



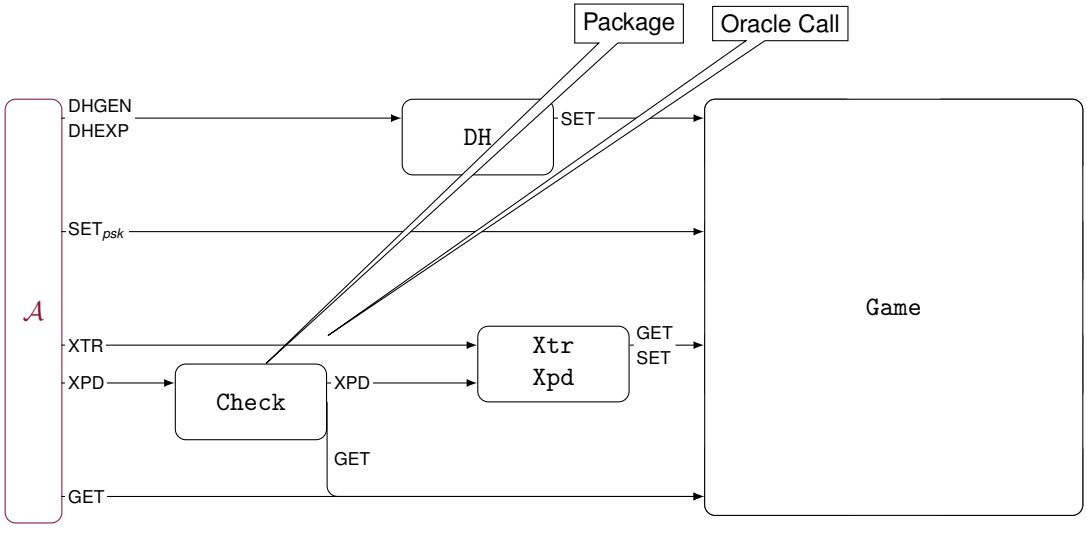
Key Schedule Model



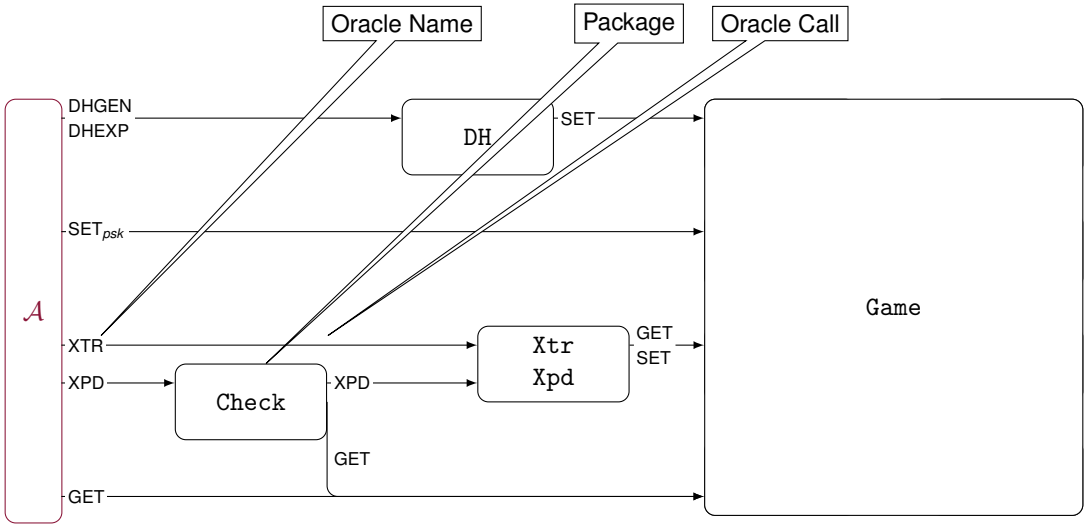
Key Schedule Model



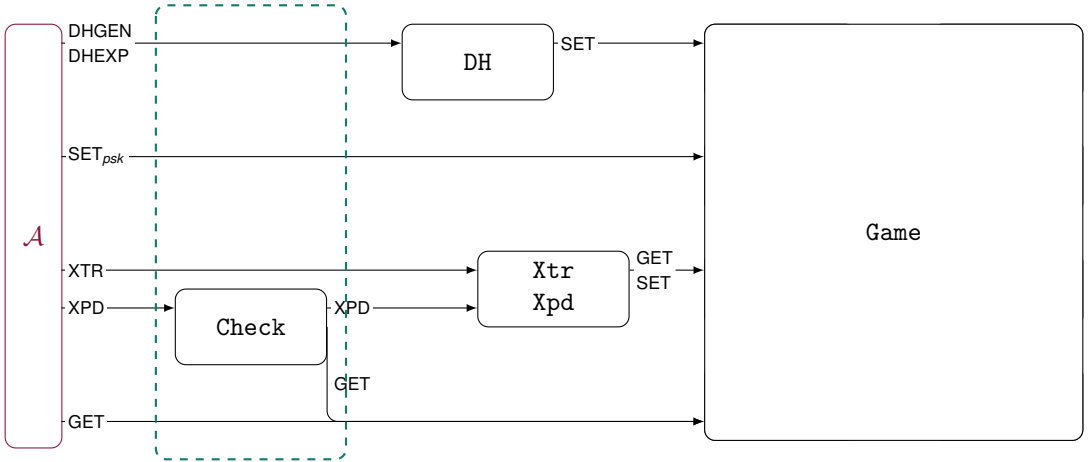
Key Schedule Model



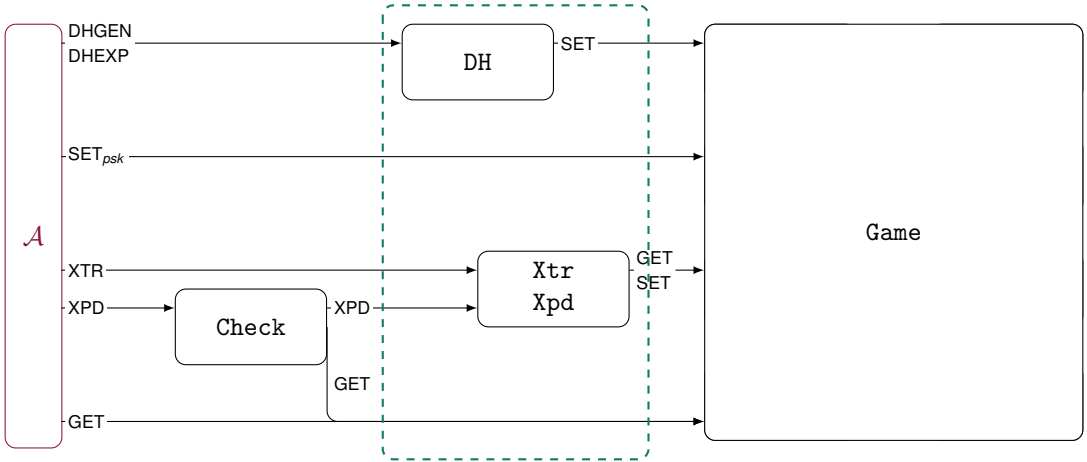
Key Schedule Model



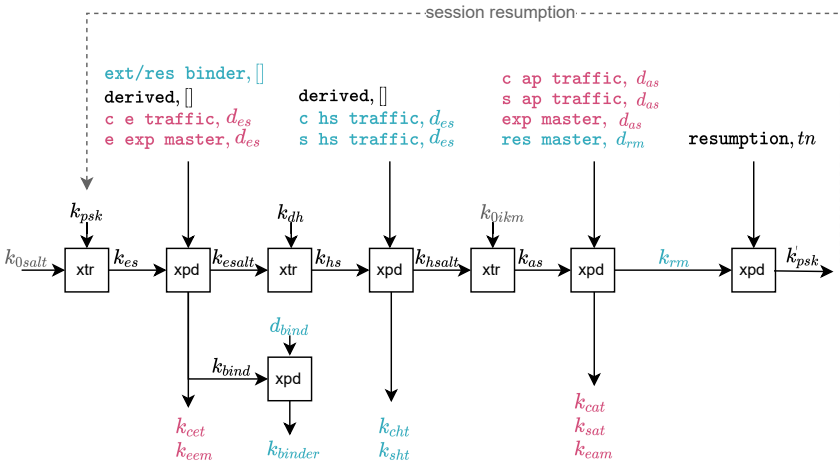
Key Schedule Model



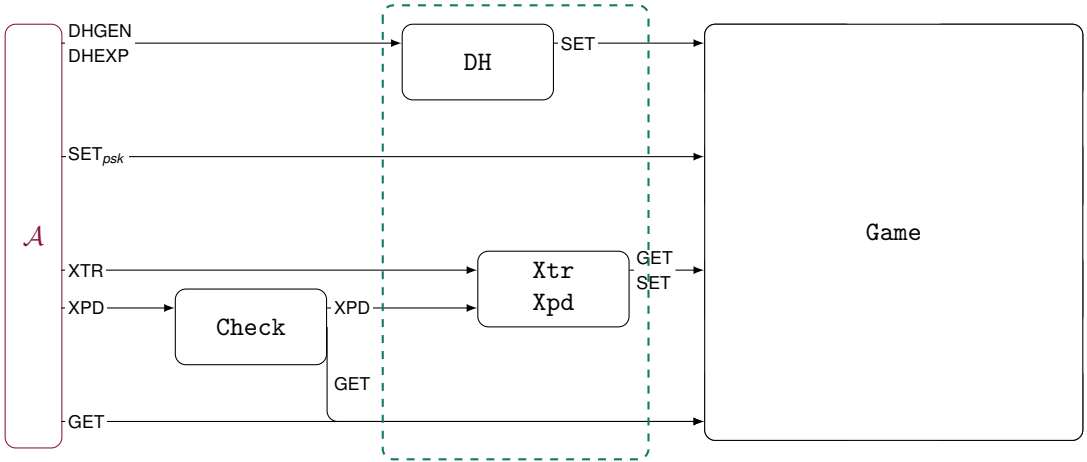
Key Schedule Model



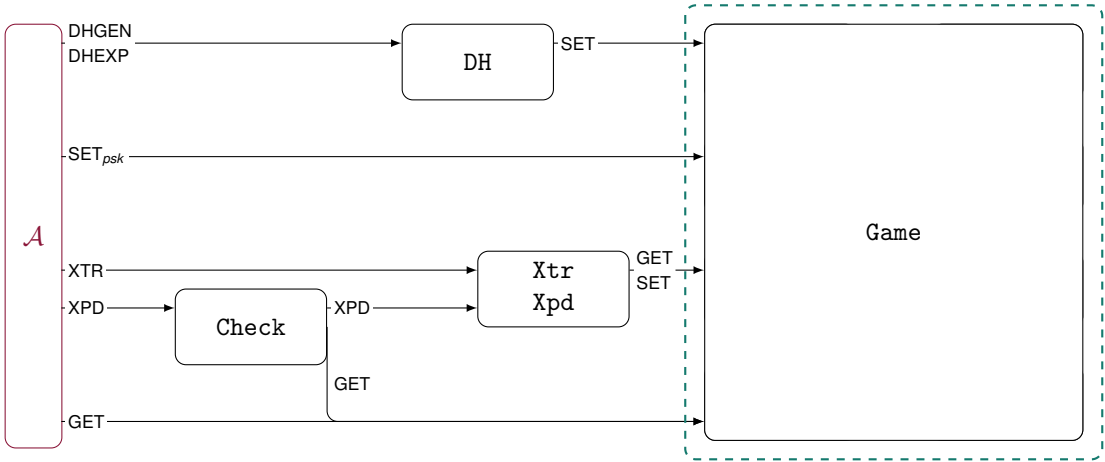
TLS 1.3 Key Schedule



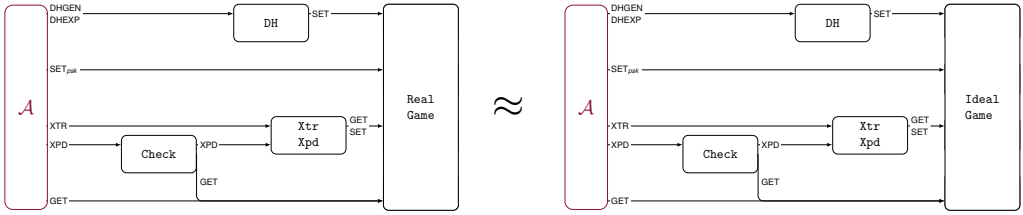
Key Schedule Model



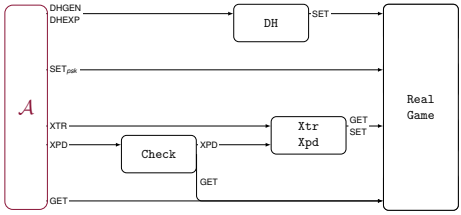
Key Schedule Model



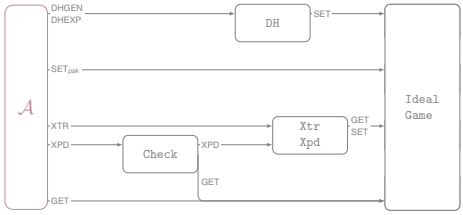
Simulation Based Security



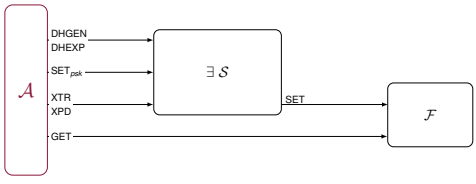
Simulation Based Security



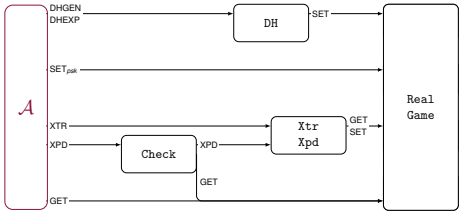
$\approx$



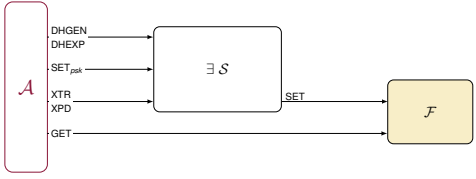
$\approx$

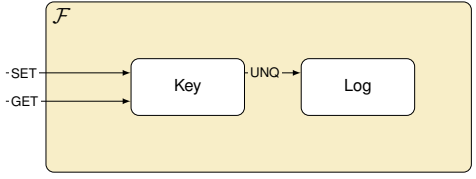


Simulation Based Security

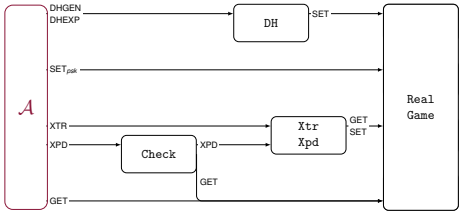


$\approx$

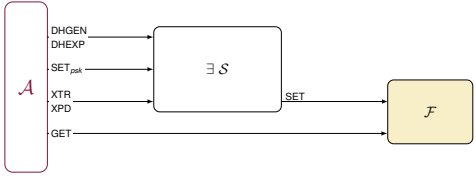




Simulation Based Security



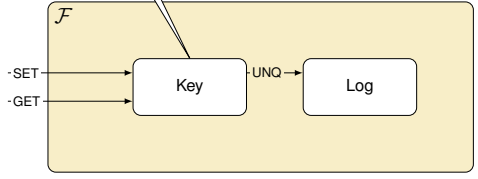
$\approx$



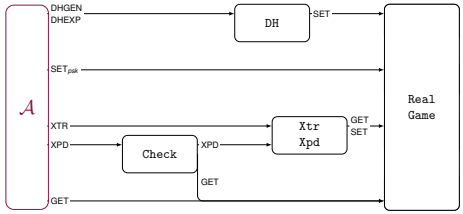
Pseudo-Randomness

if $hon = 1$

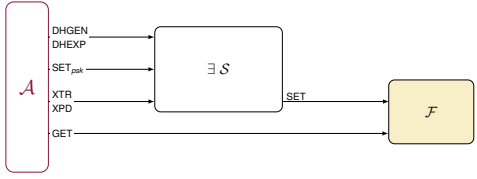
$k \leftarrow_{\$} \{0,1\}^{|k|}$



Simulation Based Security



$\approx$



