

---

# Practical Verifiable Encryption and Decryption of Discrete Logarithms

Jan Camenisch

*IBM Zurich Research Lab*

Victor Shoup

*New York University*

# Verifiable encryption of discrete logs

---

- ▶ Three players:  $A$ ,  $B$ , and  $T$
- ▶  $T$  has  $(PK, SK)$  for an encryption scheme
- ▶  $A, B$  have group elements  $\gamma, \delta \in G$ , with  $|G|$  prime
- ▶  $A$  has  $x$  such that  $\gamma^x = \delta$
- ▶  $A$  encrypts  $x$  under  $PK$ , obtaining  $\psi$ , and proves to  $B$  that  $\psi$  is a correct encryption of  $\log_{\gamma} \delta$

# Verifiable decryption . . .

---

- ▶ of discrete logarithms:
  - $B$  has a ciphertext  $\psi$  and  $\gamma, \delta \in G$
  - $T$  proves to  $B$  *whether or not*  $\psi$  decrypts to  $\log_{\gamma} \delta$
- ▶ of messages:
  - $B$  has a ciphertext  $\psi$  and a message  $m$
  - $T$  proves to  $B$  *whether or not*  $\psi$  decrypts to  $m$

# Generalization to representations

---

- ▶ Verifiably encrypt/decrypt a *representation* of  $\delta$  with respect to  $\gamma_1, \dots, \gamma_k$ , i.e.,  $x_1, \dots, x_k$  such that  $\delta = \gamma_1^{x_1} \cdots \gamma_k^{x_k}$ .
- ▶ more general verifiable encryption:
  - $A$  makes a Pedersen commitment  $C$  to a secret  $s$ , and an encryption  $\psi$  under PK of an opening of  $C$
  - $A$  proves to  $B$  that  $C$  is a commitment to a value that satisfies some property
  - $A$  proves to  $B$  that  $\psi$  is an encryption of an opening of  $C$

## Application: fair key escrow

---

$A$  encrypts its own secret key under  $T$ 's public key, and proves to  $B$  that this was done correctly.

- ▶ For Schnorr/DSS signatures, or for ElGamal encryption, use verifiable encryption of DL
- ▶ To implement  $T$ 's decryption policy, we need an encryption scheme with a *label*
- ▶ To force  $T$  to behave correctly, use verifiable decryption
- ▶ For other public-key primitives, use the “commit, encrypt, and prove” technique

# Application: optimistic fair exchange

---

$A$  and  $B$  want to exchange some valuable digital data (e.g., signatures on a contract, e-cash), but in a *fair* way.

A trusted third party  $T$  is used, but only when necessary [Asokan, Shoup, Waidner '98].

- ▶  $A$  and  $B$  first verifiably encrypt their secrets
- ▶ After verification, each party reveals its secret
- ▶ If either party backs out, invoke  $T$
- ▶ Label needed to implement protocol

# Application: publicly verifiable secret sharing

---

A “dealer” shares a secret with proxies  $P_1, \dots, P_n$ , in such a way that another party  $Q$  can verify that the sharing was done correctly [Stadler '96].

- ▶ Dealer shares his secret (Shamir)
- ▶ Dealer encrypts  $P_i$ 's share under  $P_i$ 's public key, and gives these ciphertexts to  $Q$ , along with commitments to the shares and the coefficients of the “blinding” polynomial
- ▶ Dealer proves to  $Q$  that the ciphertexts encrypt openings of the share commitments

# Application: UC commitments

---

Universally composable (UC) commitments [Canetti, Fischlin 2001] act like “ideal commitments” implemented using a trusted party.

A common reference string (CRS) is needed.

- ▶ CRS is a public key for an “imaginary” party  $T$  and a Pedersen commitment scheme
- ▶ To commit: make a Pedersen commitment  $C$  and verifiable encrypt its opening
- ▶ To open: reveal the opening of  $C$

## Application: UC commitments (cont'd)

---

- ▶ Simulator can “extract” commitments made by bad guys (using  $T$ 's secret key)
- ▶ Simulator can “equivocate” the commitments made by good guys (using the Pedersen trapdoor)
- ▶ Prove properties on committed values using established techniques

Compare with [Canetti, Fischlin 2001],  
[Damgaard, Nielsen 2001] ...

