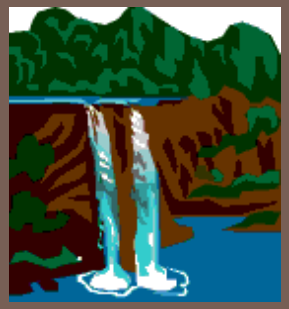
LEAKAGE-RESILIENCE AND THE BOUNDED-RETRIEVAL MODEL



ICITS'09

Speaker: Yevgeniy Dodis (NYU)

Motivation: Leakage-Resilient Crypto

- Security proofs in crypto assume idealized adversarial model.
  - e.g. adversary sees public-keys, ciphertexts but not secret-keys.
- Reality: schemes broken using "key-leakage attacks".
  - Side-channels: timing, power consumption, heat, acoustics, radiation.
  - The cold-boot attack.
  - Hackers, Malware, Viruses.
- Usual Crypto Response: Not our problem.
  - Blame the Electrical Engineers, OS programmers…
- Leakage-Resilient Crypto: Let's try to help.
  - Primitives that provably allow some leakage of secret key.
  - Assume leakage is arbitrary but incomplete.

Models of Leakage Resilience

- Adversary can learn *any efficiently computable function* $g : \{0,1\}^* \rightarrow \{0,1\}^L$ of the secret key. L = Leakage Bound.
- **Relative Leakage Model** [AGV09,DKL09,NS09,DGK+10]:
  - "Standard" cryptosystem with small keys (e.g. 1,024 bits).
  - Leakage L is a large portion of key size (e.g. 50% of key size).
- **Bounded Retrieval Model** [Dzi06,CLW06,…,ADW09,ADN+09]:
  - Leakage L is a parameter. Can be large. (e.g. few bits or many Gigabytes).
  - Increase sk size to allow L bits of leakage.
  - System must remain efficient as L grows: Public keys, ciphertexts, signatures, enc-dec, sig-ver times, etc. should be small, independent of L.

Why design schemes for the BRM?

- Security against Hackers/Malware/Trojans/Viruses:
  - Attacker can download arbitrary info from compromised system.
  - Leakage is large, but still bounded (e.g. < 10 GB).
    - Bandwidth too low, Cost too high, System security may detect.
  - Protect against such attacks by making secret key large.
    - OK since storage is cheap. Everything else needs to remain efficient!

- Security against side-channel attacks:
  - After many physical measurements, overall leakage may be large.
  - Still may be reasonable that it is bounded on absolute scale.
  - How “bounded” is it? Varies! (few Kb – few Mb).

Prior Work on Leakage Resilience

- **Restricted classes of leakage functions.**
  - Individual bits of memory [CDH+00, DSS01,KZ03]. Individual wires of comp [ISW03]
  - "Only Computation Leaks Information" [MR04, DP08, Pie09, DP10]
  - Low Complexity functions [FRT09]

  Does not seem applicable to e.g. hacking/malware attacks.
- **Relative Leakage Model.**
  - Symmetric-Key Authenticated Encryption [DKL09]
  - Public-Key Signatures [ADW09, KV09, DHLW09]
  - Public-Key Encryption [AGV09, NS09, DGK+10]
- **Bounded Retrieval Model.**
  - Symmetric-Key Identification, Authenticated Key Agreement [Dzi06,CDD+07]
  - Secret Sharing [DP08] , Password Authentication [CLW06]
  - Public-Key Authenticated Key Agreement, Identification, "Entropic" Sigs [ADW09]
  - Public-Key Encryption (and IBE) [ADN+09].

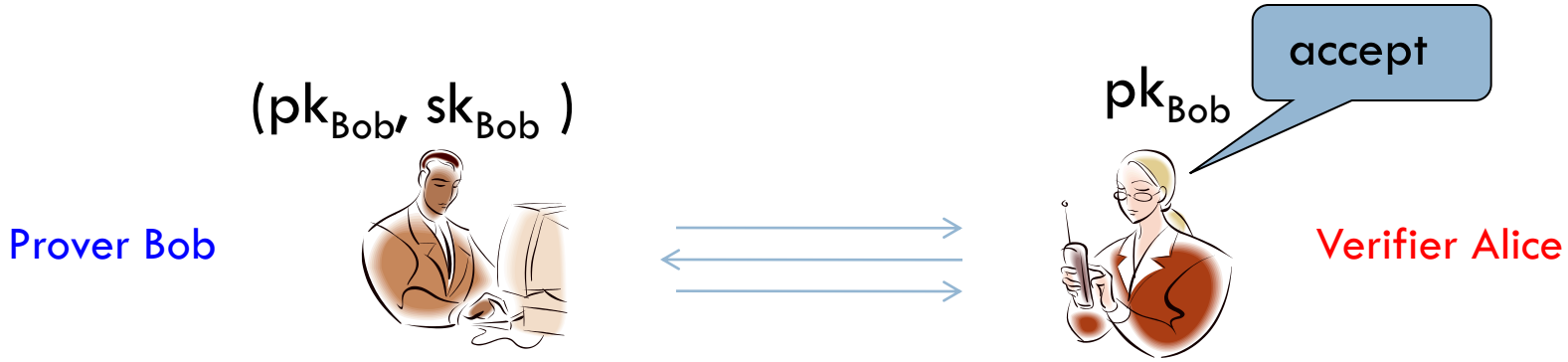
Prior Work on Leakage Resilience

- **Restricted classes of leakage functions.**
  - Individual bits of memory [CDH+00, DSS01,KZ03]. Individual wires of comp [ISW03]
  - "Only Computation Leaks Information" [MR04, DP08, Pie09, DP10]
  - Low Complexity functions [FRT09]
- I will try to emphasize information-theoretic techniques throughout the presentation
- Symmetric-Key Identification, Authenticated Key Agreement [Dzi06,CDD+07]
- Secret Sharing [DP08] , Password Authentication [CLW06]
- Public-Key Authenticated Key Agreement, Identification, "Entropic" Sigs [ADW09]
- Public-Key Encryption (and IBE) [ADN+09].

