

# TREAD: A generic construction for provably TF resistant distance bounding

Gildas Avoine<sup>3</sup>   Xavier Bultel<sup>1</sup>   Sébastien Gambs<sup>2</sup>   **David Gerault**<sup>1</sup>   Pascal Lafourcade<sup>1</sup>   Cristina Onete<sup>3</sup>   Jean-Marc Robert<sup>4</sup>

<sup>1</sup>University Clermont Auvergne, <sup>2</sup>UQAM, Montréal, <sup>3</sup>INSA/IRISA Rennes, <sup>4</sup>ÉTS, Montréal

FutureDB



UNION EUROPÉENNE



# This talk : Outline

## TREAD

- ▶ Family of protocols
- ▶ Provably secure
- ▶ Modular : several possible instances
  - ▶ Lightweight
  - ▶ Private vs eavesdropper
  - ▶ Anonymous vs verifiers
- ▶ New mechanism for TF resistance

### **The DFKO framework**

- ▶ Another security model for DB
- ▶ Dürholz, Fischlin, Kasper, Onete (2011)
- ▶ Overview of the proofs for TREAD

# Why provable ?

- ▶ Too many broken protocols
- ▶ A 2015 survey on 40+ protocols found only 9 survivors
- ▶ New attack strategies appear regularly
  - ▶ TF against noise resistant protocols
  - ▶ MF via key recovery/bit flipping
  - ▶ PRF programming attacks
  - ▶ ...

# Why anonymous ?

- ▶ User tracking has become pervasive
- ▶ Location history reveal a lot :
  - ▶ Work address
  - ▶ Home address
  - ▶ Social links
- ▶ DB is a good candidate for tracking

## Why a new approach for TF resistance ?

- ▶ TF resistance : if  $P$  helps  $\mathcal{A}$  once, then  $\mathcal{A}$  can impersonate  $P$  at will
- ▶ Generally done by forcing  $P$  to leak his key for  $\mathcal{A}$  to succeed
- ▶  $r^0 = a, r^1 = a \oplus X$
- ▶  $P$  can give  $\mathcal{A}$  a noisy version of his secret : proving it is enough to pass is difficult

Previous solutions :

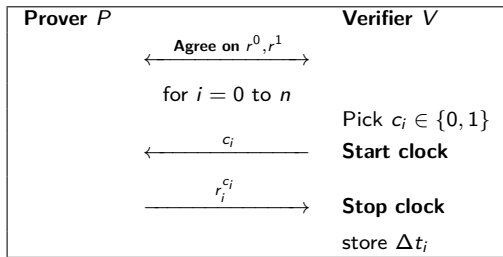
- ▶ FO, SPADE : Backdoor
- ▶ SKI, DBopt : Leakage scheme
- ▶ Proprox : Extractor

Additionally, it helped with anonymity

# Combining TF resistance and anonymity

- ▶ **Challenging** (with a revocation mechanism)
- ▶ Metro access system
  - ▶ Anonymity for location privacy
  - ▶ DF/DH/TF resistance to prevent access subletting
  - ▶ MF for obvious reasons

# TREAD : Design



Everything is in the setting of  $r^0$  and  $r^1$   
(Usually PRF)

## One problem : PRF programming

- ▶ PRF security :  $PRF_X(input) \equiv random$  if  $x$  random and hidden
- ▶  $\neq$  unpredictable if the key is known :  $Enc_x(input)$  is a PRF
- ▶ DF :  $PRF_X(NP, NV) =$ 
  - ▶  $Cste$  if  $NP = H(X)$
  - ▶  $f_x(NP, NV)$  otherwise

is a PRF

- ▶ Not safe if combined with the key either
- ▶ MF :  $PRF_X(NP, NV, a) =$ 
  - ▶  $X$  if  $a = f_x(NP, NV) \oplus X$
  - ▶  $f_x(NP, NV, a)$  otherwise