## Application: confirmer signatures

---

In a confirmer signature [Chaum '84], a party  $A$  creates an “opaque signature”  $\psi$  on a message  $m$ .

The signer may confirm the validity of  $\psi$ .

A trusted third party  $T$ , can *confirm or deny* the validity of  $\psi$ , or *convert*  $\psi$  to an ordinary signature.

# Application: confirmer signatures (cont'd)

---

A simple Schnorr/DSS implementation:

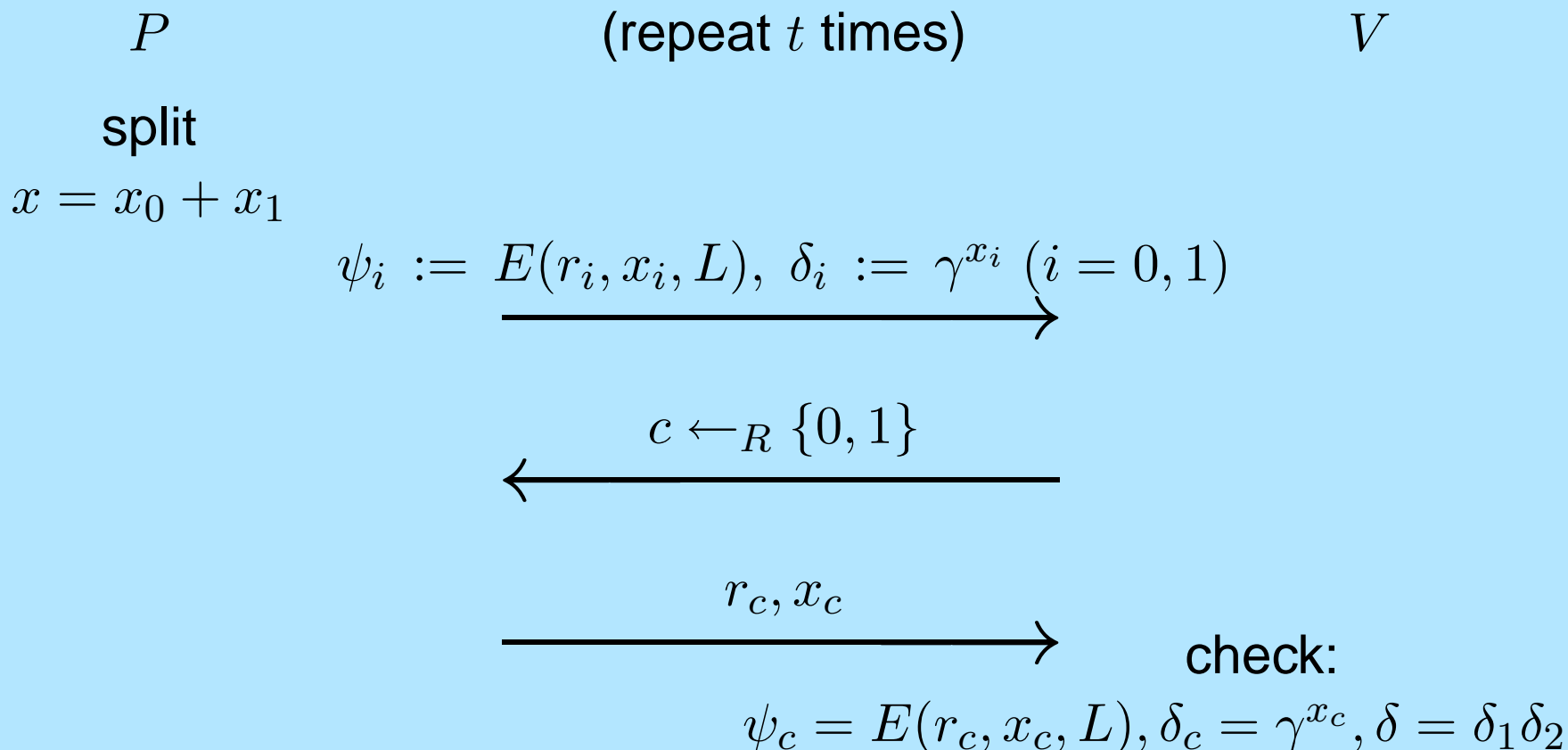
- ▶  $T$  has an encryption key
- ▶ To sign:
  - $A$  makes a commitment  $C$  to a value that will yield an ordinary signature, and
  - an encryption  $\psi$  of the opening of  $C$
- ▶  $A$  can use DL verifiable encryption to confirm
- ▶  $T$  can use DL verifiable decryption to confirm/deny or convert

