Legally Fair Contract Signing
Interruption-Immunity Without Third Parties

Houda Ferradi Rémi Géraud Diana Maimut
David Naccache David Pointcheval

June 24, 2015

Outline

1. Preliminaries :
   - Context
   - Motivations
2. Our work
   - The proposed signature paradigm
   - Security proof
3. The concept of Legal Fairness
4. Use case: Legally fair contract signing

Context

- In many operations, such as contract signing, all participants must show their commitment to a given document. **This is done by exchanging digital signatures on the agreed document or by using *co-signature protocol***.
- Typically, co-signature is used for joint bank account management.
- In electronic transactions, *fair exchange* of digital contract signing remains a fundamental problem (Fairness is defined at the next slide).
- In our construction, we mainly focus on contract signing between two parties.

Prior Concepts

The two following properties are desirable in contract signing protocols:

Viability

If both parties follow the protocol properly, then at its termination each party will have his counterpart's signature on the contract.

Fairness

If one party, say Alice, follows the protocol properly then Bob has Alice's signature on the contract iff Alice also has Bob's signature on the contract.

Prior Work

- Ben-Or, Goldreich, Micali and Rivest showed that any viable fair contract signing protocol must rely on a Trusted Third Party (TTP).
- There are 3 degrees of TTP involvement: visible TTPs, semi-TTPs and optimistic protocols.
- The concept of semi-trusted third parties was introduced by Franklin and Reiter.
- Early efforts mainly focused on optimistic protocols to achieve computational fairness *i.e.* "bit-by-bit" secret exchange.

Our Work and New Results

We introduce a novel form of fairness without TTPs called *legal fairness* defined as follows:

Legal Fairness

Any transferable proof of involvement tying one party to a message, also ties the other party to the message.

Idea: In our construction verifiers will be given the means to determine when Alice tries to involve Bob. When this happens, verifiers will be able to contact Bob who will provide the proof of Alice's involvement.

This will be achieved without TTPs

Schnorr Signatures

The proposed signature paradigm is based on Schnorr signatures.

- $\mathbb{G}$ cyclic group of prime order q and g is a generator of $\mathbb{G}$
- secret key: $x \in_{\mathrm{r}} \mathbb{Z}_q^*$
- public key: $y = g^x$
- $\texttt{Sign}(m)$, $m \in \{0,1\}^*$:
  - $k \in_{\mathrm{r}} \mathbb{Z}_q$, $r = g^k$ (commitment) $\xrightarrow{r=g^k}$
  - $e = H(m, r)$ (challenge) $\xleftarrow{e}$
  - $s = k - ex \mod q$ (answer) $\xrightarrow{s=k-ex}$
  - signature is (s, e)
- $\texttt{Verif}(m, (s, e))$:
  - $r = g^s y^e$
  - check $H(m, r) = e$

Schnorr Co-Signatures

Alice		Bob
Read Bob's directory entry		Read Alice's directory entry
$y_{A,B} \leftarrow y_A \times y_B, k_A \in_R \mathbb{Z}_q^*$		$y_{A,B} \leftarrow y_A \times y_B, k_B \in_R \mathbb{Z}_q^*$
$r_A \leftarrow g^{k_A}$		$r_B \leftarrow g^{k_B}$
	$\xleftarrow{\rho}$	$\rho \leftarrow H(0\|r_B)$
	$\xrightarrow{r_A}$	
if $H(0\|r_B) \neq \rho$ **then** abort	$\xleftarrow{r_B}$	
$r \leftarrow r_A \times r_B$		$r \leftarrow r_A \times r_B$
$e \leftarrow H(1\|m\|r)$		$e \leftarrow H(1\|m\|r)$
$s_A \leftarrow k_A - ex_A \bmod q$		$s_B \leftarrow k_B - ex_B \bmod q$
if s_B is incorrect **then** abort	$\xleftarrow{s_B}$	
$s \leftarrow s_A + s_B \bmod q$	$\xrightarrow{s_A}$	$s \leftarrow s_A + s_B \bmod q$
		if s_A is incorrect **then** tant pis!

r, s is verified by checking that: $r = g^s y_{A,B}^e$ and $H(m, r) = e$

Classical Schnorr Signatures and the Forking Lemma

- Pointcheval and Stern 1996: DLP + ROM $\Rightarrow$ Schnorr is secure
- Pointcheval and Stern establish that in the ROM, the opponent can obtain from the forger two valid forgeries $\{\ell, s, e\}$ and $\{\ell, s', e'\}$ for the same oracle query $\{m, r\}$ but with different message digests $e \neq e'$. Consequently, $r = g^s y^{-e} = g^{s'} y^{-e'}$ and from that it becomes straightforward to compute the discrete logarithm of $y = g^x$. Indeed, the previous equation can be rewritten as $y^{e-e'} = g^{s'-s}$, and therefore:

$$y = g^{\frac{s'-s}{e-e'}} \Rightarrow \mathrm{Dlog}_g(y) = \frac{s'-s}{e-e'}$$

- This proof extends to the co-signature protocol introduced in the previous slide (refer to handouts).