is a PRF

**Let's remove the PRF !**



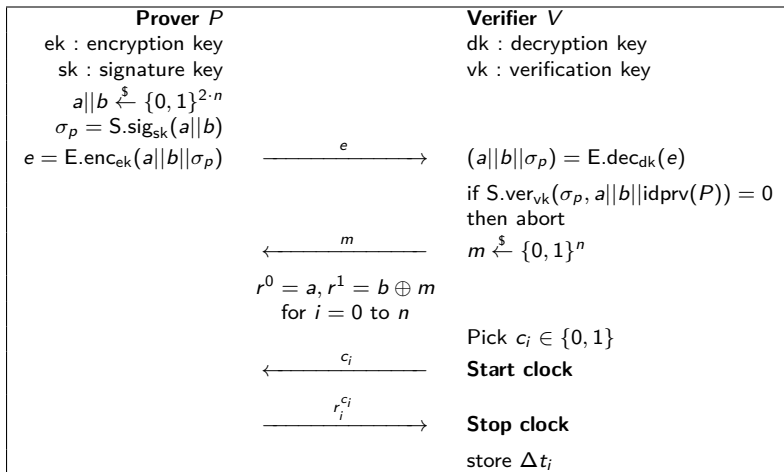
## How to pick the response vectors

- ▶  $\forall i, r_i^0 = r_i^1 : \text{DF, MF}$
- ▶  $\forall i, r_i^0 \neq r_i^1 : \text{DF, MF}$
- ▶  $\forall i, r_i^1 = r_i^0 \oplus X_i : \text{TF, MF}$

Solution : P pick  $a, b$ , V picks  $m$ ,  $r^0 = a$ ,  $r^1 = b \oplus m$

P authenticates  $a, b$  with a signature  $S$  and an encryption scheme  $E$

# TREAD...almost



# TREAD with identities

Prover $P$		Verifier $V$
ek : encryption key sk : signature key idpub( $P$ ) : public identity idprv( $P$ ) : private identity		dk : decryption key vk : verification key
$a  b \xleftarrow{\$} \{0,1\}^{2 \cdot n}$ $\sigma_p = S.sig_{sk}(a  b  idprv(P))$ $e = E.enc_{ek}(a  b  idprv(P)  \sigma_p)$	$\xrightarrow{e  idpub(P)}$	$(a  b  idprv(P)  \sigma_p) = E.dec_{dk}(e)$ if $S.ver_{vk}(\sigma_p, a  b  idprv(P)) = 0$ then abort $m \xleftarrow{\$} \{0,1\}^n$
	$\xleftarrow{m}$	
	$r^0 = a, r^1 = b \oplus m$ for $i = 0$ to $n$	
	$\xleftarrow{c_i}$	Pick $c_i \in \{0,1\}$ <b>Start clock</b>
	$\xrightarrow{r_i^{c_i}}$	<b>Stop clock</b> store $\Delta t_i$

# Security of TREAD

If  $E$  is a CCA-2 secure encryption scheme, and  $S$  is unforgeable, then TREAD is

- ▶ DF/DH resistant, and  $Pr^{DF} = (\frac{3}{4})^n$
- ▶ MF resistant, and  $Pr^{MF} = (\frac{3}{4})^n + \text{negl.}$  (signature+encryption)
- ▶ And SimTF Resistant

**Proofs in the DFKO (Dürholz, Fischlin, Kasper, Onete) model**

# The DFKO model

- ▶ Session based (P-V, P- $\mathcal{A}$ ,  $\mathcal{A}$ -V)
- ▶ A strictly increasing clock  $marker(sid, m)$  per session
- ▶  $\mathcal{A}$  quantified in terms of  $t, q_{obs}, q_P, q_V$
- ▶ Notion of tainted session
- ▶  $\mathcal{A}$  wins if he is accepted in an untainted session

## DFKO : DF/DH

A malicious, far away prover  $P^*$  tries to be accepted by  $V$ , possibly in the presence of honest provers near the verifier.

- ▶  $CommitTo(r_i) \equiv$  send in advance
- ▶  $Prompt(r_i) \equiv$  use  $P$ 's response
- ▶ In practice, two dummy sessions

A session is tainted if during the time critical part

- ▶  $P^*$  does not either commit or prompt for one round
- ▶  $P^*$  changes his response after receiving the challenge

$P^*$  wins if  $\exists pid$  such that

- ▶  $pid$  is not tainted
- ▶  $P^*$  is accepted in  $pid$

# TREAD : DH resistance

<b>Prover <math>P</math></b> ek : encryption key sk : signature key idpub( $P$ ) : public identity idprv( $P$ ) : private identity	<b>Verifier <math>V</math></b> dk : decryption key vk : verification key
$a  b \xleftarrow{\$} \{0, 1\}^{2 \cdot n}$ $\sigma_p = \text{S.sig}_{\text{sk}}(a  b  \text{idprv}(P))$ $e = \text{E.enc}_{\text{ek}}(a  b  \text{idprv}(P)  \sigma_p) \xrightarrow{e  \text{idpub}(P)} (a  b  \text{idprv}(P)  \sigma_p) = \text{E.dec}_{\text{dk}}(e)$ $r^0 = a, r^1 = b \oplus m$	

- ▶  $r_i^0 \neq r_i^1$  for half the rounds due to  $m$
- ▶ commit correct with probability  $\frac{3}{4}$  for each round
- ▶ For DH-like attacks :
- ▶  $Pr [r_i^{P^*} = r_i^P] = \frac{1}{2}$