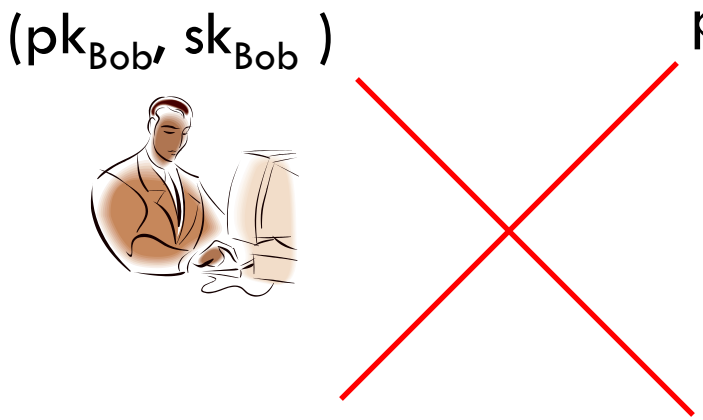
Roadmap of This Survey

- Relative Leakage Model
  - ☑ Password Authentication and OWFs
  - Identification Schemes
  - Signature Schemes
  - Encryption Schemes (and IBE)
  - Authenticated Key Agreement (AKA)
- Bounded Retrieval Model
  - From Relative to Absolute leakage

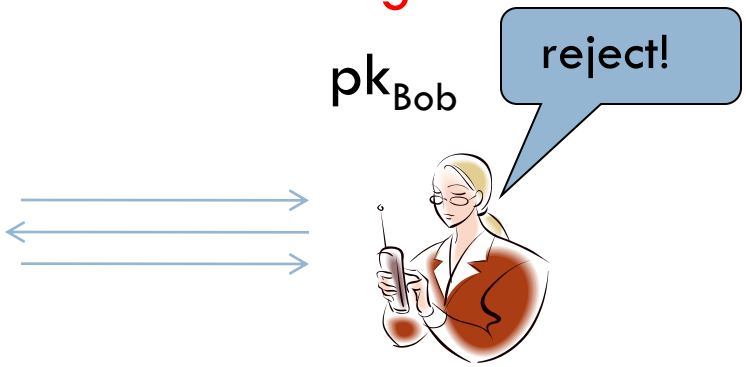
Password Authentication Schemes



No Learning Stage

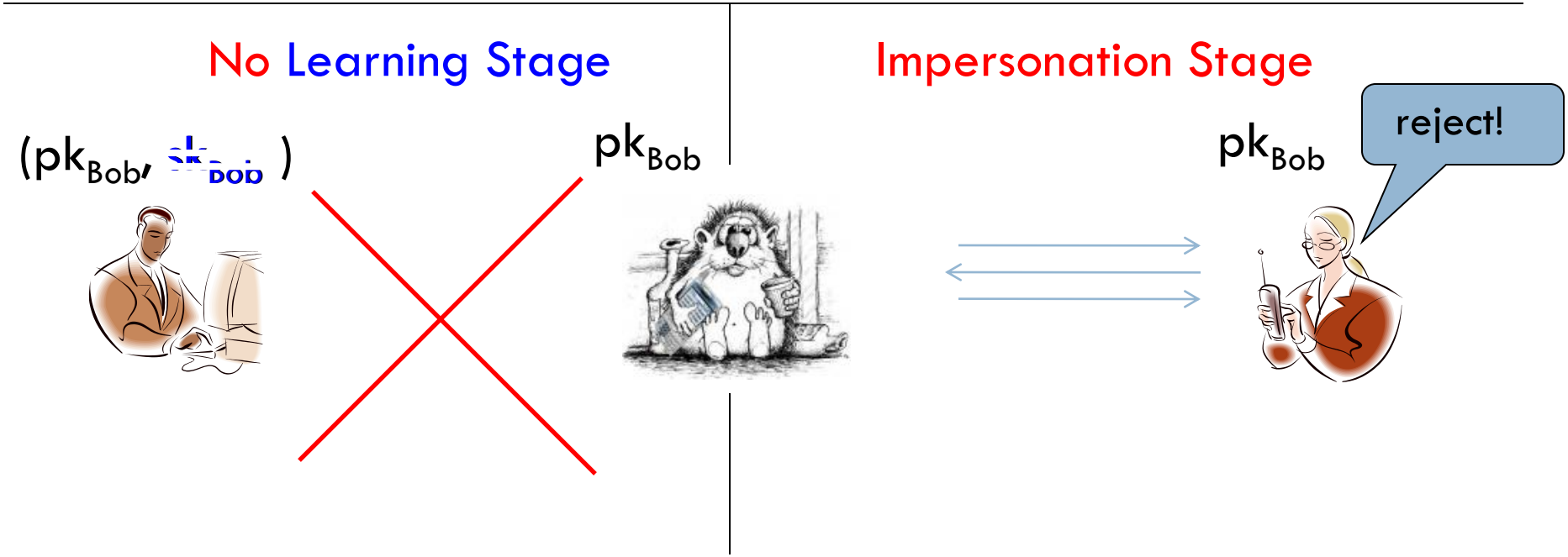


Impersonation Stage

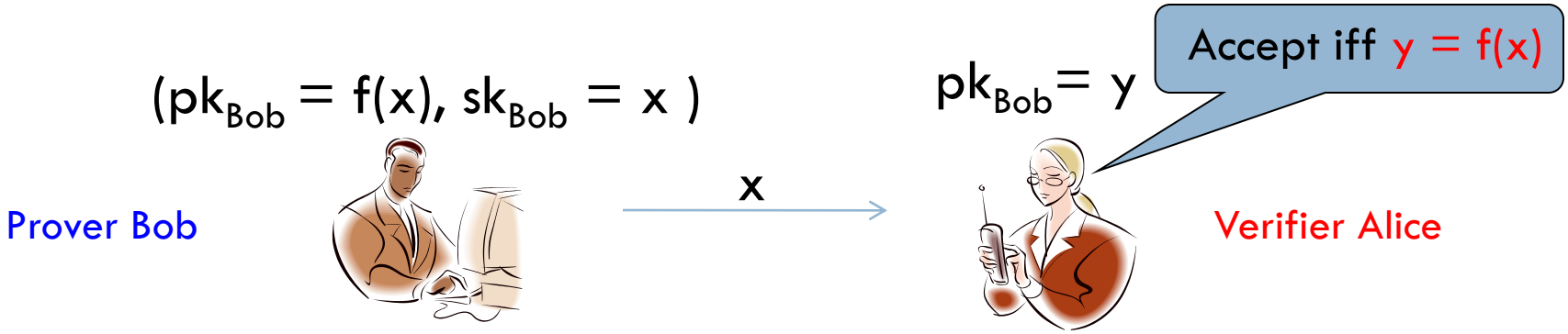


Leakage-Resilient PA Schemes

- Bob's key can leak !!!
  - Allow up to L bits of leakage about sk_{Bob}
  - Building L-LR PA Schemes?