Pseudo-Randomness

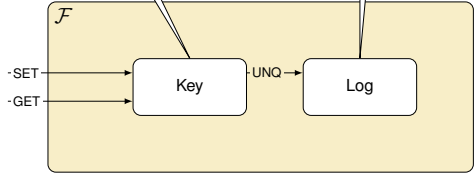
if $hon = 1$

$k \leftarrow_{\$} \{0,1\}^{|k|}$

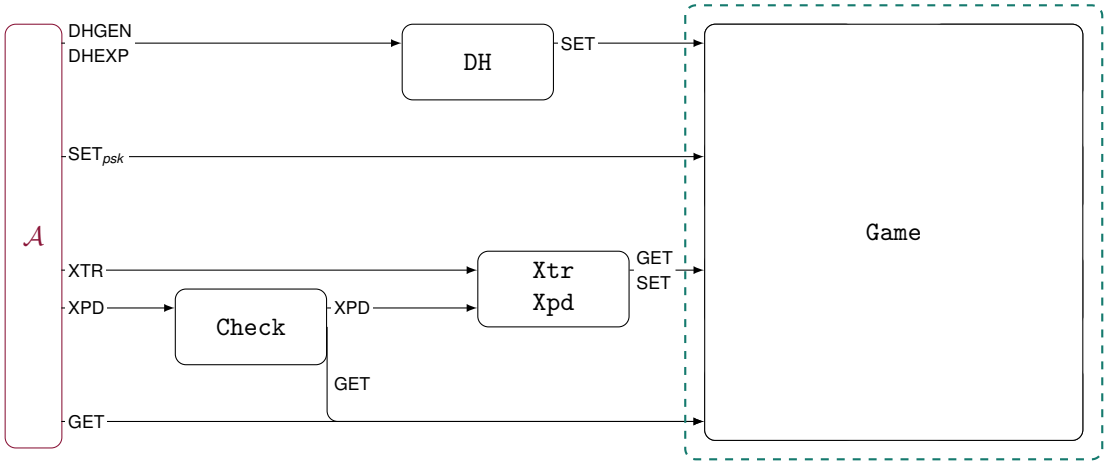
Key-Uniqueness

if $\exists h' : k = Log[h]$
$= Log[h']$

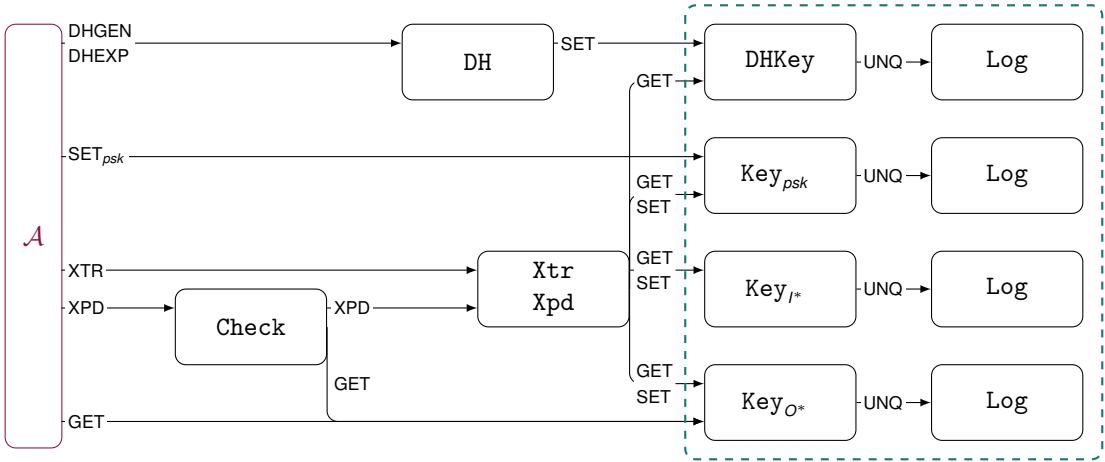
abort



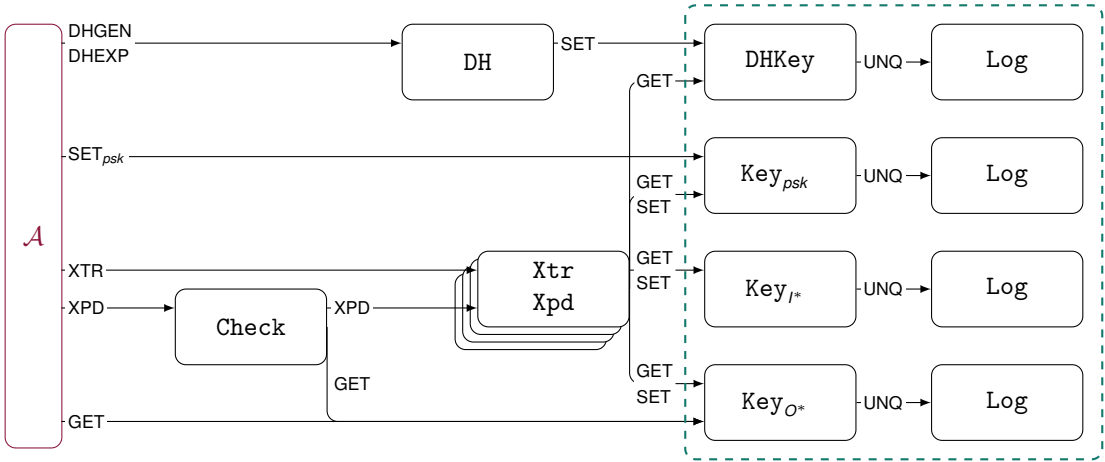
Key Schedule Model



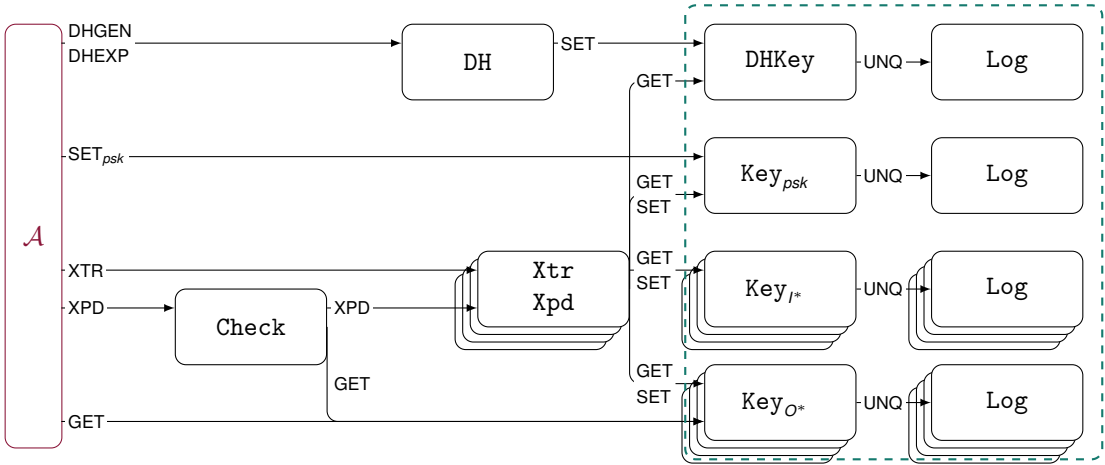
Key Schedule Model



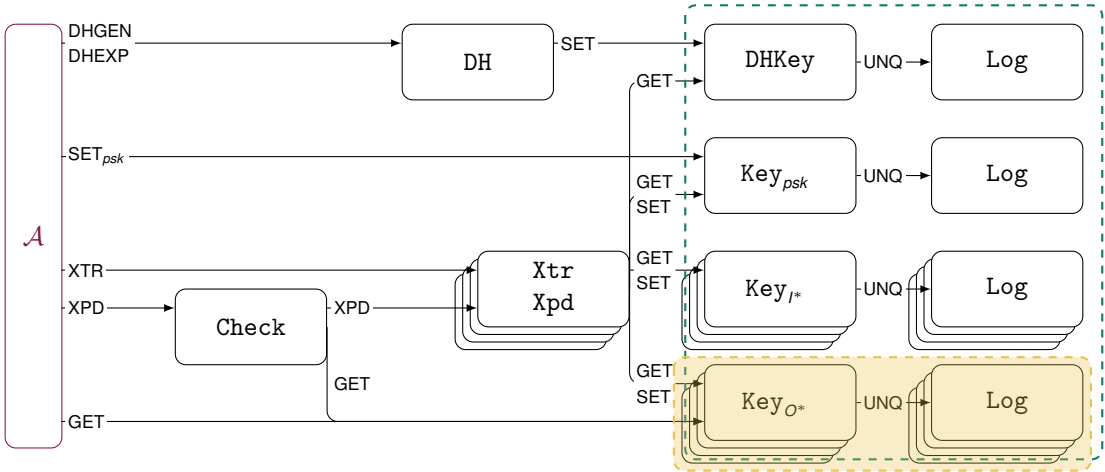
Key Schedule Model



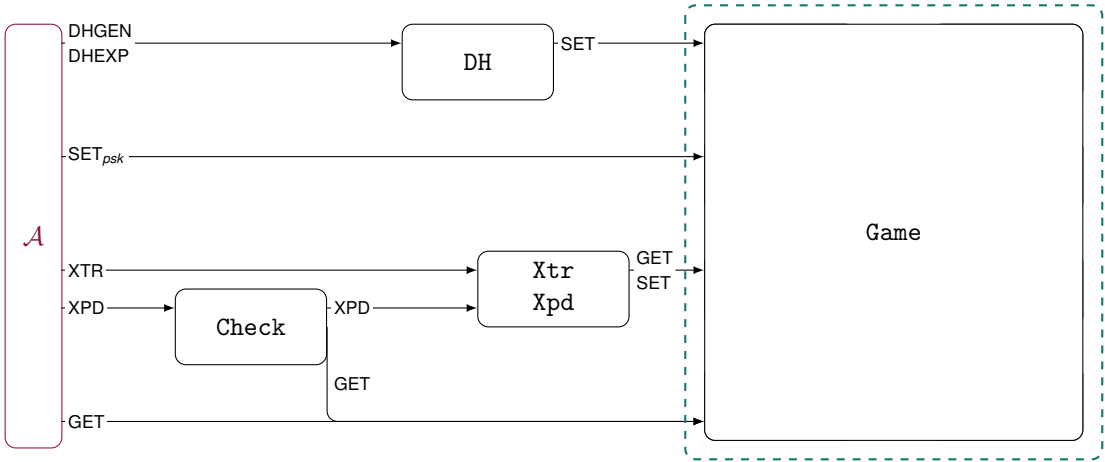
Key Schedule Model



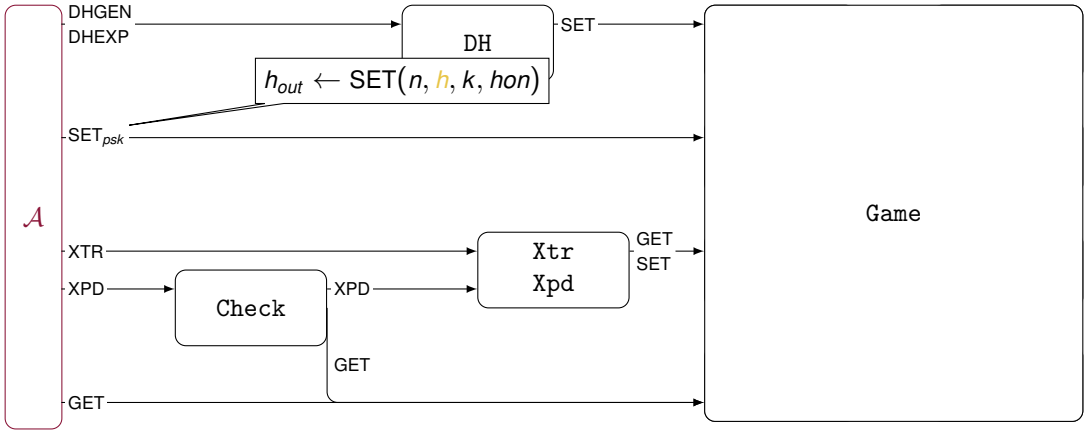
Key Schedule Model



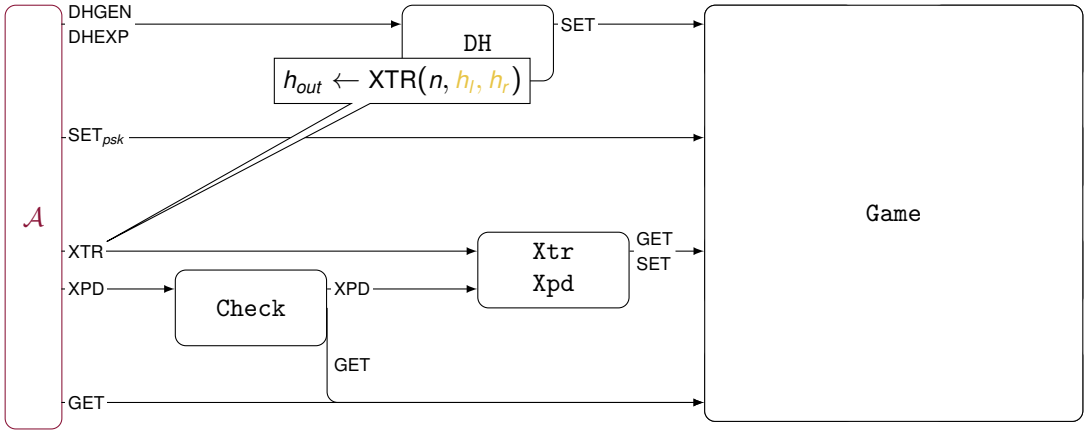
Key Schedule Model



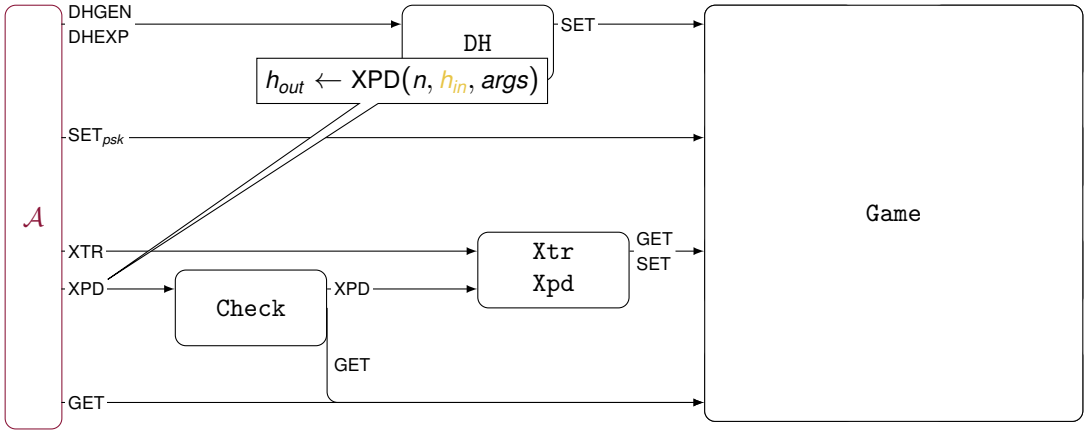
Key Schedule Model



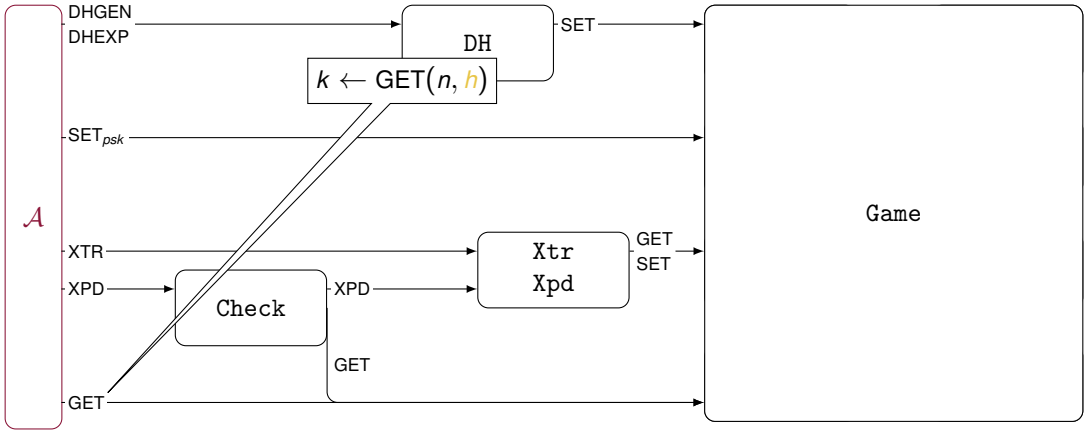
Key Schedule Model



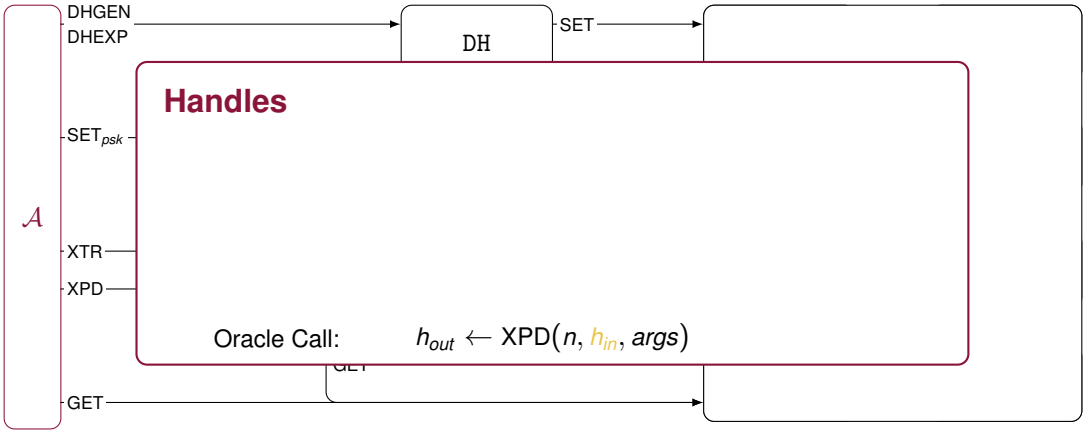
Key Schedule Model



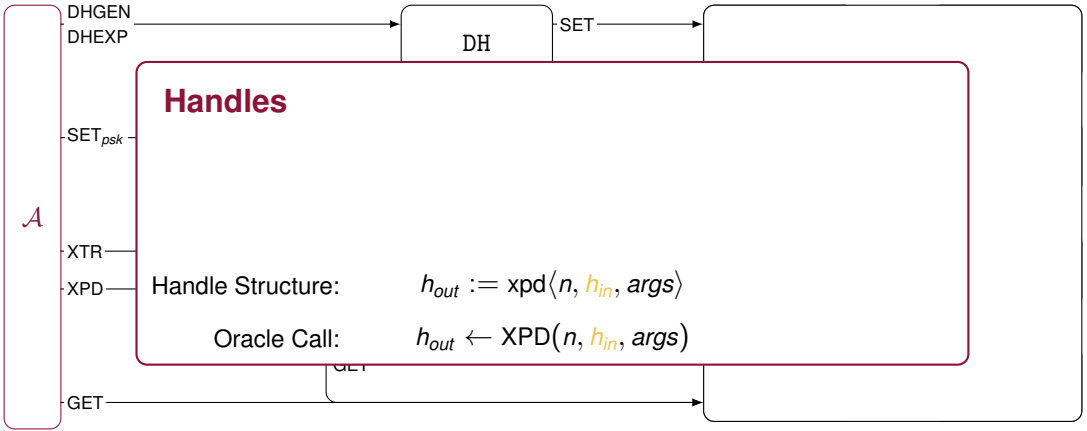
Key Schedule Model



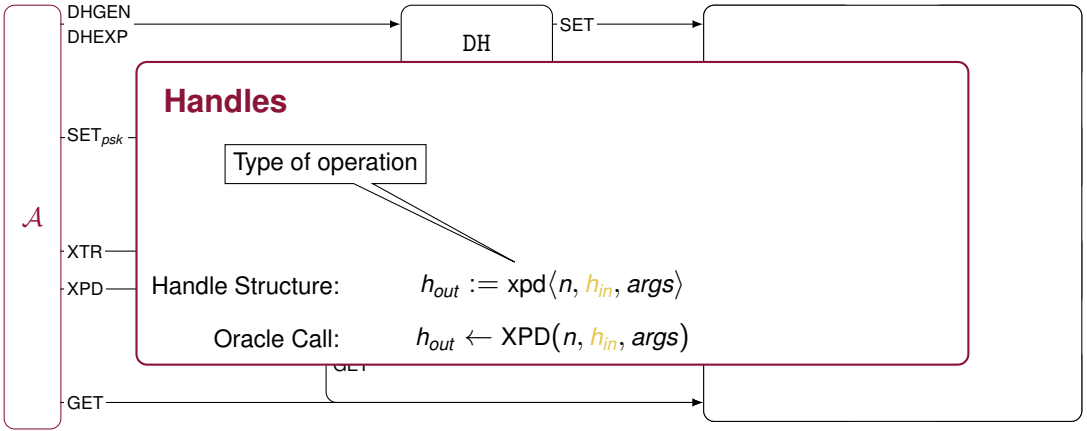
Key Schedule Model



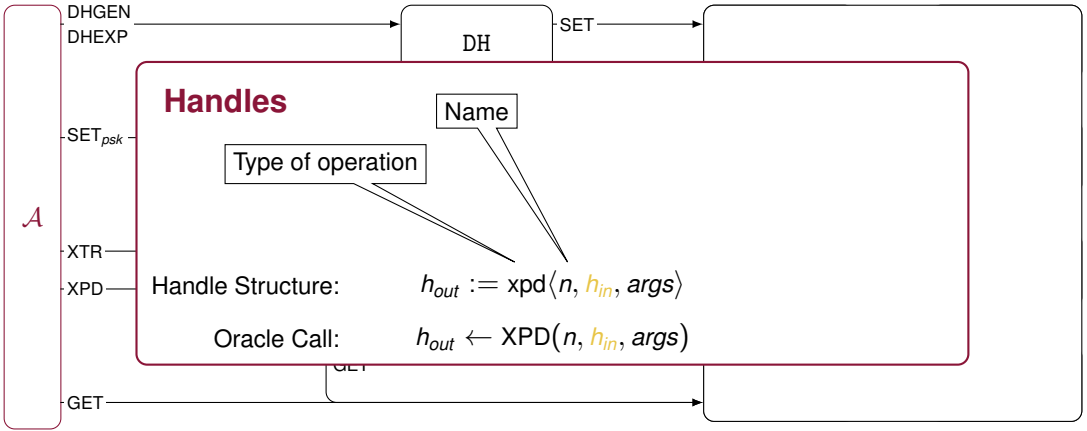
Key Schedule Model



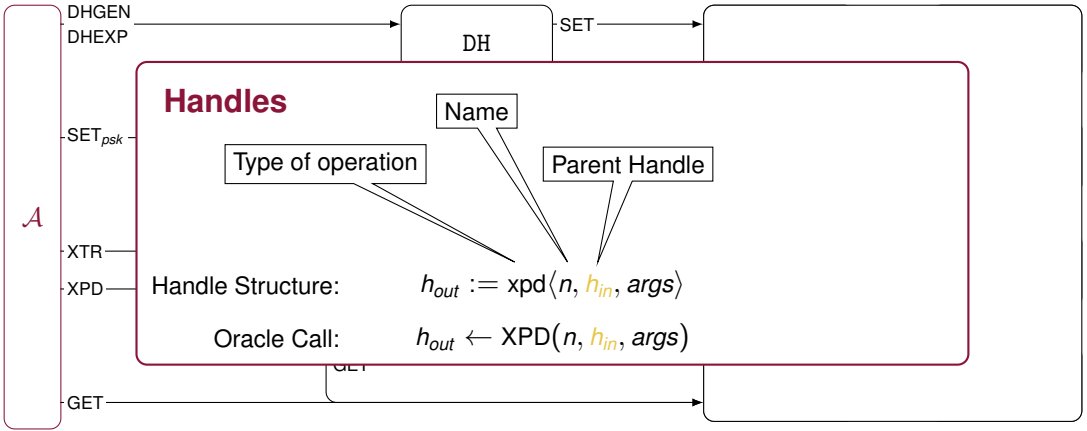
Key Schedule Model



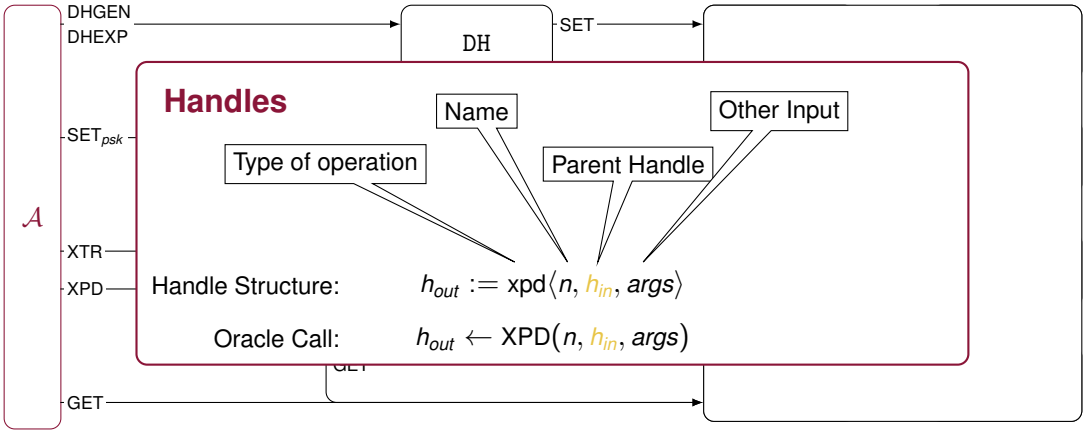
Key Schedule Model



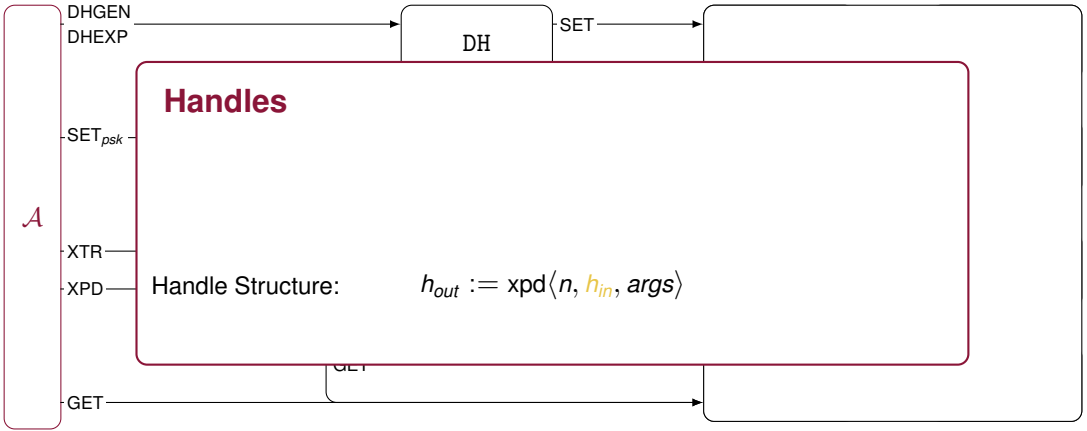
Key Schedule Model



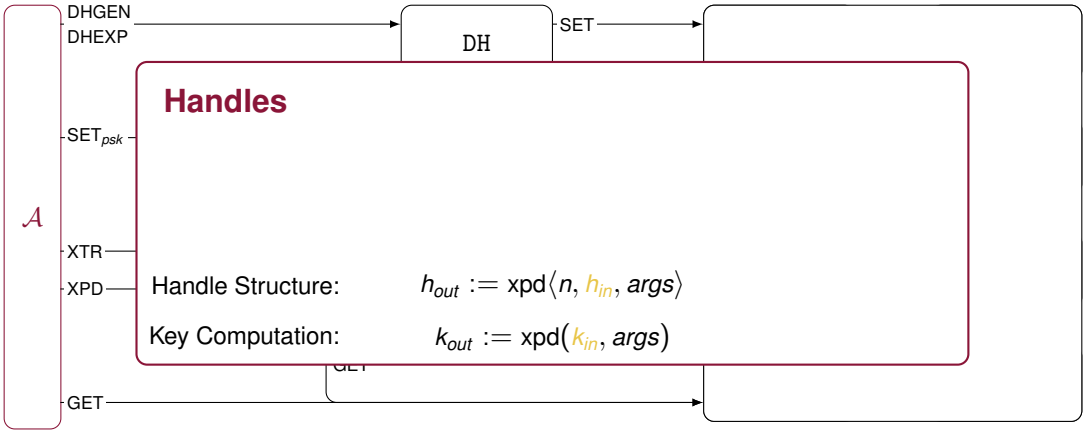
