How Secret-sharing can Defeat Terrorist Fraud

Gildas Avoine[1] Cédric Lauradoux[2] Benjamin Martin[1]

[1]Université catholique de Louvain
Belgium

[2] INRIA, Université de Lyon
France

17 June 2011

INFORMATION
SECURITY
GROUP

Plan

Wireless Authentication

ISO 9798-2

Definition (From the Handbook of Applied Cryptography)

An *authentication* is a process whereby one party is assured (through acquisition of corroborative evidence) of the identity of a second party involved in a protocol, and that the second has actually participated (*i.e.*, is active at, or immediately prior to, the time evidence is acquired).

secret x

secret x

$\xleftarrow{\quad N_V \quad}$ generates N_V

computes $R = E_x(N_V, V)$ $\xrightarrow{\quad R \quad}$

Relay Attack
Mafia fraud

$\xleftrightarrow{link}$

R $\quad \xleftarrow{N_V} \quad \xrightarrow{R}$ $\quad \xleftarrow{N_V} \quad \xrightarrow{R}$ $\quad \xleftarrow{N_V} \quad \xrightarrow{R}$ $\quad N_V$

Mafia Fraud
- First mention : J.H.Conway 1974
- Reintroduced by Desmedt *et al* 87

Terrorist Fraud
- First mention : Bengio *et al* 91

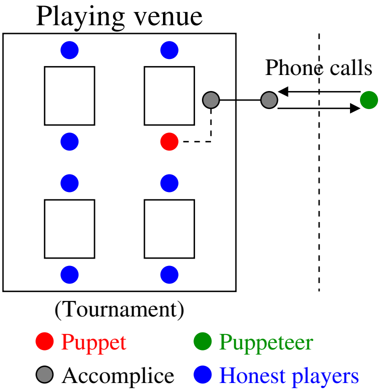
Distance Fraud
- First mention : Brands *et al* 93

Which counter measure ?

Measuring the time spent for an exchange.

Terrorist Fraud

An example : 2010 Chess Olympiad



Terrorist Fraud

The notions

Problematic on terrorist fraud

- Bart helps the adversaries.
- Bart wants its key to remain secret.

What we want to achieve

- If Bart shares too many informations, the protocol must reveal its key.
- If Bart is honest, the protocol must not reveal its key

The solution

The secret-sharing.
First use by Bussard and Bagga in 2005.

Secret-sharing

Definitions

Secret-sharing

- A dealer shares a secret key s between n parties.
- Each party $i \in [1, n]$ receives a share.
- **Predefined groups** of parties can cooperate to recover s.
- **Any other group of parties have no idea on what is s.**

Threshold cryptography

Let Λ be an (n, k) threshold scheme :

- A dealer shares a secret key s between n parties.
- Each party $i \in [1, n]$ receives a share.
- Any group of k participants can cooperate to recover s.
- Groups of $a < k$ participants cannot get anything on s.

Hancke and Khun 2005

The protocol

secret x

secret x

slow phase

fast phase

Hancke and Khun 2005

The protocol

secret x

secret x

slow phase

fast phase

secret x

secret x

slow phase

generates N_P

generates N_V

$\xrightarrow{N_P}$

$\xleftarrow{N_V}$

$H^{2n} = PRF(x, N_V, N_P)$

$R^0 : \boxed{}\boxed{}\boxed{\dots}\boxed{}$

$R^1 : \boxed{}\boxed{}\boxed{\dots}\boxed{}$

$H^{2n} = PRF(x, N_V, N_P)$

$R^0 : \boxed{}\boxed{}\boxed{\dots}\boxed{}$

$R^1 : \boxed{}\boxed{}\boxed{\dots}\boxed{}$

fast phase

Hancke and Khun 2005

The protocol

secret x | secret x

slow phase

generates N_P | generates N_V

N_P →

← N_V

$H^{2n} = PRF(x, N_V, N_P)$
$R^0 : \boxed{}\boxed{}\boxed{\;\ldots\;}\boxed{}$
$R^1 : \boxed{}\boxed{}\boxed{\;\ldots\;}\boxed{}$