Using One-Way Functions



- Standard OWF: given $y = f(x)$, hard to get x' s.t. $f(x')=y$
  - Suffices for regular PA security
- L-LR OWF: given $y = f(x)$ and L bits of leakage about x, hard to get any x' s.t. $f(x')=y$
  - Does not follow from general OWFs (easy counter-examples)
  - Follows from Second Preimage Resistant Functions (SPRF) !

Second Preimage Resistant Functions

- OWF: given $y = f(x)$, hard to get x' s.t. $f(x')=y$
- L-LR OWF: given $y = f(x)$ and L bits of leakage about x, hard to get any x' s.t. $f(x')=y$
- SPRF: given x, hard to get $x' \neq x$ s.t. $f(x')=f(x)$
  - Non-triviality: input length $n >$ output length k
  - Relaxation of collision-resistance, but (in theory) can build from OWFs for any $n = poly(k)$ [Rom90]
  - Example: $f(x_1,\ldots,x_n)=g_1^{x_1}\ldots g_n^{x_n}$ is SPR under Discrete Log
  - Folklore: f SPRF and $n > k + \lambda$ (sec. param.) $\Rightarrow$ f is OWF
- Theorem: f SPRF and $n > L + k + \lambda \Rightarrow f$ is L-LR-OWF

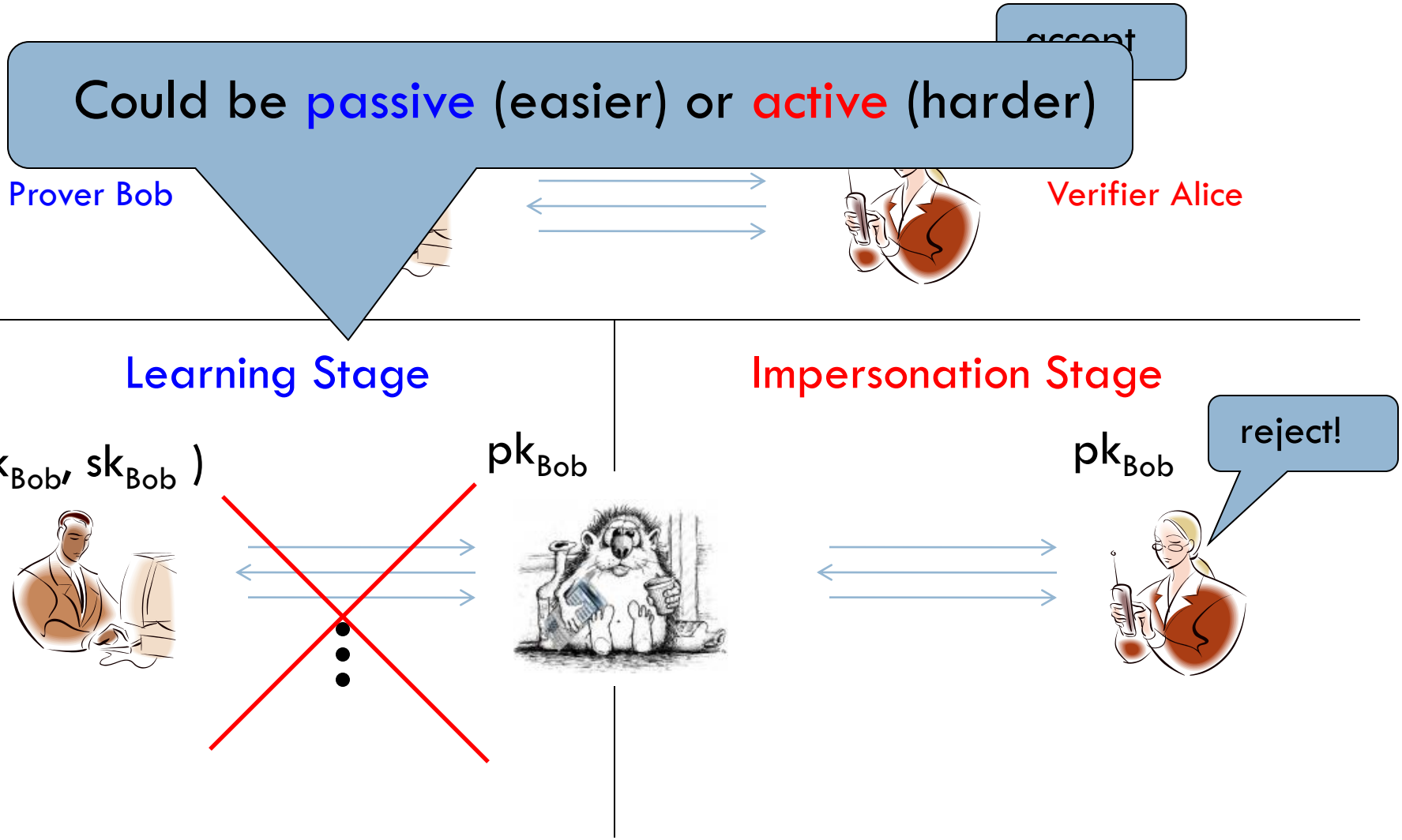
Proof that SPRF is LR-OWF [ADW09]

- Theorem: f SPRF and $n > L + k + \lambda \Rightarrow f$ is L-LR-OWF
  - Assume $\Pr[A(f(x), Leak(x)) = x' \text{ and } f(x')=f(x)] > \varepsilon$
  - Construct $B(x)$ breaking SPR: "return $A(f(x), Leak(x))$"
  - $\Pr[B \text{ wins}] = \Pr[A \text{ wins and } x' \neq x] \geq \Pr[A \text{ wins}] - \Pr[x' = x]$
  - But A only has $|f(x)|+|Leak(x)|<|x|-\lambda$ bits of info about x
  - Thus, $\Pr[x' = x] \leq (1/2)^{\lambda}$, even if A was unbounded
  - Hence, $\Pr[B \text{ wins}] \geq \varepsilon - (1/2)^{\lambda}$ is non-negligible
- Corollary: L-LR-OWFs $\Leftrightarrow$ OWFs, even for $L = n - O(\lambda)$