Key Schedule Model



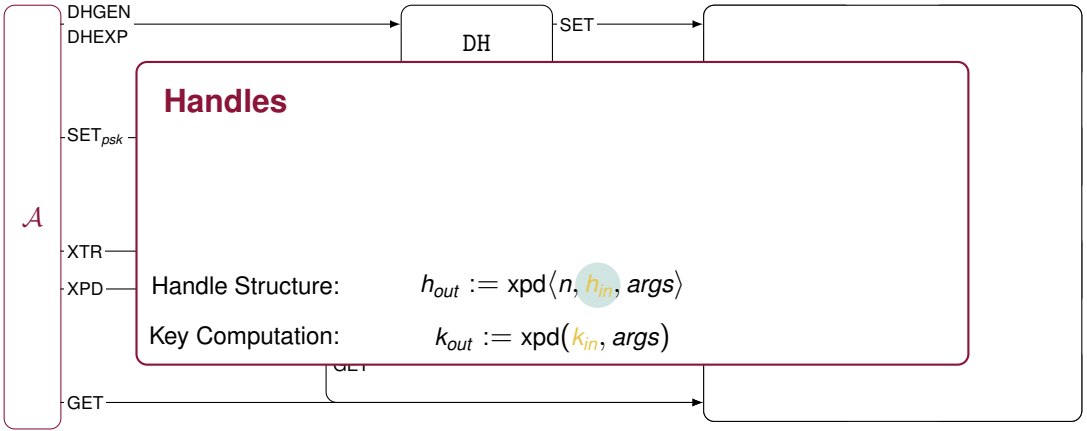
Key Schedule Model



Key Schedule Model



Key Schedule Model



Is Key-Schedule Analysis an Useful Tool

Is Key-Schedule Analysis an Useful Tool

- Key Schedule security can be proven
  - For TLS: *Key-schedule Security for the TLS 1.3 Standard*
    Brzuska, Delignat-Lavaud, Egger, Fournet, Kohbrok, Kohlweis – (This work)

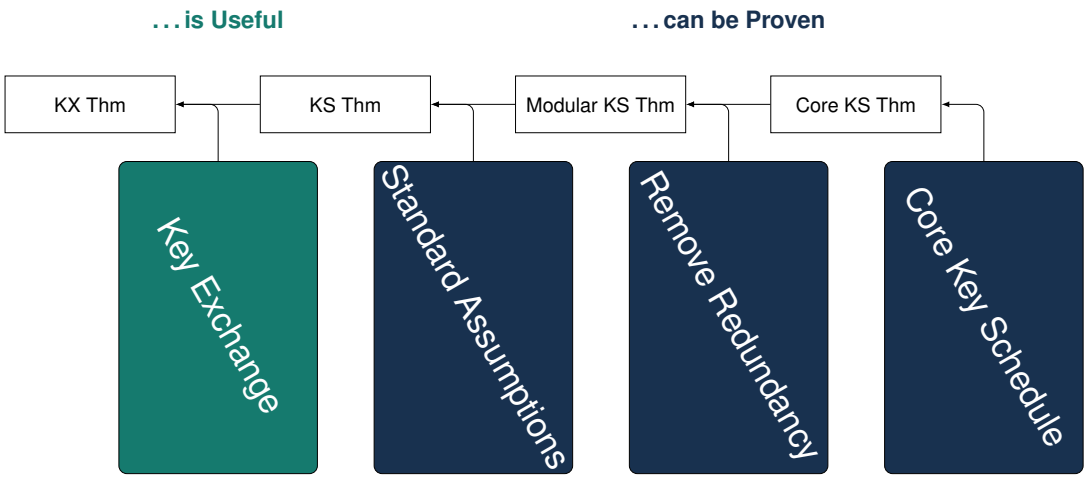
Is Key-Schedule Analysis an Useful Tool

- Key Schedule security can be proven
  - For TLS: *Key-schedule Security for the TLS 1.3 Standard*
    Brzuska, Delignat-Lavaud, Egger, Fournet, Kohbrok, Kohlweis – (This work)
  - For MLS: *Security Analysis of the MLS Key Derivation*
    Brzuska, Cornelissen, Kohbrok – IEEE S&P'22

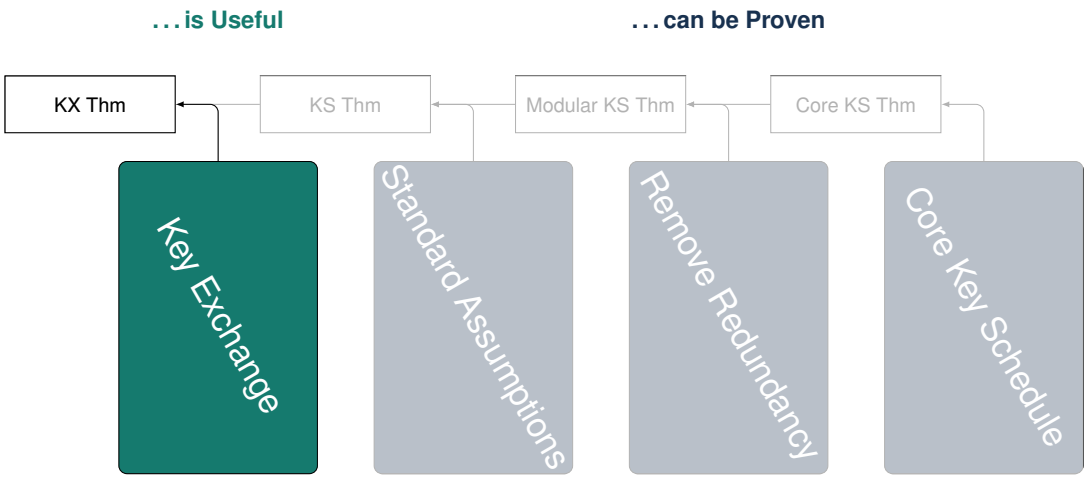
Is Key-Schedule Analysis an Useful Tool

- Key Schedule security can be proven
  - For TLS: *Key-schedule Security for the TLS 1.3 Standard*
    Brzuska, Delignat-Lavaud, Egger, Fournet, Kohbrok, Kohlweis – (This work)
  - For MLS: *Security Analysis of the MLS Key Derivation*
    Brzuska, Cornelissen, Kohbrok – IEEE S&P'22
- Useful to prove Key Exchange security:
  - For TLS: *Key Exchange to Key Schedule Reduction for TLS 1.3*
    Brzuska, Egger – Soon on ePrint