$H^{2n} = PRF(x, N_V, N_P)$
$R^0 : \boxed{}\boxed{}\boxed{\;\ldots\;}\boxed{}$
$R^1 : \boxed{}\boxed{}\boxed{\;\ldots\;}\boxed{}$

fast phase

Hancke and Khun 2005

The protocol

Prover		Verifier
secret x		secret x
	slow phase	
generates N_P		generates N_V
	$\xrightarrow{\quad N_P \quad}$	
	$\xleftarrow{\quad N_V \quad}$	
$H^{2n} = PRF(x, N_V, N_P)$		$H^{2n} = PRF(x, N_V, N_P)$
R^0 : $\boxed{}\boxed{}\boxed{\ldots}\boxed{}$		R^0 : $\boxed{}\boxed{}\boxed{\ldots}\boxed{}$
R^1 : $\boxed{}\boxed{}\boxed{\ldots}\boxed{}$		R^1 : $\boxed{}\boxed{}\boxed{\ldots}\boxed{}$
	fast phase	

Hancke and Khun 2005

The protocol

secret x | secret x

slow phase

generates N_P | generates N_V

$$\xrightarrow{N_P}$$

$$\xleftarrow{N_V}$$

$$H^{2n} = PRF(x, N_V, N_P) \qquad H^{2n} = PRF(x, N_V, N_P)$$

R^0 : | | | … | | R^0 : | | | … | |

R^1 : | | | … | | R^1 : | | | … | |

fast phase

secret x | secret x

slow phase

generates N_P | generates N_V

$$\xrightarrow{\quad N_P \quad}$$

$$\xleftarrow{\quad N_V \quad}$$

$H^{2n} = PRF(x, N_V, N_P)$ | $H^{2n} = PRF(x, N_V, N_P)$

R^0 : $\boxed{}\boxed{}\boxed{\ \dots\ }\boxed{}$ | R^0 : $\boxed{}\boxed{}\boxed{\ \dots\ }\boxed{}$

R^1 : $\boxed{}\boxed{}\boxed{\ \dots\ }\boxed{}$ | R^1 : $\boxed{}\boxed{}\boxed{\ \dots\ }\boxed{}$

fast phase

Hancke and Khun 2005

The protocol

Prover		Verifier
secret x		secret x
	slow phase	
generates N_P		generates N_V
	$\xrightarrow{N_P}$	
	$\xleftarrow{N_V}$	
$H^{2n} = PRF(x, N_V, N_P)$		$H^{2n} = PRF(x, N_V, N_P)$
R^0 : $\boxed{}\boxed{}\boxed{\dots}\boxed{}$		R^0 : $\boxed{}\boxed{}\boxed{\dots}\boxed{}$
R^1 : $\boxed{}\boxed{}\boxed{\dots}\boxed{}$		R^1 : $\boxed{}\boxed{}\boxed{\dots}\boxed{}$
	fast phase	
	for $i = 1, \dots, n$ **:**	
		picks a bit c_i
	$\xleftarrow{c_i}$	starts timer
$r_i = R_i^{c_i}$	$\xrightarrow{r_i}$	stops timer

Hancke and Khun 2005

The protocol

secret x

secret x

slow phase

generates N_P

generates N_V

$\xrightarrow{N_P}$

$\xleftarrow{N_V}$

$H^{2n} = PRF(x, N_V, N_P)$

R^0 : | | | … | |

R^1 : | | | … | |

$H^{2n} = PRF(x, N_V, N_P)$

R^0 : | | | … | |

R^1 : | | | … | |

fast phase

for $i = 1, \ldots, n$ **:**

picks a bit c_i

starts timer

$\xleftarrow{c_i}$

$r_i = R_i^{c_i}$

$\xrightarrow{r_i}$

stops timer

Hancke and Khun 2005

The protocol

Prover		Verifier
secret x		secret x
	slow phase	
generates N_P		generates N_V
	$\xrightarrow{N_P}$	
	$\xleftarrow{N_V}$	
$H^{2n} = PRF(x, N_V, N_P)$		$H^{2n} = PRF(x, N_V, N_P)$
$R^0 : \boxed{}\boxed{}\boxed{\dots}\boxed{}$		$R^0 : \boxed{}\boxed{}\boxed{\dots}\boxed{}$
$R^1 : \boxed{}\boxed{}\boxed{\dots}\boxed{}$		$R^1 : \boxed{}\boxed{}\boxed{\dots}\boxed{}$
	fast phase **for** $i = 1, \dots, n$ **:**	
		picks a bit c_i
	$\xleftarrow{c_i}$	starts timer
$r_i = R_i^{c_i}$	$\xrightarrow{r_i}$	stops timer