Roadmap of This Survey

- Relative Leakage Model
  - Password Authentication and OWFs
  - ☑ Identification Schemes
  - Signature Schemes
  - Encryption Schemes (and IBE)
  - Authenticated Key Agreement (AKA)
- Bounded Retrieval Model
  - From Relative to Absolute leakage

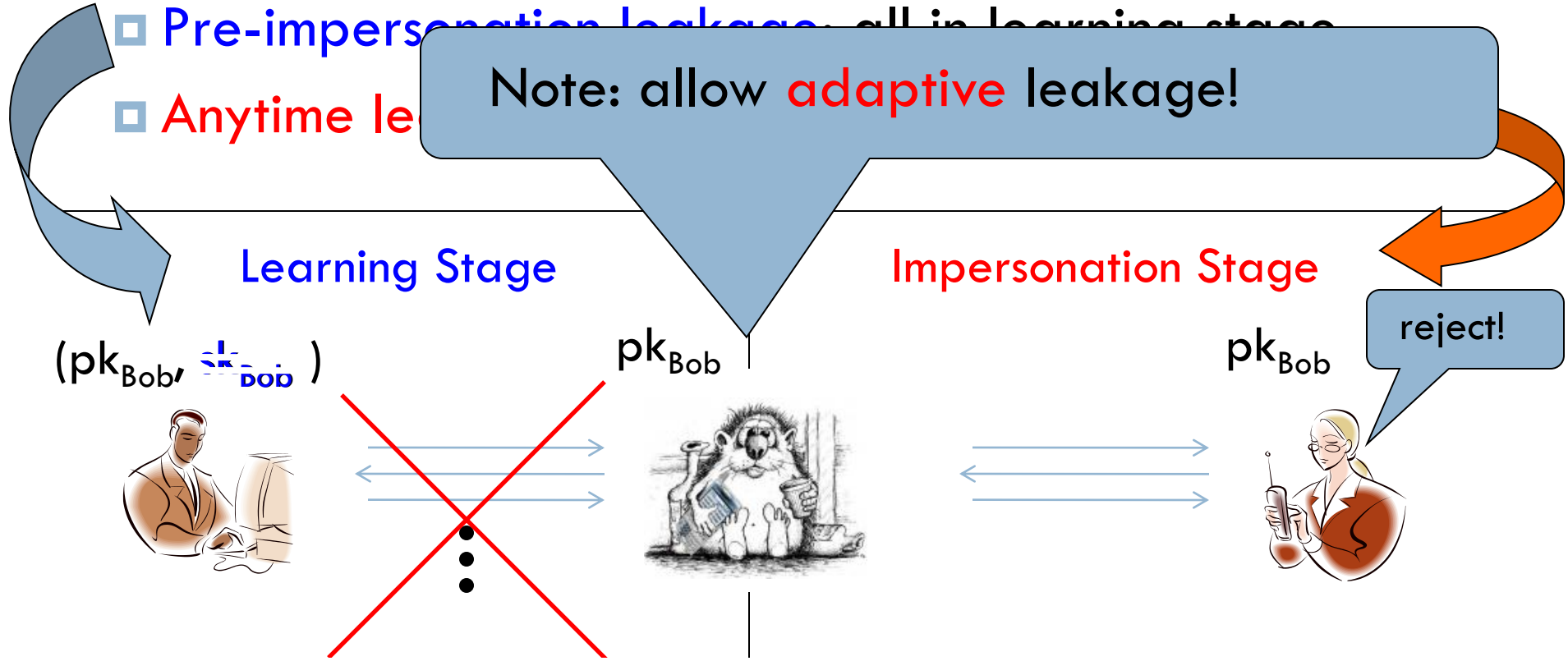
Identification Schemes



Leakage-Resilient Identification

- Bob's key can leak !!!
  - Pre-impersonation leakage: all in learning stage
  - Anytime leakage

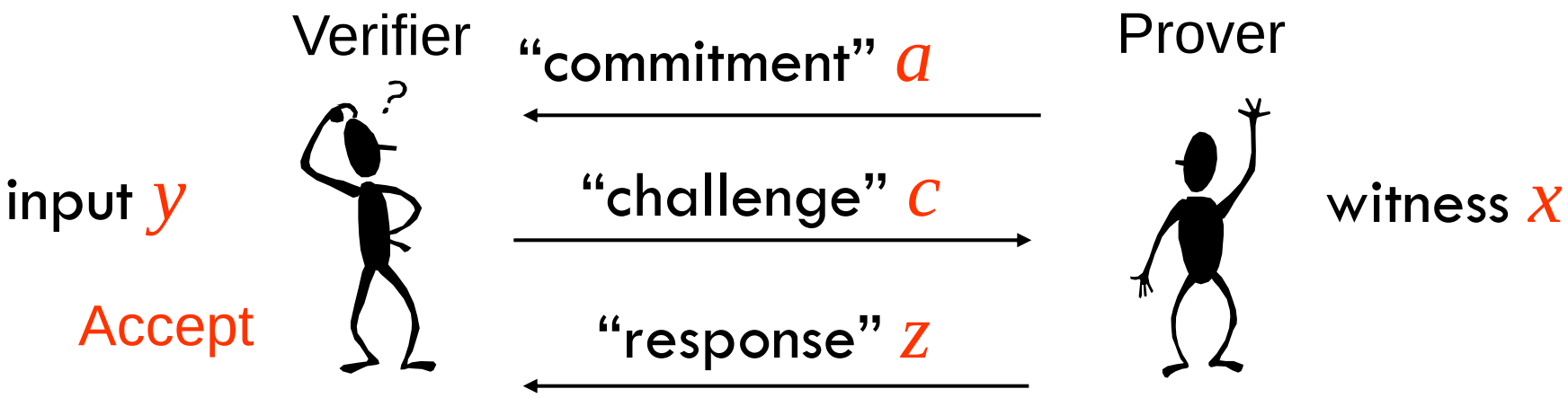
Note: allow adaptive leakage!