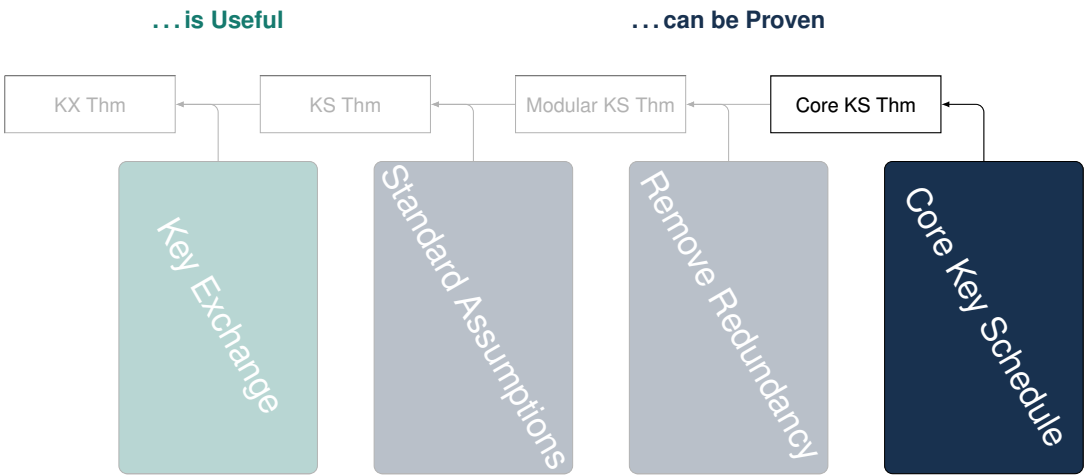
Proof Structure for TLS



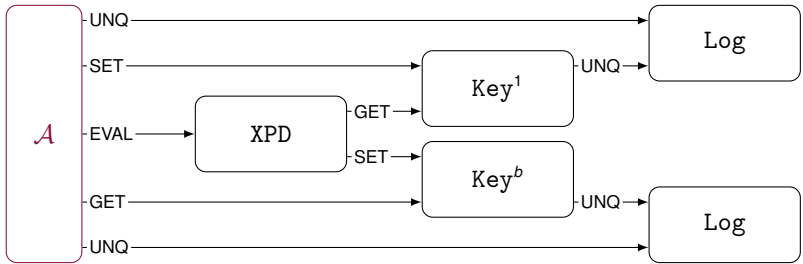
Proof Structure for TLS



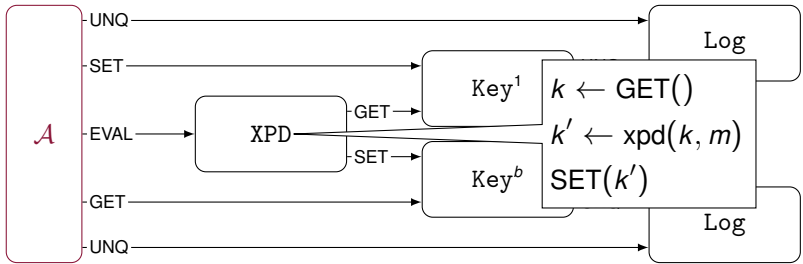
Proof Structure for TLS



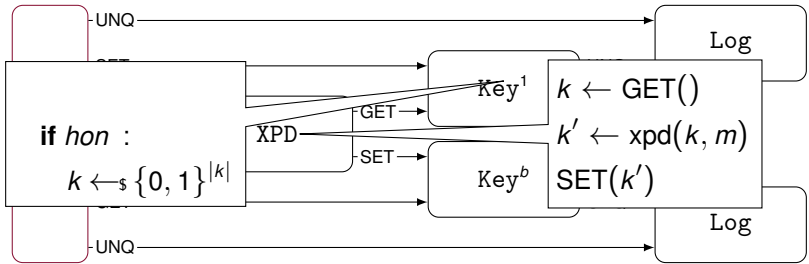
Core Key Schedule Proof



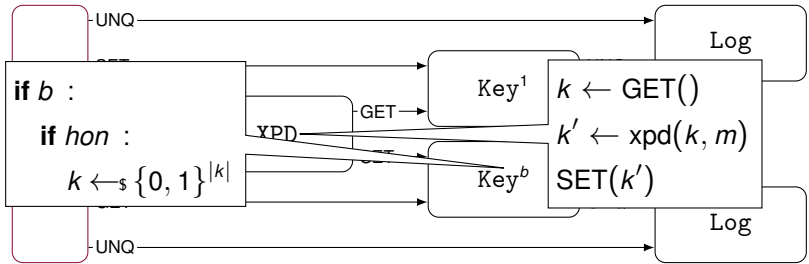
Core Key Schedule Proof



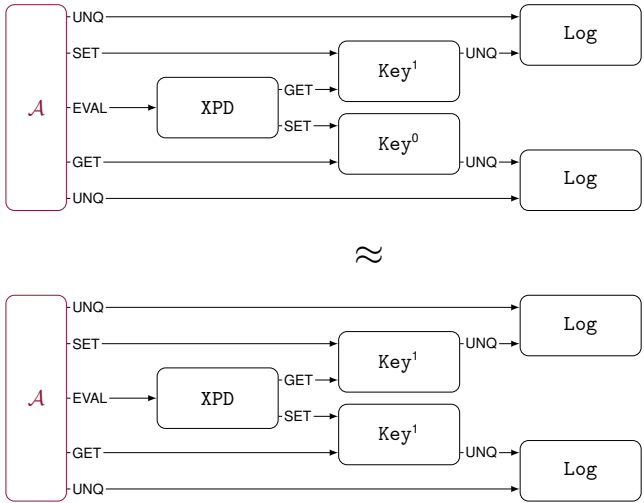
Core Key Schedule Proof



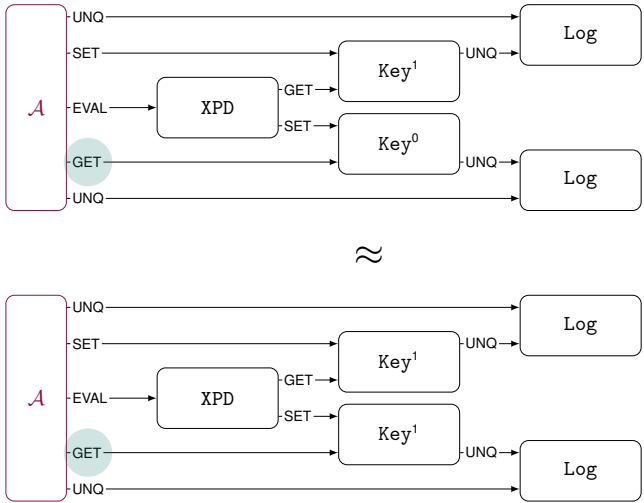
Core Key Schedule Proof



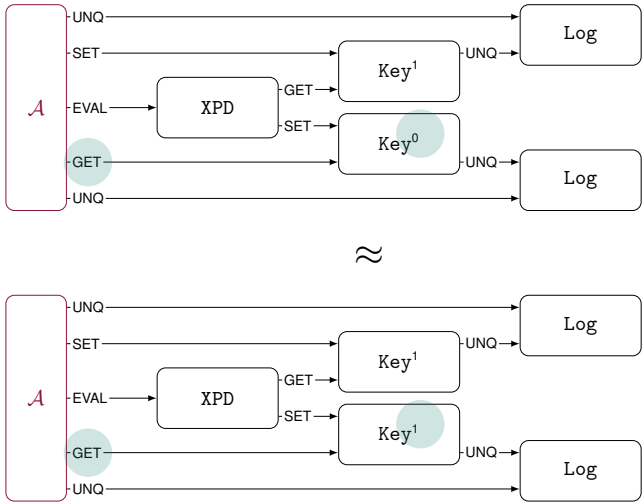
Core Key Schedule Proof



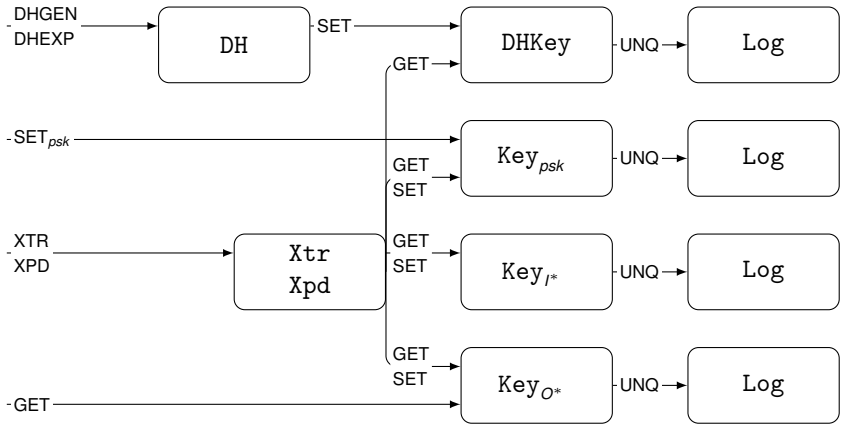
Core Key Schedule Proof



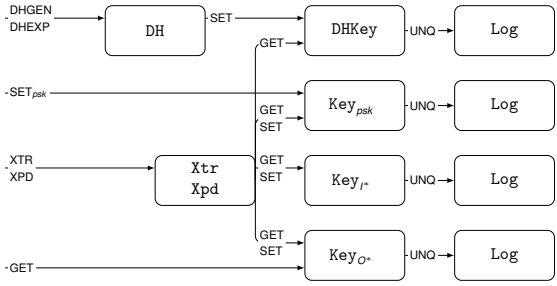
Core Key Schedule Proof



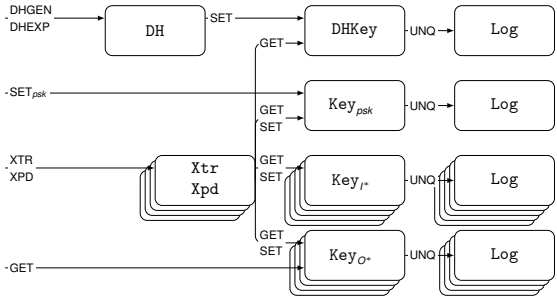