# Previous work: verifiable encryption

---

“Cut and choose” [Asokan, Shoup, Waidner '98]

Common input:  $\gamma, \delta \in G$ ; Prover:  $x = \log_\gamma \delta$



# Previous work: verifiable encryption (cont'd)

---

“Double discrete log / cut and choose” [Stadler '96, Young & Yung '98]

	ASW	S/YY
chosen ciphertext security	+	-
labels	+	-
separability	+	-
efficiency	-	-

# Previous work: verifiable decryption

---

... not much

# Previous work: verifiable decryption

---

... not much

Verifiable decryption of messages using original Cramer-Shoup encryption is straightforward, but

- ▶ does not allow verifiable encryption of DL,
- ▶ does not generalize to verifiable decryption of DL.

# Our contributions

---

A new encryption scheme with protocols for verifiable encryption and decryption of discrete logarithms and representations:

- ▶ chosen ciphertext security (with labels)
- ▶ separable
- ▶ efficient — standard “ $\Sigma$  protocols,” a handful of exponentiations, “stand-alone” ciphertext
- ▶ no random oracles

# A convergence of technologies

---

- ▶ Proofs of multiplicative relations among committed integers based on the Strong RSA assumption [Fujisaki, Okamoto '97]
- ▶ Encryption based on the Decisional Composite Residuosity (DCR) assumption [Paillier '99]
- ▶ Encryption using Universal Hash Proofs [Cramer, Shoup '02]

# The encryption scheme

---

- ▶  $p, q, p', q'$  are distinct odd primes with  $p = 2p' + 1$  and  $q = 2q' + 1$ , where  $p'$  and  $q'$  are both  $\ell$  bits long
- ▶  $n := pq, n' := p'q', \zeta := (1 + n \bmod n^2) \in \mathbb{Z}_{n^2}^*$
- ▶ Note:  $\mathbb{Z}_{n^2}^* \cong \mathbb{Z}_2 \times \mathbb{Z}_2 \times \mathbb{Z}_n \times \mathbb{Z}_{n'}$ .
- ▶ Note:  $\zeta^x = (1 + xn \bmod n^2)$
- ▶ DCR assumption: hard to distinguish  $\mathbb{Z}_{n^2}^*$  from  $(\mathbb{Z}_{n^2}^*)^n$
- ▶  $H$  is a collision resistant,  $\ell$ -bit hash

# The encryption scheme (cont'd)

---

**Key Gen:**  $x_1, x_2, x_3 \leftarrow_R \mathbb{Z}_{[0, n^2/4)}$ ,  $g' \leftarrow_R \mathbb{Z}_{n^2}^*$ ,  
 $g \leftarrow (g')^{2n}$ ,  $y_1 \leftarrow g^{x_1}$ ,  $y_2 \leftarrow g^{x_2}$ , and  $y_3 \leftarrow g^{x_3}$ ;  
 $\text{PK} = (n, H; g, y_1, y_2, y_3)$ ,  $\text{SK} = (x_1, x_2, x_3)$

## The encryption scheme (cont'd)

---

**Key Gen:**  $x_1, x_2, x_3 \leftarrow_R \mathbb{Z}_{[0, n^2/4)}$ ,  $g' \leftarrow_R \mathbb{Z}_{n^2}^*$ ,  
 $g \leftarrow (g')^{2n}$ ,  $y_1 \leftarrow g^{x_1}$ ,  $y_2 \leftarrow g^{x_2}$ , and  $y_3 \leftarrow g^{x_3}$ ;  
 $\text{PK} = (n, H; g, y_1, y_2, y_3)$ ,  $\text{SK} = (x_1, x_2, x_3)$