- Leads to defining (L_1, L_2)-LR ID schemes [ADW09]

Sigma-Protocols

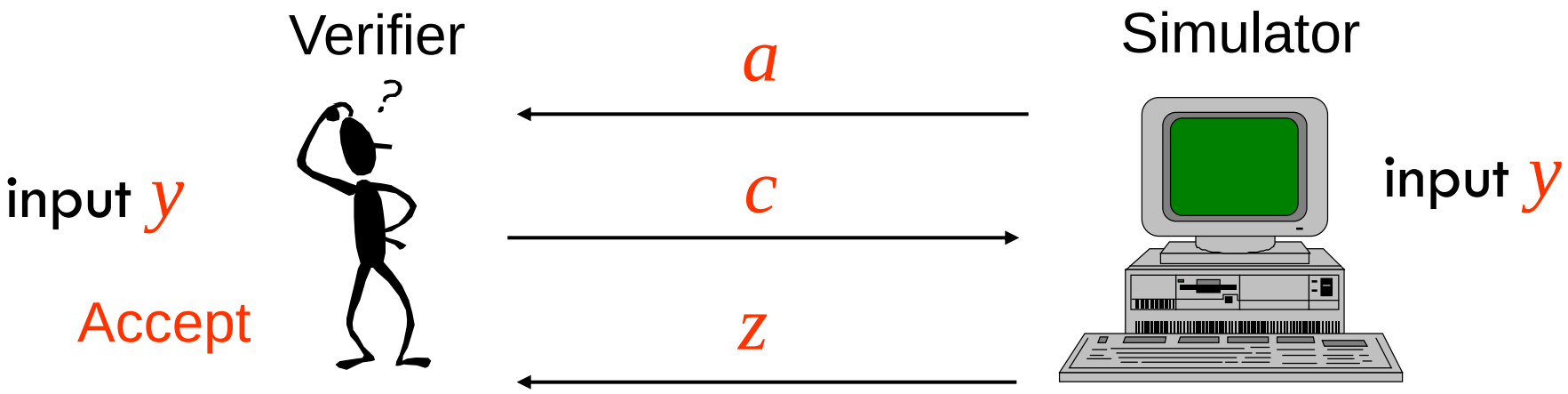
- "Special" 3-round HVZK PoK:



- Special HVZK:
  - Know c in advance $\Rightarrow$ can fake proofs for any y, even without knowing x

Sigma-Protocols

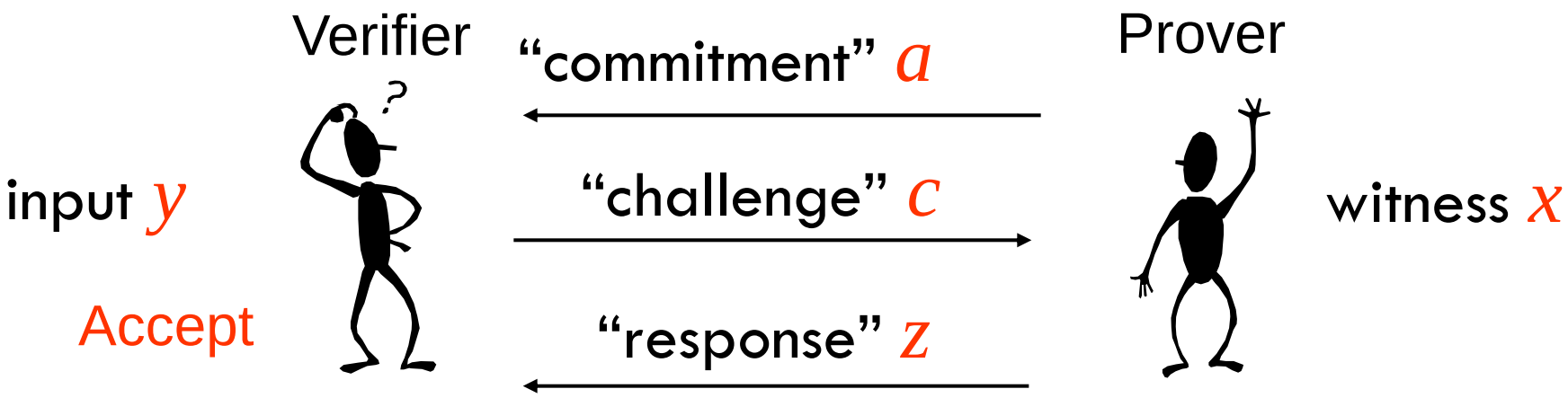
- "Special" 3-round HVZK PoK:



- Special HVZK:
  - Know c in advance $\Rightarrow$ can fake proofs for any y, even without knowing x
  - Implies passive security: Sim picks random c and fakes consistent (a,z)
  - Not good for active security: what if c depends on a ?

Sigma-Protocols

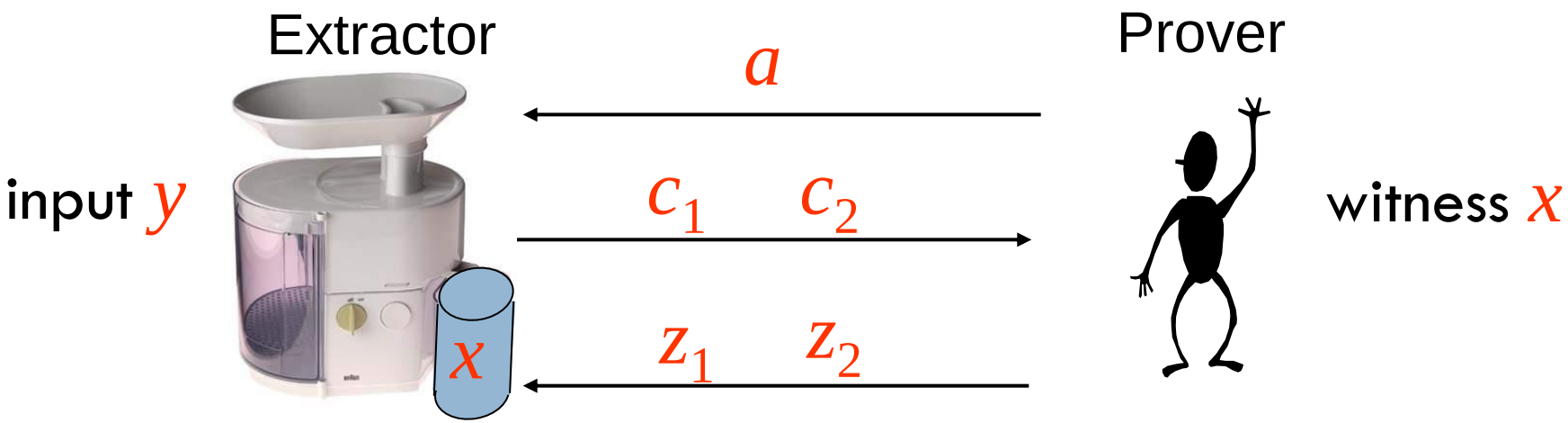
- "Special" 3-round HVZK PoK:



- Special HVZK:
  - Know c in advance $\Rightarrow$ can fake proofs for any y, even without knowing x
- Special Soundness:
  - Know two distinct conversations with same a $\Rightarrow$ recover witness x

Sigma-Protocols

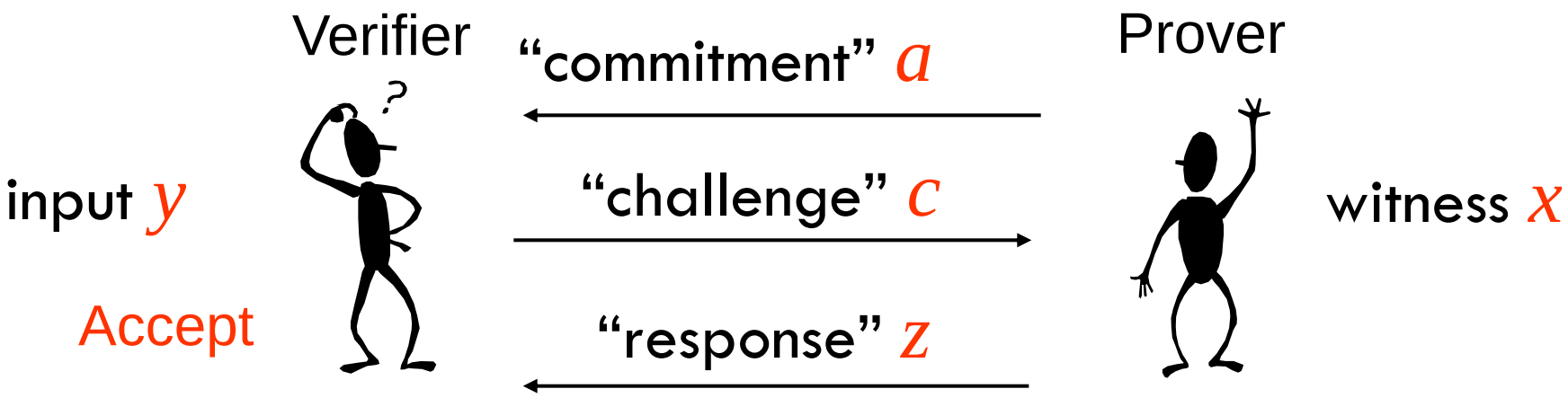
- "Special" 3-round HVZK PoK:



- Special HVZK:
  - Know c in advance $\Rightarrow$ can fake proofs for any y, even without knowing x
- Special Soundness:
  - Know two distinct conversations with same a $\Rightarrow$ recover witness x
  - Implies soundness/knowledge error = 1/#challenges

Sigma-Protocols

- "Special" 3-round HVZK PoK:



- Special HVZK:
  - Know c in advance $\Rightarrow$ can fake proofs for any y, even without knowing x
- Special Soundness:
  - Know two distinct conversations with same a $\Rightarrow$ recover witness x

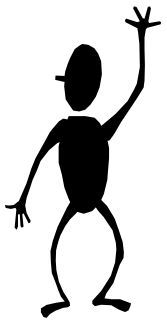
Proving Knowledge of DL (Representation)

input $y = g^x$

$a = g^r$

random c

$z = r - cx$



witness x

Accept iff $a = g^z y^c$

- Generalizes to proving knowledge of discrete log representation [Oka92]
- Resulting function becomes SPR !

- Special HVZK:
  - Know c in advance $\Rightarrow$ pick random z, and let $a = g^z y^c$
- Special soundness:
  - Know accepting $(a, c_1, z_1), (a, c_2, z_2) \Rightarrow a = g^{z_1} y^{c_1} = g^{z_2} y^{c_2}$

$$\Rightarrow x = (z_1 - z_2) / (c_2 - c_1)$$

ID Schemes from Sigma-protocols

- Assume Π is Σ-protocol for $y = f(x)$, where $|x| = n$, $|y| = k$

No Leakage	(L_1, L_2)-Leakage
Thm 1: f – OWF $\Rightarrow$ Π – passively secure ID scheme • simulate passive attack using y • rewinding extracts witness x'	**Thm 1'**: f – (L_1+2L_2)-LR-OWF $\Rightarrow$ Π – passively (L_1,L_2)-LR secure ID scheme • LR of f used to handle leakage • rewinding doubles "L_2-leakage"
Thm 2: f – SPRF & $n > k + \lambda$ $\Rightarrow$ Π – actively secure ID scheme • simulate active attack using x • Witness Indistinguishability (WI) $\Rightarrow$ no extra info about x leaked • rewinding extracts witness $x' \neq x$	**Thm 2'**: f – SPRF & $n > k+L_1+2L_2+\lambda$ $\Rightarrow$ Π– actively (L_1,L_2)-LR secure ID scheme • already what we need for leakage! • proof = hybrid of Thms 1' and 2

ID Schemes from Sigma-protocols

- Assume Π is Σ-protocol for $y = f(x)$, where $|x| = n$, $|y| = k$

No Leakage	(L_1, L_2)-Leakage
Thm 1: f – OWF $\Rightarrow$	**Thm 1'**: f – (L_1+2L_2)-LR-OWF $\Rightarrow$
Π – passi	
Thm 2: f	$L_2+\lambda$
Π – active	e