Core Key Schedule Proof



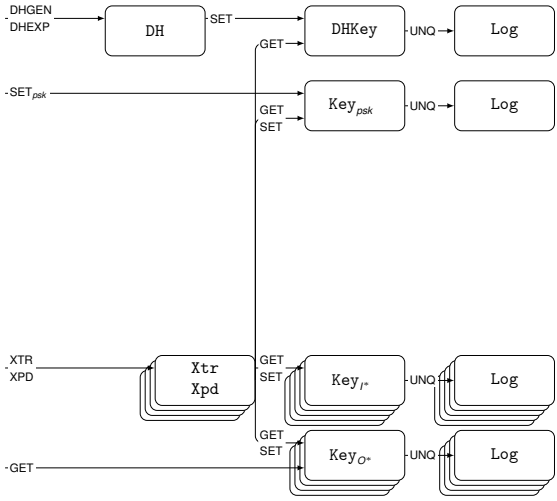
Core Key Schedule Proof



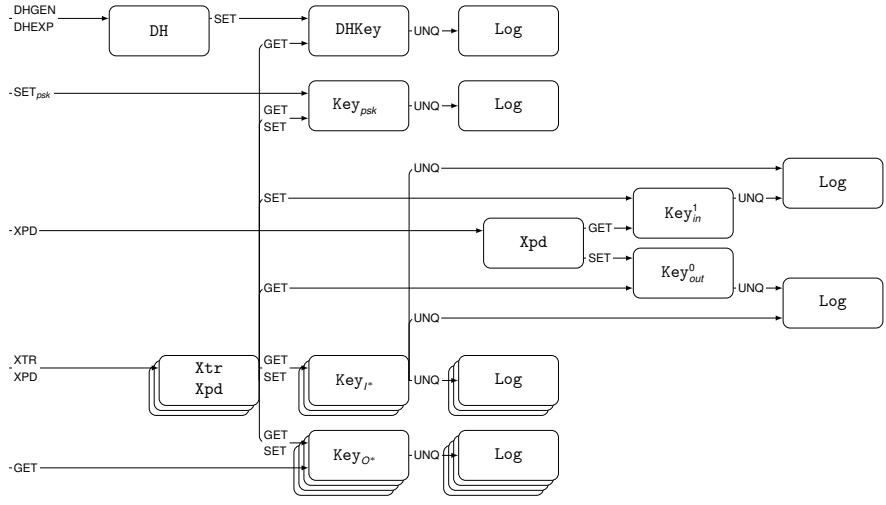
Core Key Schedule Proof



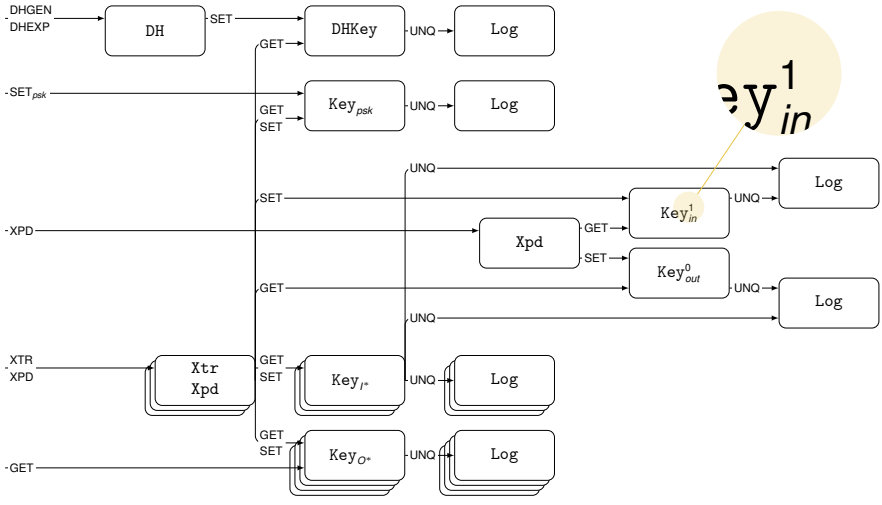
Core Key Schedule Proof



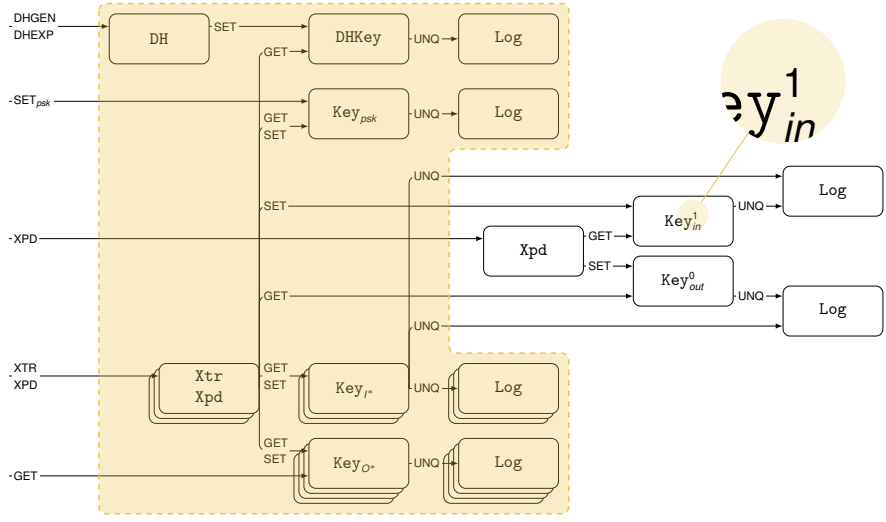
Core Key Schedule Proof



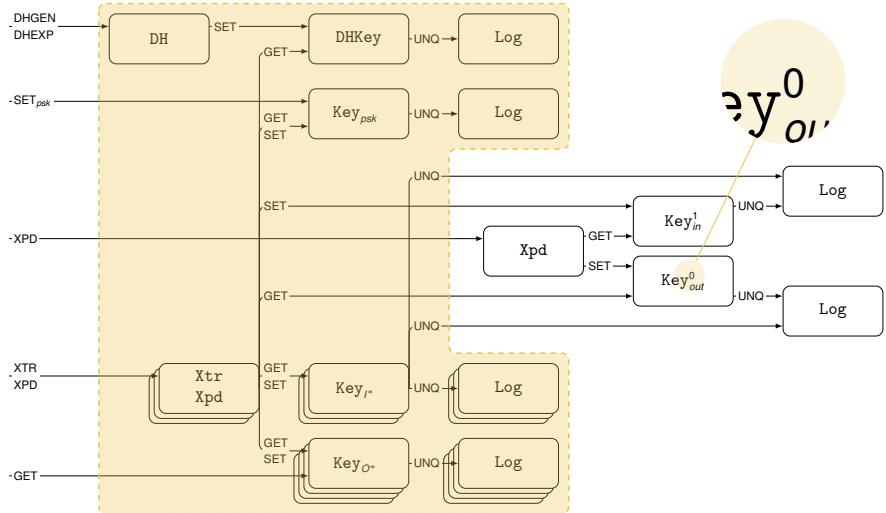
Core Key Schedule Proof



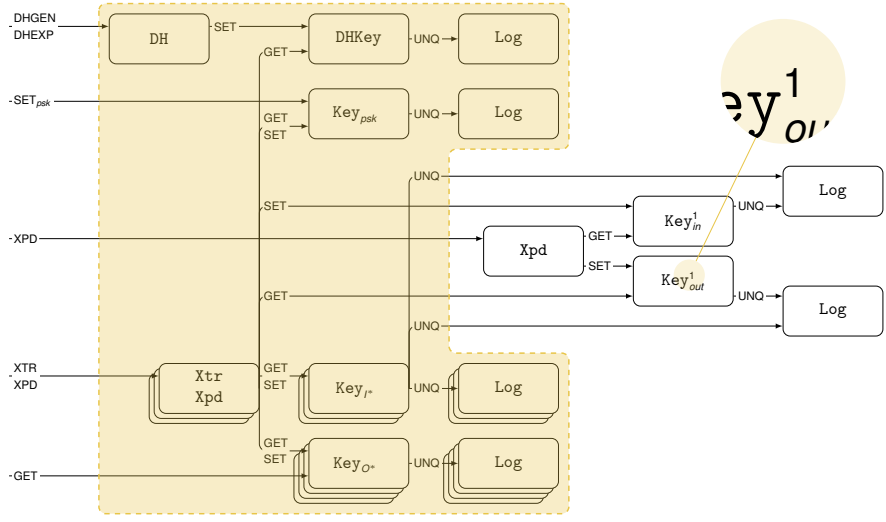
Core Key Schedule Proof



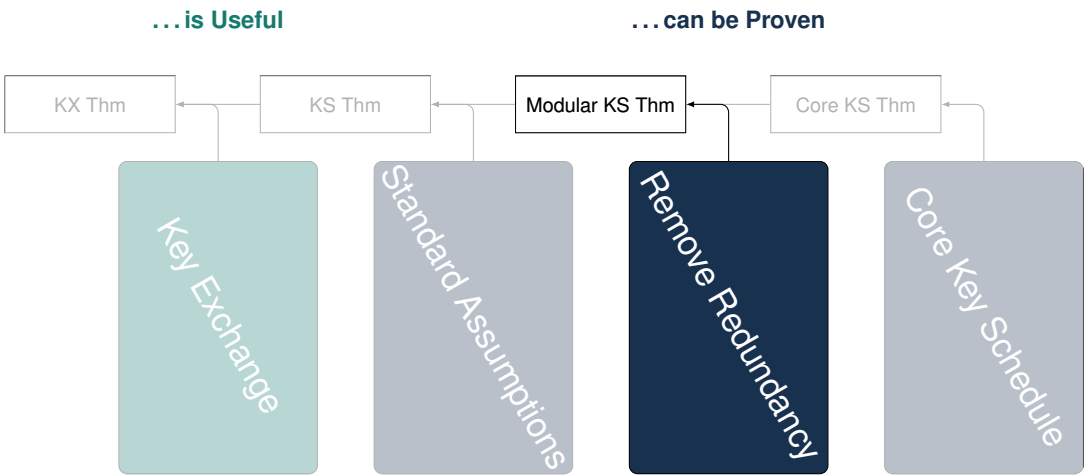
Core Key Schedule Proof



Core Key Schedule Proof



Proof Structure for TLS

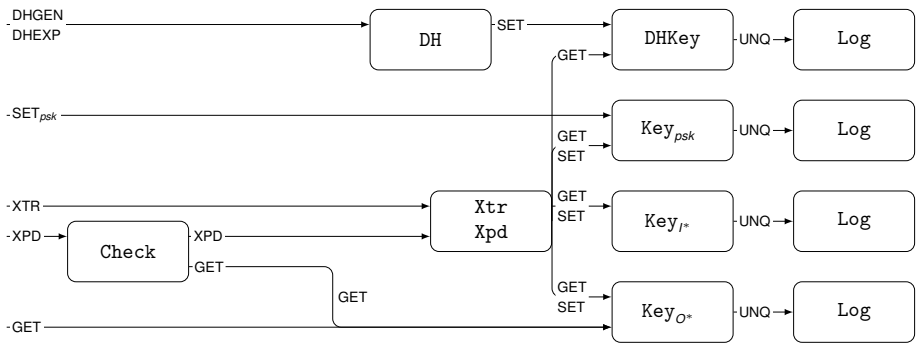


Mapping

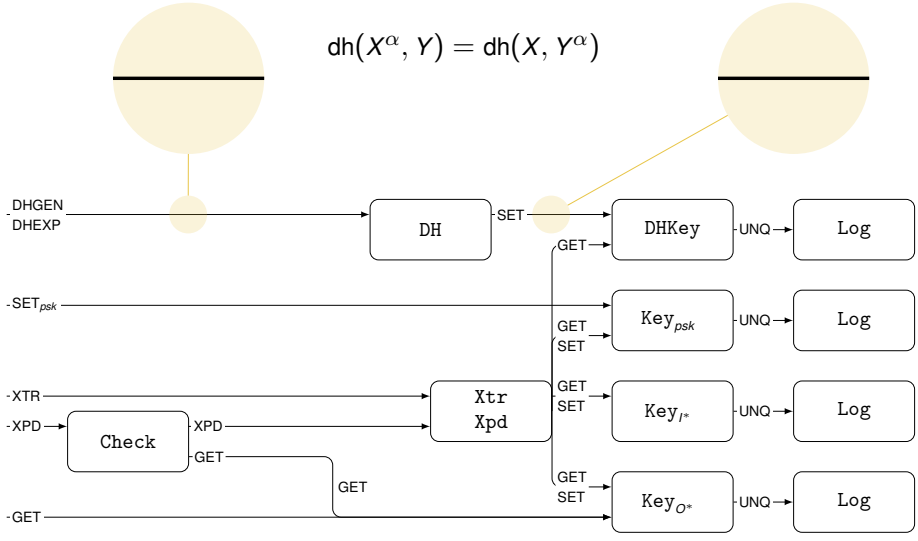
$$\mathrm{dh}(X^{\alpha}, Y) = \mathrm{dh}(X, Y^{\alpha})$$