Hancke and Khun 2005

The protocol

secret x

secret x

slow phase

generates N_P

generates N_V

$$\xrightarrow{N_P}$$

$$\xleftarrow{N_V}$$

$H^{2n} = PRF(x, N_V, N_P)$

$R^0 : \boxed{}\boxed{}\boxed{\dots}\boxed{}$

$R^1 : \boxed{}\boxed{}\boxed{\dots}\boxed{}$

$H^{2n} = PRF(x, N_V, N_P)$

$R^0 : \boxed{}\boxed{}\boxed{\dots}\boxed{}$

$R^1 : \boxed{}\boxed{}\boxed{\dots}\boxed{}$

fast phase

for $i = 1, \dots, n$ **:**

picks a bit c_i

starts timer

$$\xleftarrow{c_i}$$

$r_i = R_i^{c_i}$

$$\xrightarrow{r_i}$$

stops timer

Hancke and Khun 2005

The protocol

secret x | secret x

slow phase

generates N_P | generates N_V

$$\xrightarrow{\quad N_P \quad}$$

$$\xleftarrow{\quad N_V \quad}$$

$H^{2n} = PRF(x, N_V, N_P)$ | $H^{2n} = PRF(x, N_V, N_P)$

$R^0 : \boxed{}\boxed{}\boxed{\ldots}\boxed{}$ | $R^0 : \boxed{}\boxed{}\boxed{\ldots}\boxed{}$

$R^1 : \boxed{}\boxed{}\boxed{\ldots}\boxed{}$ | $R^1 : \boxed{}\boxed{}\boxed{\ldots}\boxed{}$

fast phase

for $i = 1, \ldots, n$ **:**

picks a bit c_i

starts timer

$$\xleftarrow{\quad c_i \quad}$$

$r_i = R_i^{c_i}$

$$\xrightarrow{\quad r_i \quad}$$

stops timer

Hancke and Khun 2005

The protocol

secret x | secret x

slow phase

generates N_P | generates N_V

$$\xrightarrow{\quad N_P \quad}$$

$$\xleftarrow{\quad N_V \quad}$$

$$H^{2n} = PRF(x, N_V, N_P) \qquad H^{2n} = PRF(x, N_V, N_P)$$

$R^0 : \boxed{\ }\boxed{\ }\boxed{\dots}\boxed{\ }$ | $R^0 : \boxed{\ }\boxed{\ }\boxed{\dots}\boxed{\ }$

$R^1 : \boxed{\ }\boxed{\ }\boxed{\dots}\boxed{\ }$ | $R^1 : \boxed{\ }\boxed{\ }\boxed{\dots}\boxed{\ }$

fast phase

for $i = 1, \dots, n$ **:**

picks a bit c_i

starts timer

$$\xleftarrow{\quad c_i \quad}$$

$r_i = R_i^{c_i}$

$$\xrightarrow{\quad r_i \quad}$$

stops timer

Hancke and Khun 2005

The protocol

secret x

secret x

slow phase

generates N_P | generates N_V

$\xrightarrow{N_P}$

$\xleftarrow{N_V}$

$H^{2n} = PRF(x, N_V, N_P)$ | $H^{2n} = PRF(x, N_V, N_P)$

$R^0 : \boxed{}\boxed{}\boxed{\;\dots\;}\boxed{}$ | $R^0 : \boxed{}\boxed{}\boxed{\;\dots\;}\boxed{}$