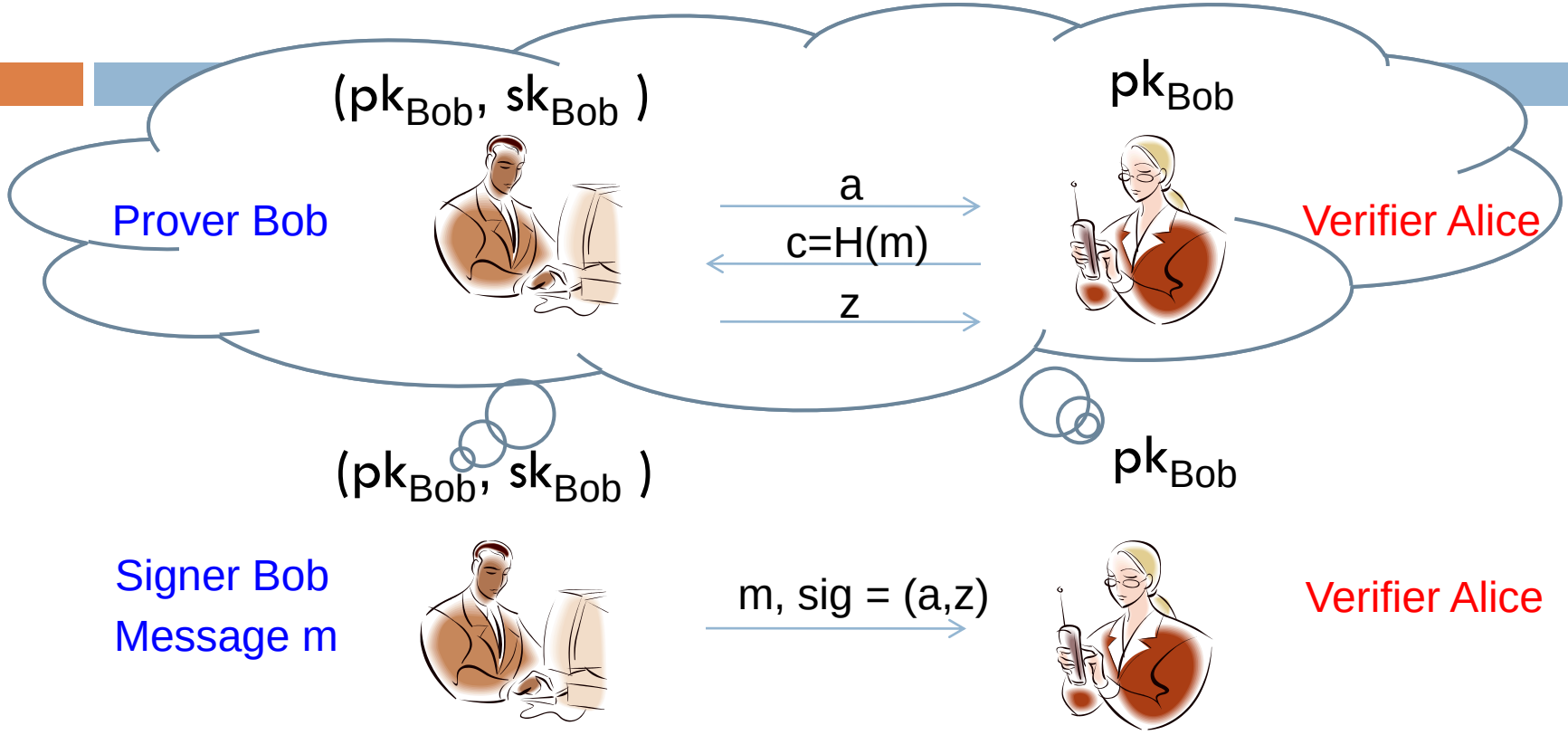
Bottom-Line for LR-ID Schemes:

- Pre-Impersonation Leakage $\approx$ n
- Anytime Leakage $\approx$ n/2

Roadmap of This Survey

- Relative Leakage Model
  - Password Authentication and OWFs
  - Identification Schemes
  - ☑ Signature Schemes
  - Encryption Schemes (and IBE)
  - Authenticated Key Agreement (AKA)
- Bounded Retrieval Model
  - From Relative to Absolute leakage

Fiat-Shamir: Signatures from ID



- 3 round (public-coin) passive ID scheme $\Rightarrow$ Signature.
  - Only works in the Random Oracle Model.

From ID to Signatures

- **Theorem**: Applying Fiat-Shamir to ID scheme with
  - Anytime Leakage $\Rightarrow$ Existentially Unforgeable Sig.
  - Pre-imperson. Leakage $\Rightarrow$ Entropically Unforgeable Sig.

Entropically Unforgeable Signatures:

(will be useful for later applications)

Adversary cannot forge signatures of random messages from any "high-entropy" distribution

(even after leakage)