Mapping

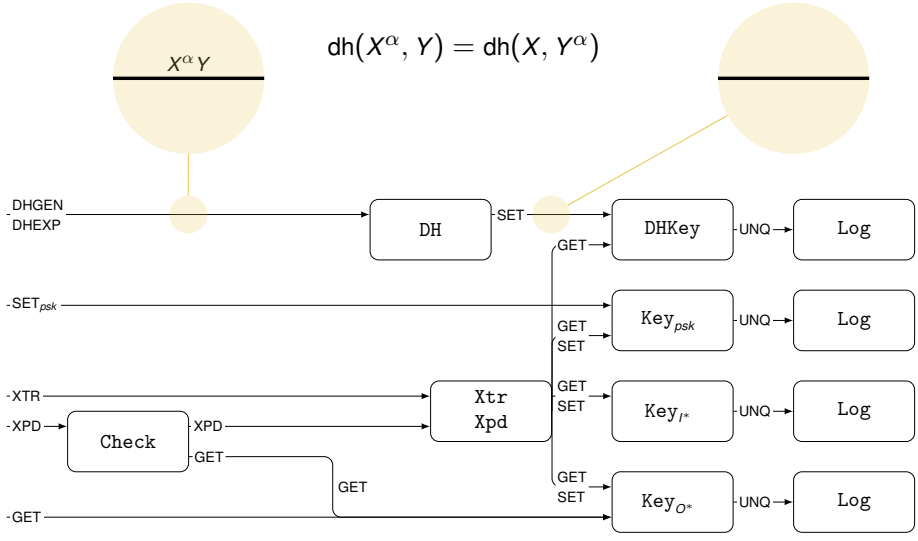
$$\mathsf{dh}(X^{\alpha}, Y) = \mathsf{dh}(X, Y^{\alpha})$$



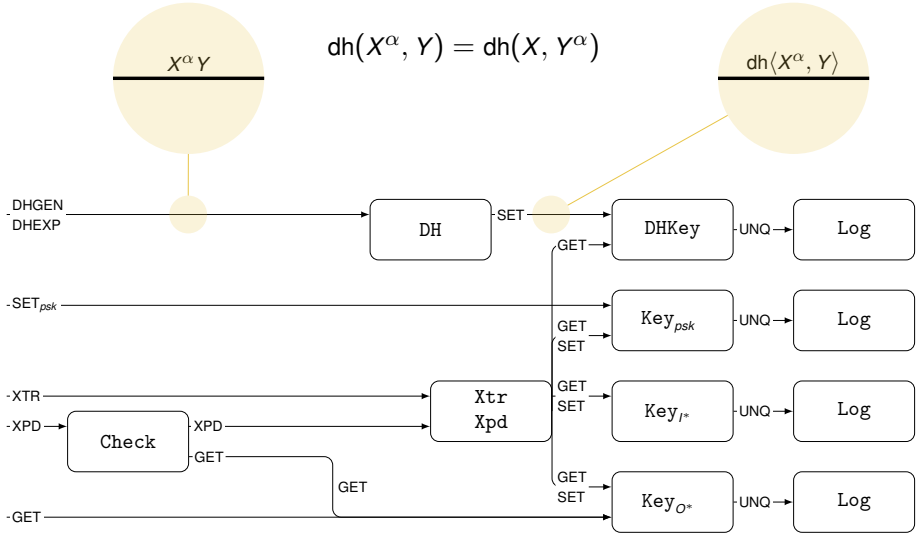
Mapping


$\mathrm{dh}(X^{\alpha}, Y) = \mathrm{dh}(X, Y^{\alpha})$
DHGEN
DHEXP
DH
SET
DHKey
UNQ
Log
GET
SET$_{psk}$
Key$_{psk}$
UNQ
Log
GET
SET
XTR
XPD
Check
XPD
GET
Xtr
Xpd
GET
SET
Key$_{I^*}$
UNQ
Log
GET
GET
SET
Key$_{O^*}$
UNQ
Log
GET

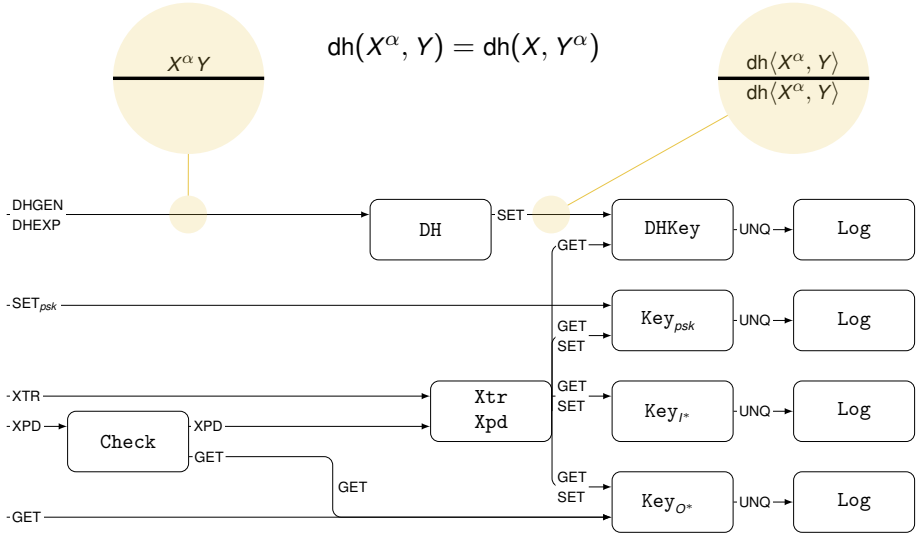
Mapping


$X^{\alpha}Y$
$\mathrm{dh}(X^{\alpha}, Y) = \mathrm{dh}(X, Y^{\alpha})$
DHGEN
DHEXP
DH
SET
DHKey
UNQ
Log
GET
SET$_{psk}$
Key$_{psk}$
UNQ
Log
GET
SET
XTR
XPD
Check
XPD
GET
Xtr
Xpd
GET
SET
Key$_{I^*}$
UNQ
Log
GET
GET
GET
SET
Key$_{O^*}$
UNQ
Log
GET

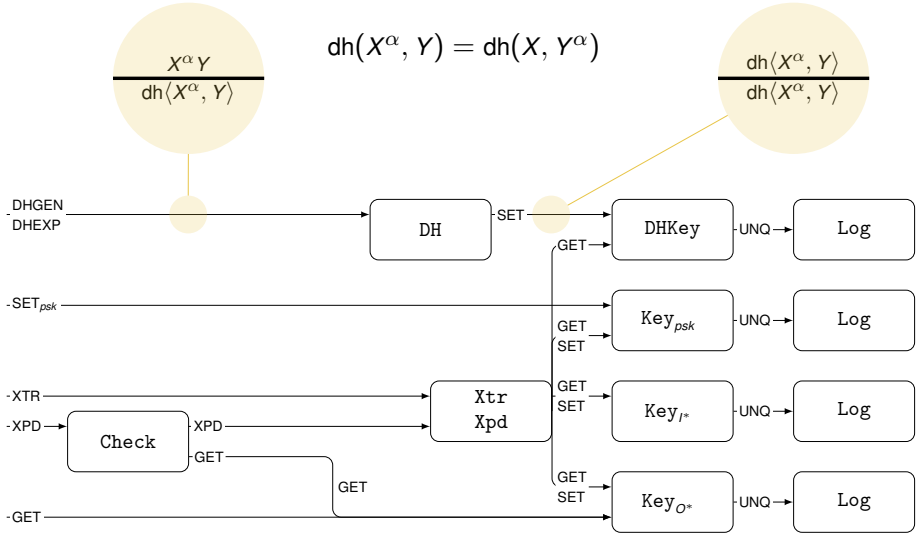
Mapping


$X^\alpha Y$
$\mathrm{dh}(X^\alpha, Y) = \mathrm{dh}(X, Y^\alpha)$
$\mathrm{dh}\langle X^\alpha, Y \rangle$
DHGEN
DHEXP
DH
SET
DHKey
UNQ
Log
GET
SET_{psk}
Key_{psk}
UNQ
Log
GET
SET
XTR
XPD
Check
XPD
GET
Xtr
Xpd
GET
SET
Key_{I^*}
UNQ
Log
GET
GET
GET
SET
Key_{O^*}
UNQ
Log
GET

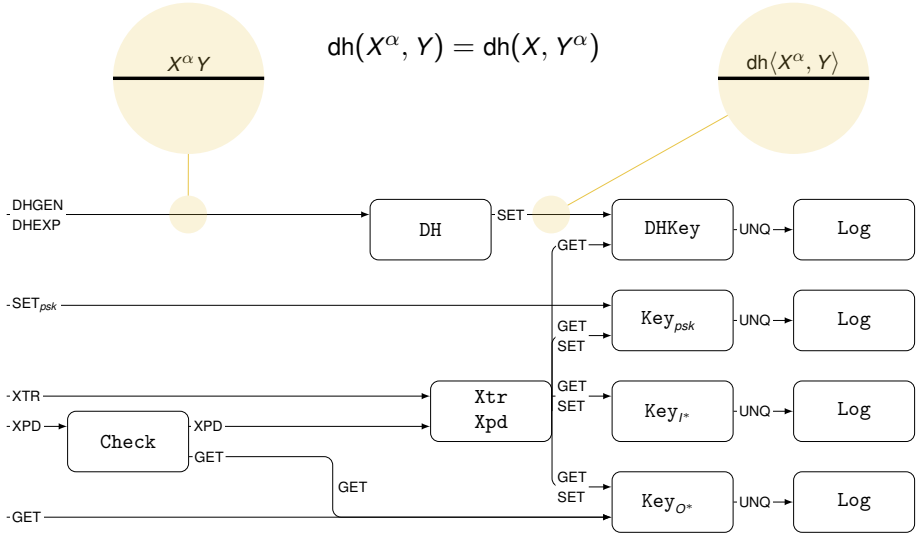
Mapping


$X^\alpha Y$
$\mathrm{dh}(X^\alpha, Y) = \mathrm{dh}(X, Y^\alpha)$
$\mathrm{dh}\langle X^\alpha, Y\rangle$
$\mathrm{dh}\langle X^\alpha, Y\rangle$
DHGEN
DHEXP
DH
SET
DHKey
UNQ
Log
GET
SET_{psk}
Key_{psk}
UNQ
Log
GET
SET
XTR
XPD
Check
XPD
GET
Xtr
Xpd
GET
SET
Key_{I^*}
UNQ
Log
GET
SET
GET
GET
SET
Key_{O^*}
UNQ
Log

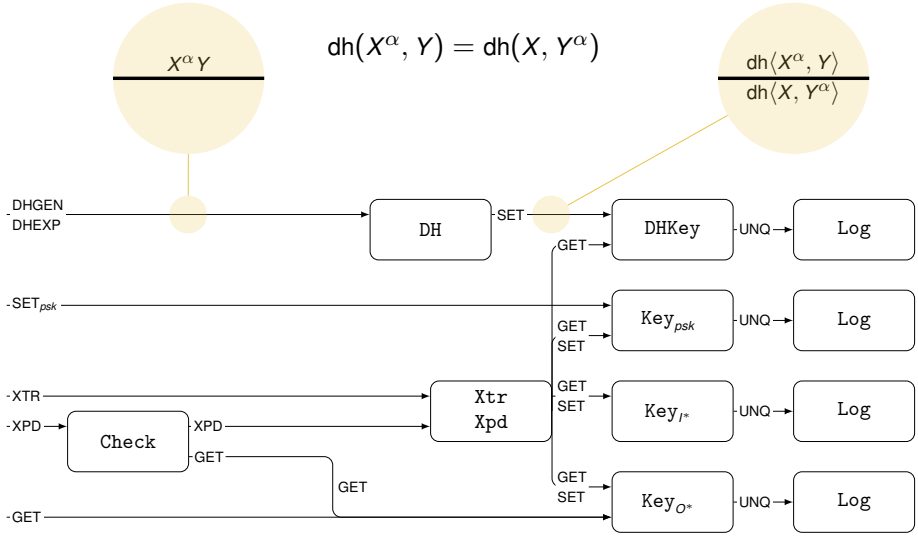
Mapping


$\frac{X^\alpha Y}{\mathrm{dh}\langle X^\alpha, Y\rangle}$
$\mathrm{dh}(X^\alpha, Y) = \mathrm{dh}(X, Y^\alpha)$
$\frac{\mathrm{dh}\langle X^\alpha, Y\rangle}{\mathrm{dh}\langle X^\alpha, Y\rangle}$
DHGEN
DHEXP
DH
SET
DHKey
UNQ
Log
GET
SET_{psk}
Key_{psk}
UNQ
Log
GET
SET
XTR
XPD
Check
XPD
GET
Xtr
Xpd
GET
SET
Key_{I^*}
UNQ
Log
GET
GET
GET
SET
Key_{O^*}
UNQ
Log
GET