$R^1 : \boxed{}\boxed{}\boxed{\;\dots\;}\boxed{}$ | $R^1 : \boxed{}\boxed{}\boxed{\;\dots\;}\boxed{}$

fast phase

for $i = 1, \dots, n$ **:**

picks a bit c_i

starts timer

$\xleftarrow{c_i}$

$r_i = R_i^{c_i}$

$\xrightarrow{r_i}$

stops timer

Hancke and Khun 2005

The protocol

Prover		Verifier
secret x		secret x
	slow phase	
generates N_P		generates N_V
	$\xrightarrow{N_P}$	
	$\xleftarrow{N_V}$	
$H^{2n} = PRF(x, N_V, N_P)$		$H^{2n} = PRF(x, N_V, N_P)$
$R^0 : \boxed{\ }\boxed{\ }\boxed{\ldots}\boxed{\ }$		$R^0 : \boxed{\ }\boxed{\ }\boxed{\ldots}\boxed{\ }$
$R^1 : \boxed{\ }\boxed{\ }\boxed{\ldots}\boxed{\ }$		$R^1 : \boxed{\ }\boxed{\ }\boxed{\ldots}\boxed{\ }$
	fast phase	
	for $i = 1, \ldots, n$ **:**	
		picks a bit c_i
	$\xleftarrow{c_i}$	starts timer
$r_i = R_i^{c_i}$	$\xrightarrow{r_i}$	stops timer

Hancke and Khun 2005

Protocol analysis

Mafia fraud strategies

- Post-ask strategy : $\frac{1}{2}$
- Pre-ask strategy : $\frac{3}{4}$

Mafia fraud success probability

The adversary chooses the pre-ask strategy, and succeeds with probability :

$$\Pr_{MF} = \left(\frac{3}{4}\right)^n$$

Terrorist fraud success probability

The prover provides R^0 and R^1 to the adversary.

$$\Pr_{\mathrm{TF}} = 1.$$

Our Contribution

Refinement of the adversary model

Based on the knowledge of the protocol output.

Introduction of the three adversary types.

Closer look on key recovery attacks.

Review of existing solutions.

New approach on terrorist fraud

(Explicit) introduction of secret sharing.

Use/misuse of the secret-sharing in distance bounding.

New protocols : TDB,TTDB.

Threshold Distance Bounding (TDB)

A simple instance

$x \in \mathbb{F}_2^4, PRF, \Lambda$

$x \in \mathbb{F}_2^4, PRF, \Lambda$

INITIALIZATION PHASE

Picks N_P

$\xrightarrow{N_P}$

$\xleftarrow{N_V}$

Picks N_V

$$\mathcal{R} = \begin{pmatrix} r_{1,1} & r_{1,2} & r_{1,3} & r_{1,4} \\ r_{2,1} & r_{2,2} & r_{2,3} & r_{2,4} \\ r_{3,1} & r_{3,2} & r_{3,3} & r_{3,4} \end{pmatrix}$$

$$\mathcal{R} = \begin{pmatrix} r_{1,1} & r_{1,2} & r_{1,3} & r_{1,4} \\ r_{2,1} & r_{2,2} & r_{2,3} & r_{2,4} \\ r_{3,1} & r_{3,2} & r_{3,3} & r_{3,4} \end{pmatrix}$$

INTERACTIVE PHASE

For $i = 1 \cdots 4$

$\xleftarrow{c_i}$

Picks $c_i \in [1, 3]$

Sends $r_{c_i,i}$

$\xrightarrow{r'_{c_i,i}}$

Measures δt_i

RESULT

If $\forall i \ r_{c_i,i} \stackrel{?}{=} r'_{c_i,i}$
and $\forall i, \ \delta t_i \leq \Delta$
Then SUCCESS
Otherwise
FAILURE

Distance Bounding and secret-sharing

How to compute $\mathcal{R}$?

Answer computation

If Bart receives the challenges $(3, 1, 2, 2)$, he replies :

$$\begin{pmatrix} r_{1,1} & \color{red}{r_{1,2}} & r_{1,3} & r_{1,4} \\ r_{2,1} & r_{2,2} & \color{red}{r_{2,3}} & \color{red}{r_{2,4}} \\ \color{red}{r_{3,1}} & r_{3,2} & r_{3,3} & r_{3,4} \end{pmatrix}.$$

Matrix computation

- The two first rows are the output of $PRF(x, N_P, N_V)$
- The last row of $\mathcal{R}$ is given by :

$$\forall i \in [1, 4], r_{3,i} = s_i \oplus r_{1,i} \oplus r_{2,i}.$$

Each column of $\mathcal{R}$ is a **system of shares** obtained from Λ for the coordinate s_i $(s = (s_1, s_2, s_3, s_4))$.