Legally Fair Contract Signing

- We will now present our main contribution: the concept of *legally fair contract signatures*.
- The protocol assumes that Bob is stateful. i.e. that Bob keeps in an internal nonvolatile memory $\mathcal{L}$ traces of *problematic* sessions.
- The protocol assumes that Alice uses a second digital signature algorithm σ.

Legally Fair Contract Signing

Alice		Bob
Alice	share m, $y_{A,B}$ $\longleftrightarrow$	**Bob**
$k_A \in_R \mathbb{Z}_q^*, r_A \leftarrow g^{k_A}$		$k_B \in_R \mathbb{Z}_q^*, r_B \leftarrow g^{k_B}$
	$\xleftarrow{\rho}$	$\rho \leftarrow H(0\|r_B)$
$t \leftarrow \sigma(r_A\|\text{Alice}\|\text{Bob})$	$\xrightarrow{r_A, t}$	**if** t is incorrect **then** abort
		store t in $\mathcal{L}$
if $H(0\|r_B) \neq \rho$ **then** abort	$\xleftarrow{r_B}$	
$r \leftarrow r_A \times r_B$		$r \leftarrow r_A \times r_B$
$e \leftarrow H(1\|m\|r\|\text{Alice}\|\text{Bob})$		$e \leftarrow H(1\|m\|r\|\text{Alice}\|\text{Bob})$
$s_A \leftarrow k_A - ex_A \bmod q$		$s_B \leftarrow k_B - ex_B \bmod q$
		store s_B in $\mathcal{L}$
	breakpoint (1)	
if s_B is incorrect **then** abort	$\xleftarrow{s_B}$	
	breakpoint (2)	
	$\xrightarrow{s_A}$	**if** s_A is incorrect **then** abort
$s \leftarrow s_A + s_B \bmod q$		$s \leftarrow s_A + s_B \bmod q$
		if $\{m, r, s\}$ is valid erase $\mathcal{L}$

Intuition of the Legal Fairness Proof

To optimally follow our argument, refer to the description of the protocol in the handouts. We present here the intuition, the formal proof is in the paper.

- Nothing bad can possibly happen *before* breakpoint (1). Because before breakpoint (1) no information depending on m was released by any of the parties.
- After breakpoint (1) Bob can misbehave (go silent or send a bad s_B). In such a case Alice will detect this and punish him (she will just shut-up).
- If Bob did not misbehave we hit the core issue: breakpoint (2) is critical. Here Alice has the final say. She can hence stop sending information (or send wrong information). We need to show that if this happens Bob can either (A) deny involvement or (B) involve Alice as well. Outcomes (A) or (B) depend on the way in which Alice tries to use the information she has in her possession to involve Bob.

Case Ⓐ: Denial by Bob

Alice exhibits r and s_B:

The Bob will pretend that:

1. Alice picked s_B, r at random
2. Alice computed $e \leftarrow H(1\|m\|r\|\text{Alice}\|\text{Bob})$
3. Alice computed $r_B \leftarrow g^{s_B} y_B^e$
4. Alice computed $r_A \leftarrow r \times r_B^{-1}$
5. Alice signed $t \leftarrow \sigma(r_A\|\text{Alice}\|\text{Bob})$.

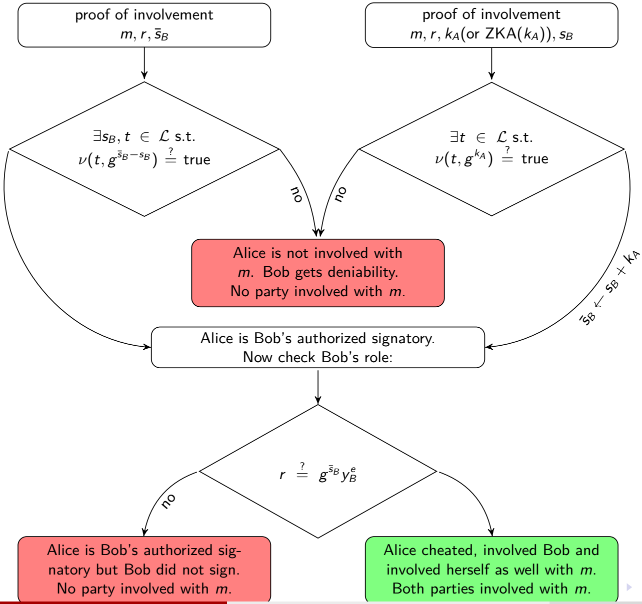
Indeed $g^{s_B} y_B^e = r_B$ and this looks exactly as if Bob produced s_B, r_B.

Note that this would *not* be the case if Alice manages to prove that r_A is not random, *i.e.* that she knows the DL of r_A. This is case Ⓑ.

Case Ⓑ: Alice Shoots a Bullet Into Her Own Leg

Alice proves that she knows the DL of r_A:
A verifier seeing that is instructed to contact Bob and ask him for t.
The verifier now has a signature t of r_A and a proof that Alice knows the DL of r_A.
We *legally define* this state of knowledge as a deliberate agreement on Alice's behalf to any message signed by Bob, and in particular m.

Schematically



Further Research

Can you generalize the above to more than two parties?