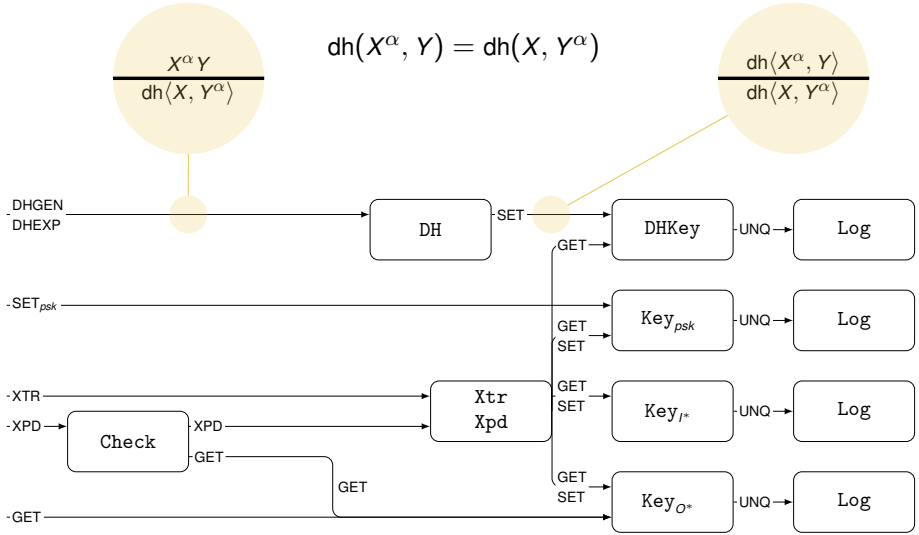
Mapping


$X^\alpha Y$
$\mathrm{dh}(X^\alpha, Y) = \mathrm{dh}(X, Y^\alpha)$
$\mathrm{dh}\langle X^\alpha, Y \rangle$
DHGEN
DHEXP
DH
SET
DHKey
UNQ
Log
GET
SET_{psk}
Key_{psk}
UNQ
Log
GET
SET
XTR
XPD
Check
XPD
GET
Xtr
Xpd
GET
SET
Key_{I^*}
UNQ
Log
GET
GET
GET
SET
Key_{O^*}
UNQ
Log
GET

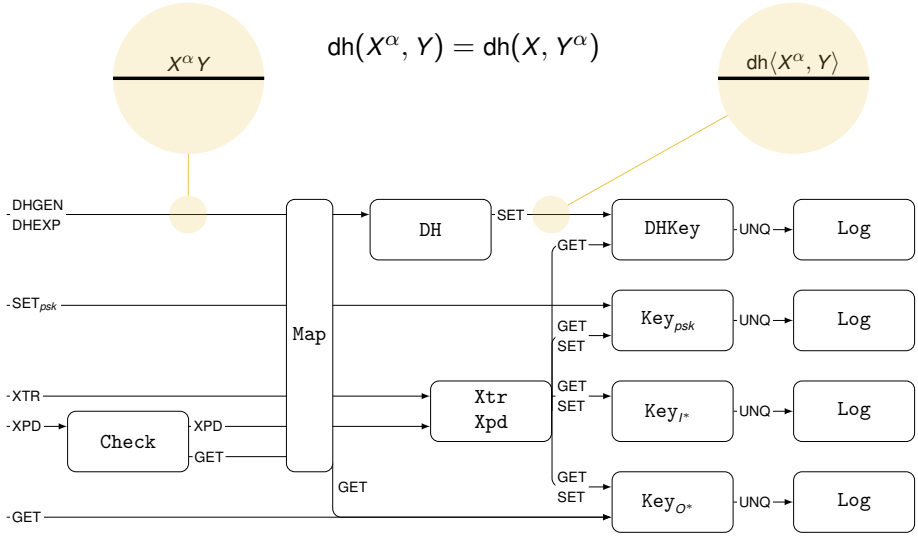
Mapping


$X^\alpha Y$
$\mathrm{dh}(X^\alpha, Y) = \mathrm{dh}(X, Y^\alpha)$
$\frac{\mathrm{dh}\langle X^\alpha, Y\rangle}{\mathrm{dh}\langle X, Y^\alpha\rangle}$
DHGEN
DHEXP
DH
SET
DHKey
UNQ
Log
GET
SET_{psk}
Key_{psk}
UNQ
Log
GET
SET
XTR
XPD
Check
XPD
GET
Xtr
Xpd
GET
SET
Key_{I^*}
UNQ
Log
GET
GET
GET
SET
Key_{O^*}
UNQ
Log

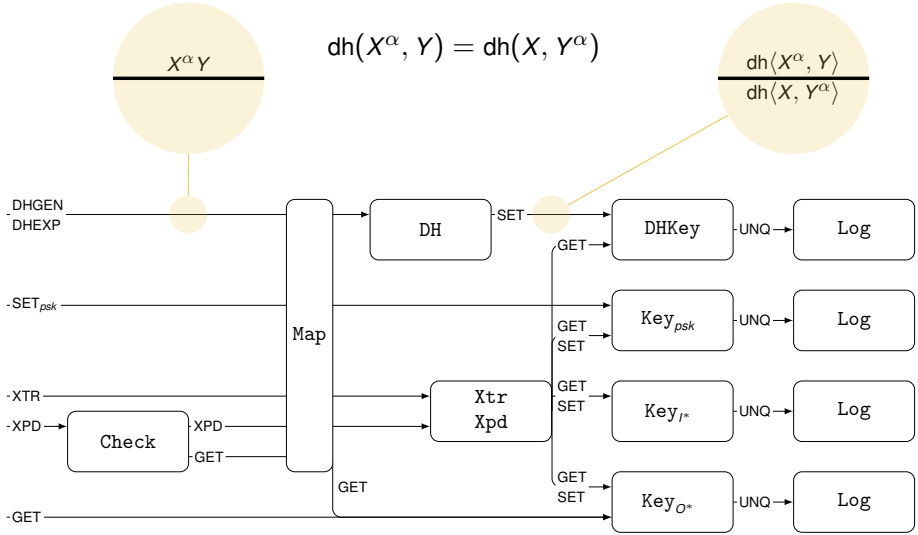
Mapping


$\frac{X^{\alpha}\,Y}{\mathrm{dh}\langle X,\, Y^{\alpha}\rangle}$
$\mathrm{dh}(X^{\alpha},\, Y) = \mathrm{dh}(X,\, Y^{\alpha})$
$\frac{\mathrm{dh}\langle X^{\alpha},\, Y\rangle}{\mathrm{dh}\langle X,\, Y^{\alpha}\rangle}$
DHGEN
DHEXP
DH
SET
DHKey
UNQ
Log
GET
SET$_{psk}$
Key$_{psk}$
UNQ
Log
GET
SET
XTR
XPD
Check
XPD
GET
Xtr
Xpd
GET
SET
Key$_{I^*}$
UNQ
Log
GET
GET
GET
SET
Key$_{O^*}$
UNQ
Log
GET

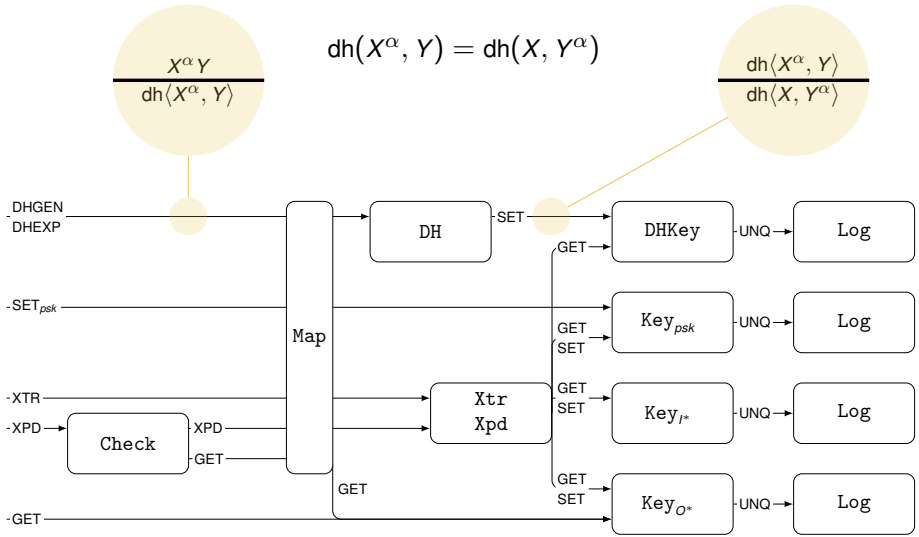
Mapping


$X^\alpha Y$
$\mathsf{dh}(X^\alpha, Y) = \mathsf{dh}(X, Y^\alpha)$
$\mathsf{dh}\langle X^\alpha, Y\rangle$
DHGEN
DHEXP
SET$_{psk}$
XTR
XPD
Check
XPD
GET
Map
GET
DH
SET
Xtr
Xpd
GET
SET
GET
SET
GET
SET
GET
DHKey
Key$_{psk}$
Key$_{I^*}$
Key$_{O^*}$
UNQ
UNQ
UNQ
UNQ
Log
Log
Log
Log

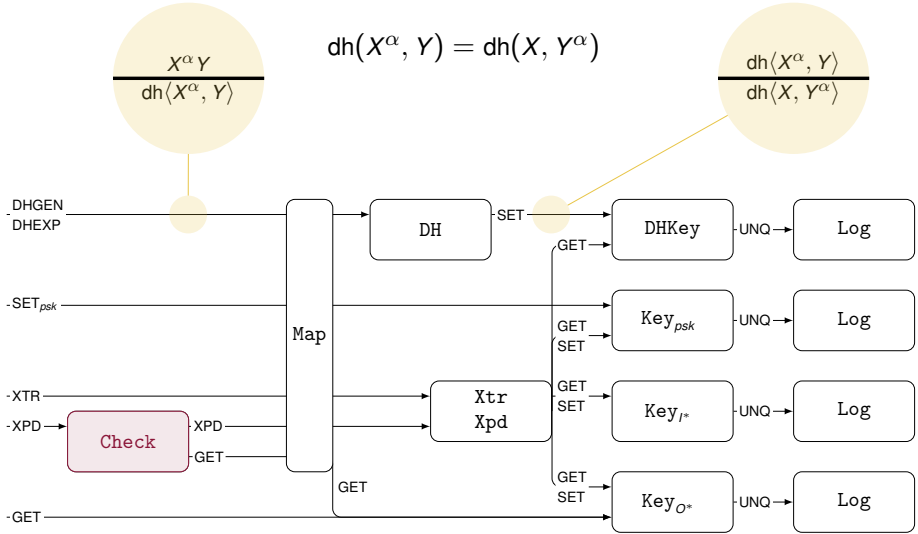
Mapping


$X^\alpha Y$
$\mathrm{dh}(X^\alpha, Y) = \mathrm{dh}(X, Y^\alpha)$
$\mathrm{dh}\langle X^\alpha, Y\rangle$
$\mathrm{dh}\langle X, Y^\alpha\rangle$
DHGEN
DHEXP
SET$_{psk}$
XTR
XPD
GET
Map
Check
XPD
GET
DH
Xtr
Xpd
SET
GET
GET
SET
GET
SET
GET
SET
DHKey
Key$_{psk}$
Key$_{I^*}$
Key$_{O^*}$
UNQ
UNQ
UNQ
UNQ
Log
Log
Log
Log

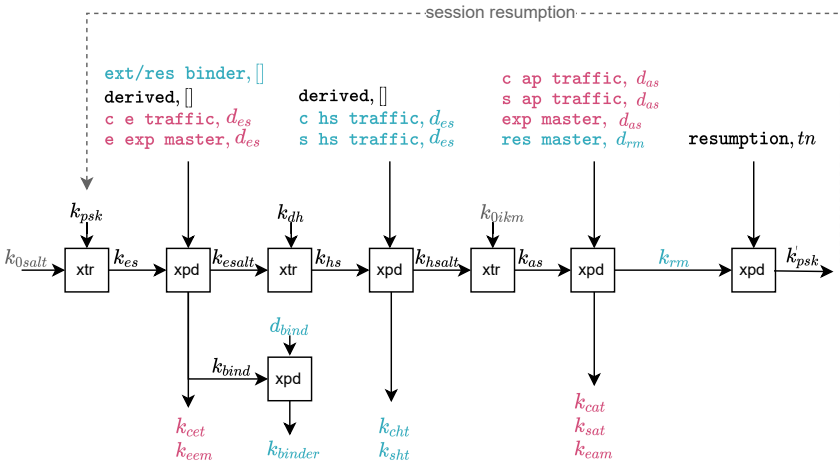
Mapping


$\frac{X^\alpha Y}{\mathrm{dh}\langle X^\alpha, Y\rangle}$
$\mathrm{dh}(X^\alpha, Y) = \mathrm{dh}(X, Y^\alpha)$
$\frac{\mathrm{dh}\langle X^\alpha, Y\rangle}{\mathrm{dh}\langle X, Y^\alpha\rangle}$
DHGEN
DHEXP
SET$_{psk}$
XTR
XPD
GET
Map
Check
XPD
GET
DH
Xtr
Xpd
SET
GET
GET
SET
GET
SET
GET
SET
DHKey
Key$_{psk}$
Key$_{I^*}$
Key$_{O^*}$
UNQ
UNQ
UNQ
UNQ
Log
Log
Log
Log

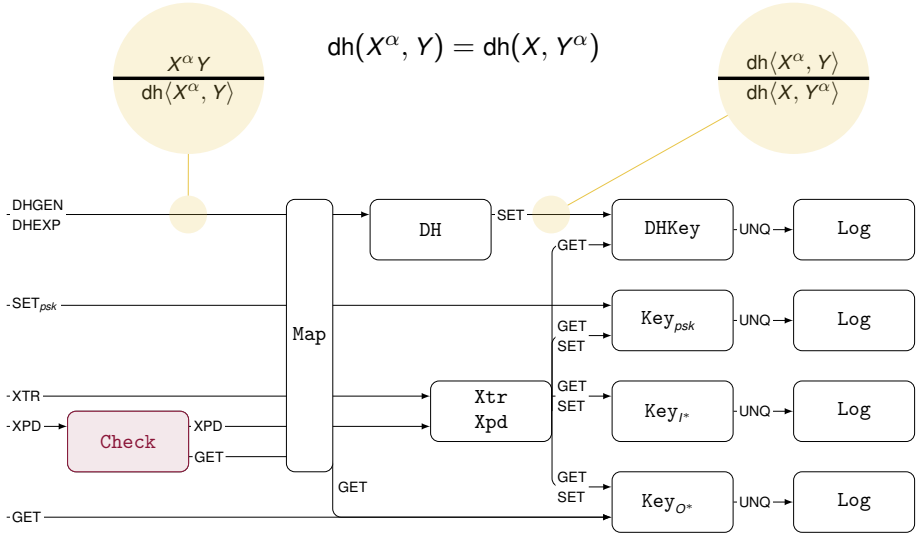
Mapping


$\frac{X^\alpha Y}{\mathrm{dh}\langle X^\alpha, Y\rangle}$
$\mathrm{dh}(X^\alpha, Y) = \mathrm{dh}(X, Y^\alpha)$
$\frac{\mathrm{dh}\langle X^\alpha, Y\rangle}{\mathrm{dh}\langle X, Y^\alpha\rangle}$
DHGEN
DHEXP
SET$_{psk}$
XTR
XPD
GET
Map
Check
XPD
GET
DH
Xtr
Xpd
GET
SET
DHKey
Key$_{psk}$
Key$_{I^*}$
Key$_{O^*}$
UNQ
Log