**The best strategy is to guess  $r_i$  in advance**

## DFKO : MF

$\mathcal{A}$  is between a far away  $P$  and  $V$

A session is tainted if  $\mathcal{A}$  performs *pure relaying* during the time critical phase, ie :

$$P \xleftarrow{c'_i} \mathcal{A} \xleftarrow{c_i} V$$

$$P \xrightarrow{r_i} \mathcal{A} \xrightarrow{r'_i} V$$

- ▶  $c_i = c'_i$
- ▶  $r_i = r'_i$
- ▶  $\mathcal{A}$  receives  $c_i$  before sending  $c'_i$
- ▶  $\mathcal{A}$  receives  $r_i$  before sending  $r'_i$

$\mathcal{A}$  wins if  $\exists pid$  such that

- ▶  $pid$  is not tainted
- ▶  $P$  is accepted in  $pid$

**$\mathcal{A}$  can perform impure relaying**  
(equivalent to the learning phase of BMV)



# TREAD : MF resistance

<b>Prover <math>P</math></b> ek : encryption key sk : signature key idpub( $P$ ) : public identity idprv( $P$ ) : private identity	<b>Verifier <math>V</math></b> dk : decryption key vk : verification key
$a  b \xleftarrow{\$} \{0,1\}^{2 \cdot n}$ $\sigma_p = S.\text{sig}_{\text{sk}}(a  b  \text{idprv}(P))$ $e = E.\text{enc}_{\text{ek}}(a  b  \text{idprv}(P)  \sigma_p) \xrightarrow{e  \text{idpub}(P)} (a  b  \text{idprv}(P)  \sigma_p) = E.\text{dec}_{\text{dk}}(e)$ $r^0 = a, r^1 = b \oplus m$	

- ▶  $r_i^0 \neq r_i^1$  for half the rounds due to  $m$
- ▶  $a, b$  protected by E
- ▶  $\mathcal{A}$  cannot forge  $e$
- ▶  $a, b$  are never reused,  $\mathcal{A}$  only learns the responses for one set of challenges (or slightly more)

# DFKO : SimTF

$P^*$  helps  $\mathcal{A}$ , the protocol is resistant if after one success, a simulator with the view of  $\mathcal{A}$   $Pr_{sim}$  can pass with probability at least  $Pr_{\mathcal{A}}$ .

A session is tainted if  $\mathcal{A}$  and  $P^*$  communicate *at all* during the time critical phase.

Other notions :

- ▶ GameTF :  $Pr_{sim} \geq Pr^{MF}$
- ▶ strSimTF : like simTF, but  $\mathcal{A}$  and  $P^*$  can communicate during the time critical phase

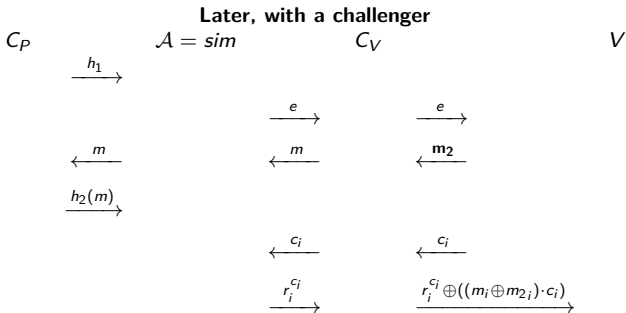
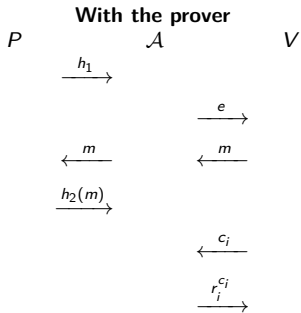
# TREAD : TF resistance

Counter intuitive

- ▶ Combining  $r^0$  and  $r^1$  gives no key
- ▶ No full secret vector recovery
- ▶ No backdoor/extractor/leakage scheme

Key idea :  $\mathcal{A}$  can replay  $e$   
Rewind  $\mathcal{A}$  and make him play again

# Replay



## No extraction

- ▶ No assumptions on  $h_1$  or  $h_2(m)$
- ▶ No assumption on how much bits  $\mathcal{A}$  knows
- ▶ If  $\mathcal{A}$  wins once, he can replay the same game and win again
- ▶ But : proof more difficult if no SimTF