Distance Bounding and secret-sharing

General case

Our protocol can be adapted to any $n \times m$ matrix $\mathcal{R}$:

- Λ is an (n, k) threshold scheme ;
- m is both the number of rounds and the key size.

Our example

- Knowing $r_{1,i}$, $r_{2,i}$ and $r_{3,i} \Rightarrow s_i$.
- Λ is an $(n = 3, k = 3)$ threshold scheme ;
- $m = 4$.

Question

How to safely choose the parameters n and k ?

Adversary model

The adversary, Eve, is a man-in-the-middle with some extra capabilities :

- BD-ADV – Eve is **not able** to distinguish a FAILURE from a SUCCESS of the protocol.
- RES-ADV – Eve knows when there is a FAILURE or a SUCCESS.
- RD-ADV – Eve is able to determine the **result of each round** of interactive phase.

Key recovery attacks

How many shares can Bart provide to Eve ?

Result of the attack

For a given round i, Eve obtains :

- α shares from Bart ;
- How many shares have Eve at the end of the protocol ?
  - For BD-ADV, α.
  - For RD-ADV, $\alpha + 1$.
  - For RES-ADV α but can decimate the key space.

Conclusion

$\alpha = k - 1$ is a bad idea, for RES-ADV and RD-ADV.
Thus, $\alpha \leq k - 2$ is the maximum value to prevent any key leakage.

Key recovery attacks

Mafia Post-ask (fault injection)

Eve

INITIALIZATION

⋮ ⋮ ⋮ ⋮ ⋮

INTERACTIVE

For $i = 1 \cdots m$

$\xleftarrow{c_i}$ Picks $c_i \in [0, n-1]$

Picks $\hat{r}_i$ $\xrightarrow{\hat{r}_i}$

$\xleftarrow{\hat{c}_i}$ Picks $\hat{c}_i \neq c_i$

Sends $r_{\hat{c}_i,i}$ $\xrightarrow{r_{\hat{c}_i,i}}$

RESULT

⋮ ⋮ ⋮ ⋮ ⋮

Key recovery attacks

How many shares can Eve recover ?

Result of the attack

For a given round i, Eve obtains :

- $r_{\hat{c}_i,i}$ from Bart ;
- Is $\hat{r}_i$ a share ?
  - BD-ADV $\longrightarrow$ Eve has no clue if $\hat{r}_i$ is a share or not !
  - RES-ADV $\longrightarrow$ Eve knows if $\hat{r}_i$ is a share or not !
  - RD-ADV $\longrightarrow$ Eve knows if $\hat{r}_i$ is a share or not !

Conclusion

$k = 2$ is a bad idea, for RES-ADV and RD-ADV.
Thus, $k \geq 3$ is the minimal setup to prevent key leakage against any adversary.

What can be achieved ?

Performance of our protocol

Summary

- No key leakage
- Mafia fraud success probability : $\left(\frac{2}{3}\right)^m$.
- Terrorist fraud success probability : $\left(\frac{2}{3}\right)^m$.

Interpretation

The mafia and terrorist fraud have the same probability of success : Involving Bart does not help the adversary !

Comparison

Protocol	BD-ADV	RES-ADV	RD-ADV
Tu and Piramithu	✔	✖	✖
Reid *et al.*	✔	✖ ($*$)	✖
Swiss-Knife	✔	✔	✖/✔ (†)
Bussard and Bagga	✔	✖ $\rightarrow$ ✔ (‡)	✖ $\rightarrow$ ✔ (‡)
TDB ($n \geq 3, k \geq 3$)	✔	✔	✔
TTDB	✔	✔	✔

$*$ Computation of the shares using a pseudo-random permutation protects against RES-ADV. Removed in the final version.

† For the Swiss-knife, everything depends on what can be observed on the RESULT PHASE and how Alice helps Eve.

‡ A modified RESULT PHASE resists to RES-ADV and BD-ADV.

Conclusion

Secret-Sharing :

- + limits the evilness of Bart ;
- - the risk of key information leakage.

Implementation, Implementation...

- Our protocols are not implemented ;
- The RESULT PHASE is critical in the terrorist fraud ;
- Appropriate secret-sharing scheme can solve this problem.

Any questions ?