**To encrypt**  $m \in \mathbb{Z}_{(-n/2, n/2)}$  **w/ label**  $L \in \{0, 1\}^*$ :

$$r \leftarrow_R \mathbb{Z}_{[0, n/4)}, u \leftarrow g^r, e \leftarrow y_1^r \zeta^m,$$

$$\hat{y} \leftarrow y_2 y_3^{H(u, e, L)}, v \leftarrow \hat{y}^r; \psi := (u, e, v)$$

# The encryption scheme (cont'd)

---

**Key Gen:**  $x_1, x_2, x_3 \leftarrow_R \mathbb{Z}_{[0, n^2/4)}$ ,  $g' \leftarrow_R \mathbb{Z}_{n^2}^*$ ,  
 $g \leftarrow (g')^{2n}$ ,  $y_1 \leftarrow g^{x_1}$ ,  $y_2 \leftarrow g^{x_2}$ , and  $y_3 \leftarrow g^{x_3}$ ;  
 $\text{PK} = (n, H; g, y_1, y_2, y_3)$ ,  $\text{SK} = (x_1, x_2, x_3)$

**To encrypt**  $m \in \mathbb{Z}_{(-n/2, n/2)}$  **w/ label**  $L \in \{0, 1\}^*$ :

$r \leftarrow_R \mathbb{Z}_{[0, n/4)}$ ,  $u \leftarrow g^r$ ,  $e \leftarrow y_1^r \zeta^m$ ,  
 $\hat{y} \leftarrow y_2 y_3^{H(u, e, L)}$ ,  $v \leftarrow \hat{y}^r$ ;  $\psi := (u, e, v)$

**To decrypt**  $\psi := (u, e, v)$  **w/ label**  $L$ :

if  $u^{2(x_2 + H(u, e, L)x_3)} = v^2$ , and  
 $((e/u^{x_1})^2)^{2^{-1} \bmod n} = \zeta^m$  for  $m \in \mathbb{Z}_{(-n/2, n/2)}$ ,  
output  $m$ ; otherwise, reject

## The encryption scheme (cont'd)

---

**Theorem.** *Scheme is secure against adaptive chosen ciphertext attack (but “benignly malleable”), assuming DCR and  $H$  is collision resistant*

*Proof.* Similar to proof in [Cramer, Shoup '02] (see paper).

Note: a similar scheme appears in [Gennaro, Lindell '03]

# Verifiable encryption of DL

---

Common inputs:  $PK := (H, n; g, y_1, y_2, y_3)$ ,  $\tilde{g}, \tilde{h} \in \mathbb{Z}_n^*$  of order  $n'$ ,  $\gamma, \delta \in G$ ,  $\psi := (u, e, v)$ , and label  $L$ ; let  $\hat{y} := y_2 y_3^{H(u,e,L)}$ .

Prover:  $m = \log_\gamma \delta \in \mathbb{Z}_{[0, n/2)}$  and  $r \in \mathbb{Z}_{[0, n/4)}$  such that  $u = g^r$ ,  $e = y_1^r \zeta^m$ ,  $v = \hat{y}^r$ .

Prover chooses  $s \in_R \mathbb{Z}_{[0, n/4)}$ , computes  $\tilde{v} \leftarrow \tilde{g}^m \tilde{h}^s$ , sends  $\tilde{v}$  to the verifier, and then:

$$PK\{(r, m, s) : u^2 = g^{2r} \wedge e^2 = y_1^{2r} \zeta^{2m} \wedge v^2 = \hat{y}^{2r} \wedge \delta = \gamma^m \wedge \tilde{v} = \tilde{g}^m \tilde{h}^s\} .$$