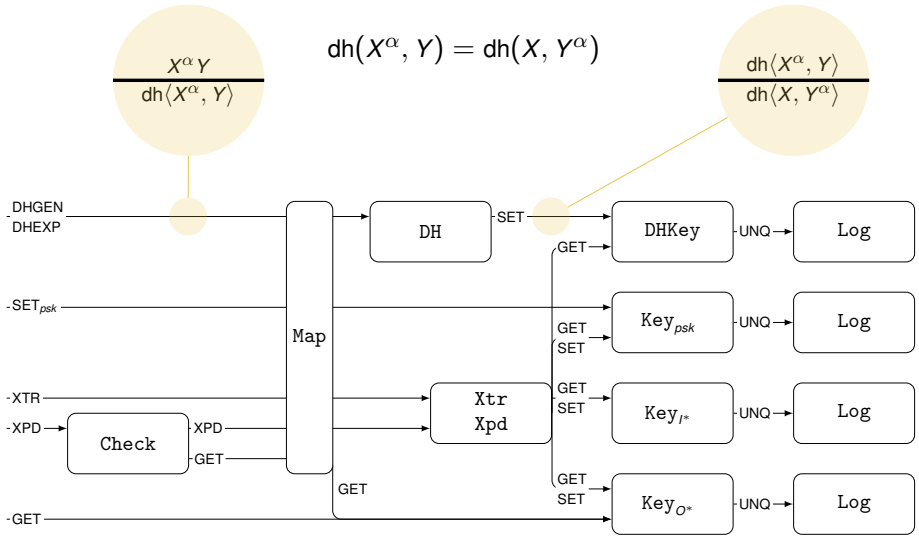
TLS 1.3 Key Schedule



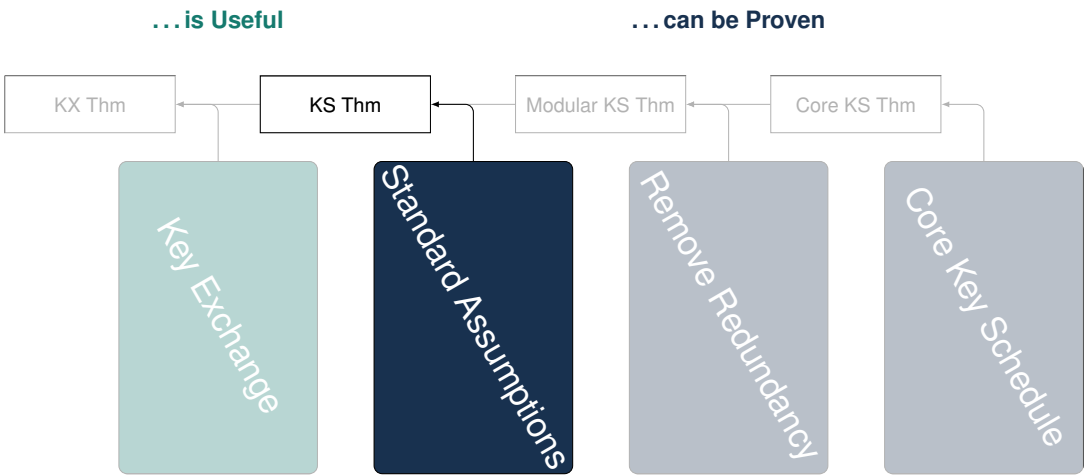
Mapping


$\frac{X^\alpha Y}{\mathrm{dh}\langle X^\alpha, Y\rangle}$
$\mathrm{dh}(X^\alpha, Y) = \mathrm{dh}(X, Y^\alpha)$
$\frac{\mathrm{dh}\langle X^\alpha, Y\rangle}{\mathrm{dh}\langle X, Y^\alpha\rangle}$
DHGEN
DHEXP
SET$_{psk}$
XTR
XPD
GET
Map
Check
XPD
GET
DH
Xtr
Xpd
GET
SET
DHKey
Key$_{psk}$
Key$_{I^*}$
Key$_{O^*}$
UNQ
Log

Mapping

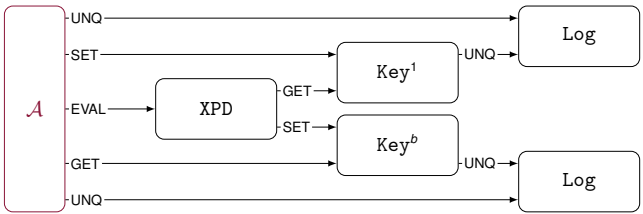

$\frac{X^\alpha Y}{\mathrm{dh}\langle X^\alpha, Y\rangle}$
$\mathrm{dh}(X^\alpha, Y) = \mathrm{dh}(X, Y^\alpha)$
$\frac{\mathrm{dh}\langle X^\alpha, Y\rangle}{\mathrm{dh}\langle X, Y^\alpha\rangle}$
DHGEN
DHEXP
SET$_{psk}$
XTR
XPD
GET
Map
Check
XPD
GET
DH
Xtr
Xpd
SET
GET
GET
SET
GET
SET
GET
SET
DHKey
Key$_{psk}$
Key$_{I^*}$
Key$_{O^*}$
UNQ
UNQ
UNQ
UNQ
Log
Log
Log
Log

Proof Structure for TLS



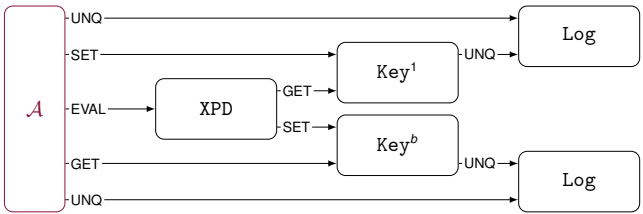
Standard Assumptions

$$\mathsf{Adv}(\mathcal{A}, \mathtt{Gxpd}^0_{n,\ell}, \mathtt{Gxpd}^1_{n,\ell}) \leq \sum_{alg \in \mathcal{H}} s_{n,\ell,alg} \cdot \mathsf{Adv}(\mathcal{A} \to \mathcal{R}^{alg}_{n,\ell}, \mathtt{Gpr}^{\text{xpd-}alg,b})$$



Standard Assumptions

$$\mathsf{Adv}(\mathcal{A}, \mathtt{Gxpd}^0_{n,\ell}, \mathtt{Gxpd}^1_{n,\ell}) \leq \sum_{alg \in \mathcal{H}} s_{n,\ell,alg} \cdot \mathsf{Adv}(\mathcal{A} \to \mathcal{R}^{alg}_{n,\ell}, \mathtt{Gpr}^{\text{xpd-}alg,b})$$



$\mathtt{Gpr}^{\text{xpd},0}$

$\mathsf{EVAL}(x)$

if $k = \perp$:

$k \leftarrow\!\!\$ \{0,1\}^\lambda$

return $\mathsf{xpd}(k, x)$

$\mathtt{Gpr}^{\text{xpd},1}$

$\mathsf{EVAL}(x)$

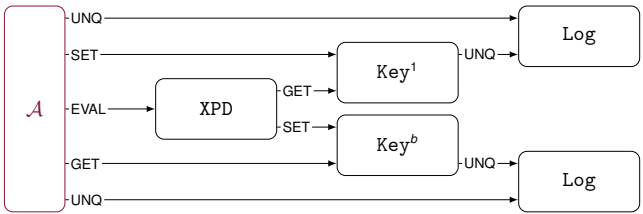
if $T[x] = \perp$:

$T[x] \leftarrow\!\!\$ \{0,1\}^\lambda$

return $T[x]$

Standard Assumptions

$$\mathsf{Adv}(\mathcal{A}, \mathtt{Gxpd}^0_{n,\ell}, \mathtt{Gxpd}^1_{n,\ell}) \leq \sum_{alg \in \mathcal{H}} s_{n,\ell,alg} \cdot \mathsf{Adv}(\mathcal{A} \to \mathcal{R}^{alg}_{n,\ell}, \mathtt{Gpr}^{\text{xpd-}alg,b})$$



$\mathtt{Gpr}^{\text{xpd},0}$ / $\mathsf{EVAL}(x)$

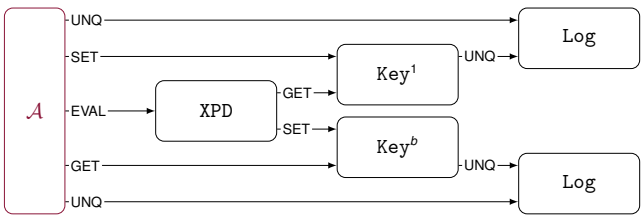
if $k = \bot$:
  $k \leftarrow\!\!\$\ \{0,1\}^\lambda$
return $\mathsf{xpd}(k, x)$

$\mathtt{Gpr}^{\text{xpd},1}$ / $\mathsf{EVAL}(x)$

if $T[x] = \bot$:
  $T[x] \leftarrow\!\!\$\ \{0,1\}^\lambda$
return $T[x]$

Standard Assumptions

$$\mathsf{Adv}(\mathcal{A}, \mathtt{Gxpd}^0_{n,\ell}, \mathtt{Gxpd}^1_{n,\ell}) \leq \sum_{alg \in \mathcal{H}} s_{n,\ell,alg} \cdot \mathsf{Adv}(\mathcal{A} \to \mathcal{R}^{alg}_{n,\ell}, \mathtt{Gpr}^{\text{xpd-}alg,b})$$



$\mathtt{Gpr}^{\text{xpd},0}$

$\mathsf{EVAL}(x)$

if $k = \bot$:

$k \leftarrow\!\!\$ \{0,1\}^\lambda$

return $\mathsf{xpd}(k, x)$

$\mathtt{Gpr}^{\text{xpd},1}$

$\mathsf{EVAL}(x)$

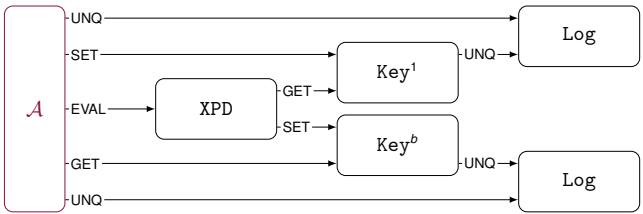
if $T[x] = \bot$:

$T[x] \leftarrow\!\!\$ \{0,1\}^\lambda$

return $T[x]$

Standard Assumptions

$$\mathsf{Adv}(\mathcal{A}, \mathtt{Gxpd}^0_{n,\ell}, \mathtt{Gxpd}^1_{n,\ell}) \leq \sum_{alg \in \mathcal{H}} s_{n,\ell,alg} \cdot \mathsf{Adv}(\mathcal{A} \to \mathcal{R}^{alg}_{n,\ell}, \mathtt{Gpr}^{\text{xpd-}alg,b})$$



$\mathtt{Gpr}^{\text{xpd},0}$

$\mathsf{EVAL}(x)$

if $k = \perp$:

$k \leftarrow\!\!\$ \{0,1\}^\lambda$

return $\mathsf{xpd}(k, x)$

$\mathtt{Gpr}^{\text{xpd},1}$

$\mathsf{EVAL}(x)$

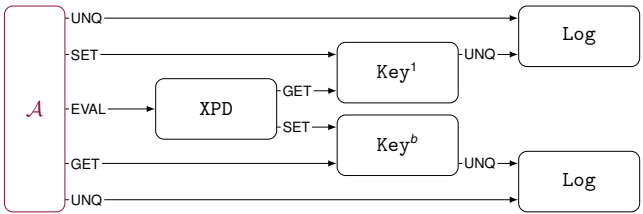
if $T[x] = \perp$:

$T[x] \leftarrow\!\!\$ \{0,1\}^\lambda$

return $T[x]$

Standard Assumptions

$$\mathsf{Adv}(\mathcal{A}, \mathtt{Gxpd}^0_{n,\ell}, \mathtt{Gxpd}^1_{n,\ell}) \leq \sum_{alg \in \mathcal{H}} s_{n,\ell,alg} \cdot \mathsf{Adv}(\mathcal{A} \to \mathcal{R}^{alg}_{n,\ell}, \mathtt{Gpr}^{\text{xpd-}alg,b})$$



$\mathtt{Gpr}^{\text{xpd},0}$

$\mathsf{EVAL}(x)$

if $k = \perp$:

$k \leftarrow\!\!\$\ \{0,1\}^\lambda$

return $\mathsf{xpd}(k, x)$

$\mathtt{Gpr}^{\text{xpd},1}$

$\mathsf{EVAL}(x)$

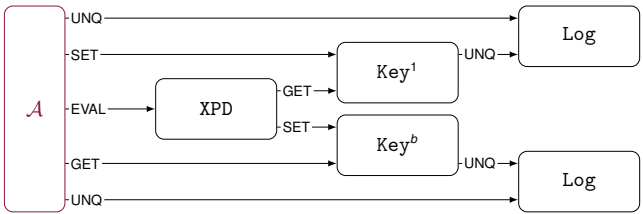
if $T[x] = \perp$:

$T[x] \leftarrow\!\!\$\ \{0,1\}^\lambda$

return $T[x]$

Standard Assumptions

$$\mathsf{Adv}(\mathcal{A}, \mathtt{Gxpd}^0_{n,\ell}, \mathtt{Gxpd}^1_{n,\ell}) \leq \sum_{alg \in \mathcal{H}} s_{n,\ell,alg} \, \mathsf{Adv}(\mathcal{A} \to \mathcal{R}^{alg}_{n,\ell}, \mathtt{Gpr}^{\text{xpd-}alg,b})$$



$\mathtt{Gpr}^{\text{xpd},0}$

$\mathsf{EVAL}(x)$

if $k = \bot$:

$k \leftarrow\!\!\$ \{0,1\}^\lambda$

return $\mathsf{xpd}(k, x)$

$\mathtt{Gpr}^{\text{xpd},1}$

$\mathsf{EVAL}(x)$

if $T[x] = \bot$:

$T[x] \leftarrow\!\!\$ \{0,1\}^\lambda$

return $T[x]$

Proof Structure for TLS