## Some instances : symmetric

### **E : Symmetric encryption, S : Mac, idpub=P**

<b>Prover <math>P</math></b>	<b>Verifier <math>V</math></b>
ek : encryption key sk : signature key idpub( $P$ ) : public identity idprv( $P$ ) : private identity	dk : decryption key vk : verification key
$a  b \xleftarrow{\$} \{0,1\}^{2 \cdot n}$ $\sigma_p = S.sig_{sk}(a  b  idprv(P))$ $e = E.enc_{ek}(a  b  idprv(P)  \sigma_p) \xrightarrow{e  idpub(P)} (a  b  idprv(P)  \sigma_p) = E.dec_{dk}(e)$	

- ▶ No anonymity, no privacy
- ▶ Lightweight

## Some instances : public key

### **E : Public key encryption, idpub= $\emptyset$ , idpriv=P**

<b>Prover <math>P</math></b>	<b>Verifier <math>V</math></b>
ek : encryption key sk : signature key idpub( $P$ ) : public identity idpriv( $P$ ) : private identity	dk : decryption key vk : verification key
$a  b \xleftarrow{s} \{0,1\}^{2 \cdot n}$	
$\sigma_p = S.sig_{sk}(a  b  idpriv(P))$	
$e = E.enc_{ek}(a  b  idpriv(P)  \sigma_p)$	$(a  b  idpriv(P)  \sigma_p) = E.dec_{dk}(e)$

- ▶ Privacy vs eavesdroppers
- ▶ Requires public key

## Some instances : group signature

**E** : Public key encryption, **S** : Group signature,  $\text{idpub}=\emptyset$ ,  
 $\text{idpriv}=\text{GroupID}$

Prover $P$	Verifier $V$
$\text{ek}$ : encryption key $\text{sk}$ : signature key $\text{idpub}(P)$ : public identity $\text{idpriv}(P)$ : private identity	$\text{dk}$ : decryption key $\text{vk}$ : verification key
$a  b \xleftarrow{\$} \{0,1\}^{2 \cdot n}$ $\sigma_p = \text{S.sig}_{\text{sk}}(a  b  \text{idpriv}(P))$ $e = \text{E.enc}_{\text{ek}}(a  b  \text{idpriv}(P)  \sigma_p)$	$(a  b  \text{idpriv}(P)  \sigma_p) = \text{E.dec}_{\text{dk}}(e)$

- ▶ Privacy vs eavesdroppers
- ▶ Anonymous wrt verifiers
- ▶ Requires group signatures

**e can be precomputed**



## Other options

<b>Prover <math>P</math></b>	<b>Verifier <math>V</math></b>
ek : encryption key sk : signature key idpub( $P$ ) : public identity idpriv( $P$ ) : private identity	dk : decryption key vk : verification key
$a  b \xleftarrow{\$} \{0,1\}^{2 \cdot n}$	
$\sigma_p = S.\text{sig}_{\text{sk}}(a  b  \text{idpriv}(P))$	
$e = E.\text{enc}_{\text{ek}}(a  b  \text{idpriv}(P)  \sigma_p)$	$(a  b  \text{idpriv}(P)  \sigma_p) = E.\text{dec}_{\text{dk}}(e)$

- ▶ More lightweight : E : Symmetric encryption, S : MAC, idpub= $\emptyset$ , idpriv=P
- ▶ Identity based encryption
- ▶ idpub=group, idpriv=identity
- ▶ Any combination, as long as E is CCA2-secure and S is unforgeable.

**e can be precomputed**

# The anonymous version

- ▶ Anonymity directly comes from the group signature
- ▶ We extend the model to perform the proofs
- ▶ Join, corrupt, revoke...

# A note on recent attacks

## Distance-Bounding Protocols : Verification without Time and Location

and

Sjouke Mauw, Zach Smith, Jorge Toro-Pozo, Rolando Trujillo-Rasua

### Proving physical proximity using symbolic models

Alexandre Debant, Stéphanie Delaune, Cyrille Wiedling

- ▶ Multiple, possibly dishonest verifiers
- ▶ Not possible in our model, but still very relevant
- ▶  $V^*$  deciphers  $e$ , and gets  $a, b, \sigma_p = S.\text{sig}_{sk}(a||b)$
- ▶  $V^*$  can use  $a, b, \sigma_p = S.\text{sig}_{sk}(a||b)$  with another verifier and impersonate  $P$

## Possible solutions

- ▶ Authenticating the verifier (eg certificate)
- ▶ Setting  $\sigma_p = S.\text{sig}_{s_k}(a, b, V)$  and  $e = E.\text{enc}_{e_k}(a, b, V, \text{idprv}(P), \sigma_p)$
- ▶ **But** then TF resistance only for one verifier
- ▶ (needs to be verified)

**Thank you for your attention !**

Questions ?