# Verifiable encryption of DL (cont'd)

$P$

$V$

gen random  $s, r', m', s'$

$$\tilde{v} := \tilde{g}^m \tilde{h}^s, u' := g^{2r'}, e' := y_1^{2r'} \zeta^{2m'},$$

$$v' := \hat{y}^{2r'}, \delta' := \gamma^{m'}, \tilde{v}' := \tilde{g}^{m'} \tilde{h}^{s'}$$



random  $c$



$$\bar{r} := r' - cr, \bar{m} := m' - cm, \bar{s} := s' - cs \text{ (in } \mathbb{Z}\text{)}$$



check:

$$u' = u^{2c} g^{2\bar{r}}, e' = e^{2c} y_1^{2\bar{r}} \zeta^{2\bar{m}}, v' = v^{2c} \hat{y}^{2\bar{r}},$$

$$\delta' = \delta^c \gamma^{\bar{m}}, \tilde{v}' = \tilde{v}^c \tilde{g}^{\bar{m}} \tilde{h}^{\bar{s}}, \bar{m} \in \mathbb{Z}_{(-n/4, n/4)}$$

## Verifiable encryption of DL (cont'd)

---

**Theorem.** *The above protocol is a special honest-verifier zero knowledge protocol for proving that a ciphertext encrypts a discrete logarithm whose soundness follows from the Strong RSA assumption.*

**Strong RSA assumption:** given  $n$  and  $x \in \mathbb{Z}_n^*$ , it is hard to compute  $y \in \mathbb{Z}_n^*$  and  $e > 1$  such that  $y^e = x$

# Proof of soundness

---

Rewind the prover, obtaining  $\Delta r, \Delta s, \Delta m, \Delta c \in \mathbb{Z}$  such that

$$|\Delta m| < n/2, \quad 0 < \Delta c < \min\{p', q', |G|\},$$

and

$$\begin{aligned} u^{2\Delta c} &= g^{2\Delta r}, & e^{2\Delta c} &= y_1^{2\Delta r} \zeta^{2\Delta m}, & v^{2\Delta c} &= \hat{y}^{2\Delta r}, \\ \delta^{\Delta c} &= \gamma^{\Delta m}, & \tilde{v}^{\Delta c} &= \tilde{g}^{\Delta m} \tilde{h}^{\Delta s}. \end{aligned}$$

# Proof of soundness

---

Rewind the prover, obtaining  $\Delta r, \Delta s, \Delta m, \Delta c \in \mathbb{Z}$  such that

$$|\Delta m| < n/2, \quad 0 < \Delta c < \min\{p', q', |G|\},$$

and

$$u^{2\Delta c} = g^{2\Delta r}, \quad e^{2\Delta c} = y_1^{2\Delta r} \zeta^{2\Delta m}, \quad v^{2\Delta c} = \hat{y}^{2\Delta r},$$
$$\delta^{\Delta c} = \gamma^{\Delta m}, \quad \tilde{v}^{\Delta c} = \tilde{g}^{\Delta m} \tilde{h}^{\Delta s}.$$

★ Strong RSA  $\implies \Delta c \mid \Delta m$  and  $\Delta c \mid \Delta s$ .

## Proof of soundness (cont'd)

---

Then we have

$$\begin{aligned}u^2 &= g^{2r}, \\e^2 &= y_1^{2r} \zeta^{2m}, \\v^2 &= \hat{y}^{2r}, \\ \delta &= \gamma^m,\end{aligned}$$

where  $m := \Delta m / \Delta c \in \mathbb{Z}_{(-n/2, n/2)}$ , and  $r := \Delta r \cdot (\Delta c^{-1} \bmod nn')$ .

It follows that  $\psi$  will decrypt to  $m = \log_{\gamma} \delta$ .

# Verifiable decryption

---

... see paper

Techniques:

- ▶ a new protocol for proving the *inequality* of two DLs
- ▶ efficient “interval” proofs [Boudot '00].
- ▶ efficient “or” proofs [Cramer, Damgaard, Schoenmakers '94]

# Conclusions

---

We have discussed protocols for verifiable encryption and decryption of discrete logarithms, which have application to

- ▶ fair key escrow,
- ▶ optimistic fair exchange,
- ▶ publicly verifiable secret sharing,
- ▶ universally composable commitments,
- ▶ confirmer signatures.

## Conclusions (cont'd)

---

We have presented a new encryption scheme and protocols for both verifiable encryption and decryption of DLs that

- ▶ provide chosen ciphertext security (with labels),
- ▶ are separable (i.e., flexible),
- ▶ are efficient (no “cut and choose”),
- ▶ are provably secure (no random oracles).