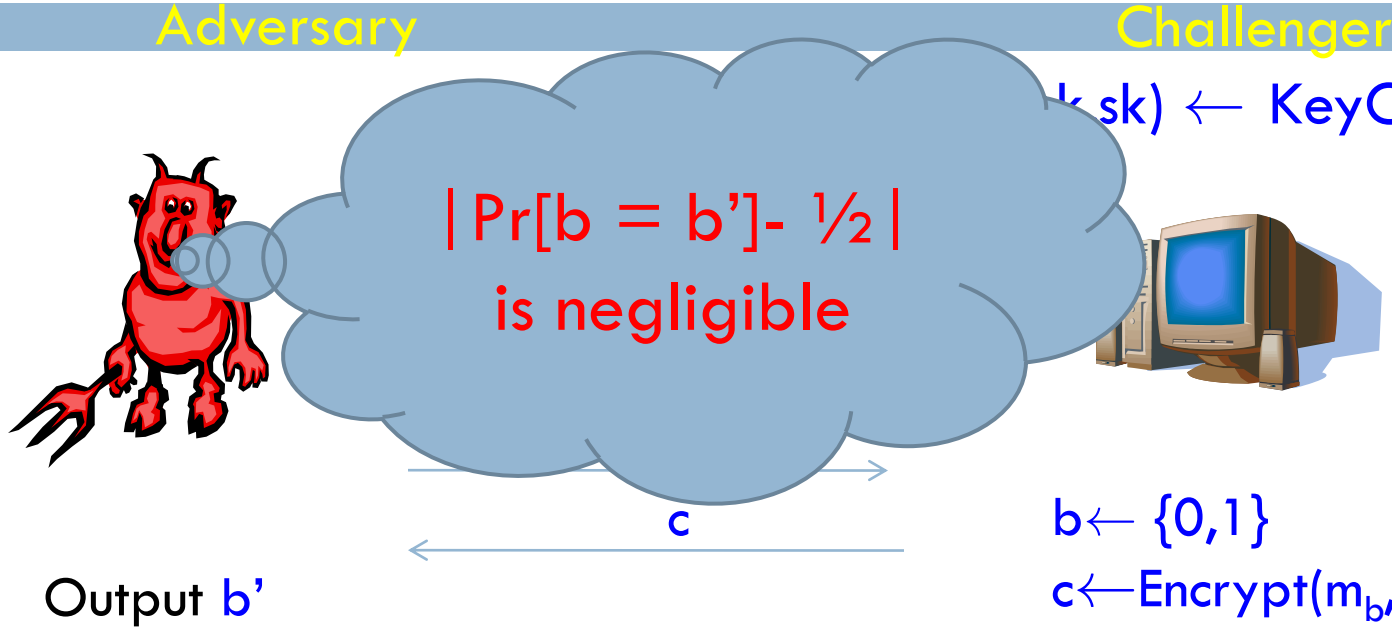
From ID to Signatures

- **Theorem**: Applying Fiat-Shamir to ID scheme with
  - Anytime Leakage $\Rightarrow$ Existentially Unforgeable Sig.
  - Pre-imperson. Leakage $\Rightarrow$ Entropically Unforgeable Sig.
- [ADW09]: Fiat-Shamir preserves leakage bound L, public/secret key sizes, communication, computation.
  - Existential UF with $L \approx |sk|/2$, Entropic UF with $L \approx |sk|$
- Standard model constructions, with $L \approx |sk|$?
  - [KV09]: Yes, based on generic SS-NIZK (inefficient)
  - [DHLW09]: Generalization + efficient instantiation

Roadmap of This Survey

- Relative Leakage Model
  - Password Authentication and OWFs
  - Identification Schemes
  - Signature Schemes
  - ☑ Encryption Schemes (and IBE)
  - Authenticated Key Agreement (AKA)
- Bounded Retrieval Model
  - From Relative to Absolute leakage

Definition of Leakage-Resilient PKE



- **Goal**: maximize L
- [NS09]: LR-PKE from Hash-Proof Systems (HPS) [CS02]
- [ADN+09]: Identity-based Hash-Proof Systems (ID-HPS)
  - Leads to Leakage-Resilient IBE (extending [AGV09])

Hash Proof Systems

- Simplified presentation as a Key-Encapsulation Mechanism:
  - $(pk, sk) \leftarrow KeyGen(1^{\lambda})$
  - $(c, k) \leftarrow Encap(pk)$
  - $k' \leftarrow Decap(c, sk)$
- Correctness: $k = k'$ (with overwhelming probability)
- KEM Security: $(pk, c, k) \approx_c (pk, c, \$)$
- HPS is a special way to prove KEM security:

Hash Proof Systems

- Simplified presentation as a Key-Encapsulation Mechanism:
  - $(pk, sk) \leftarrow KeyGen(1^\lambda)$
  - $(c, k) \leftarrow Encap(pk)$ (valid encapsulation)
  - $c^* \leftarrow Encap^*(pk)$ (invalid encapsulation)
  - $k' \leftarrow Decap(c, sk)$
- Correctness: $k = k'$ (with overwhelming probability)
- KEM Security: $(pk, c, k) \approx_c (pk, c, \$)$
- HPS is a special way to prove KEM security:
  - Replace KEM security by the following two properties…

Hash Proof Systems

- Simplified presentation as a Key-Encapsulation Mechanism:
  - $(pk, sk) \leftarrow KeyGen(1^\lambda)$
  - $(c, k) \leftarrow Encap(pk)$ (valid encapsulation)
  - $c^* \leftarrow Encap^*(pk)$ (invalid encapsulation)
  - $k' \leftarrow Decap(c, sk)$

Note: any smooth HPS with $k \in \{0,1\}^v$ can be composed with an extractor to get L-leakage smooth HPS with $L = v - \Omega(\lambda)$

- Decap(invalid ciphertext c^*) has statistical entropy:
  - Smoothness: for fixed pk, $(c^*, k^*) \approx_s (c^*, \$)$, where $k^* \leftarrow Decap(c^*, sk)$
  - L-Leakage-smoothness: $(c^*, k^*, g(sk)) \approx_s (c^*, \$, g(sk))$, where $|g(sk)| = L$

HPS $\Rightarrow$ Leakage-Resilient PKE [NS09]

- **Theorem** : A smooth HPS is a good KEM (standard).

  A L-leakage-smooth HPS is a L-leakage-resilient KEM:

  $(pk, g(sk), c, k) \approx_c (pk, g(sk), c, \$)$ where $(c, k) \leftarrow Encap(pk)$

- Proof:

  Correctness

  $(pk, g(sk), c, k) \approx_s (pk, g(sk), c, k')$ where $(c, k) \leftarrow Encap(pk)$

  $k' \leftarrow Decap(c, sk)$

  Valid/Invalid Indistinguishability

  $\approx_c (pk, g(sk), c^*, k')$ where $c^* \leftarrow Encap^*(pk)$

  $k' \leftarrow Decap(c^*, sk)$

  L-Leakage-Smoothness

  $\approx_s (pk, g(sk), c^*, \$)$

  Valid/Invalid Indistinguishability

  $\approx_c (pk, g(sk), c, \$)$

HPS Example Based on DDH

- Params: prime p, group G of order p, generators (g,h)

- KeyGen: $sk = (a,b)$ $pk = g^a h^b$
- Encap(pk): $c = (g^x, h^x)$ $k = (pk)^x = g^{ax}h^{bx}$
- Decap(c, sk): $k' = (g^x)^a(h^x)^b$
- Encap*(pk): $c^* = (g^x, h^y)$

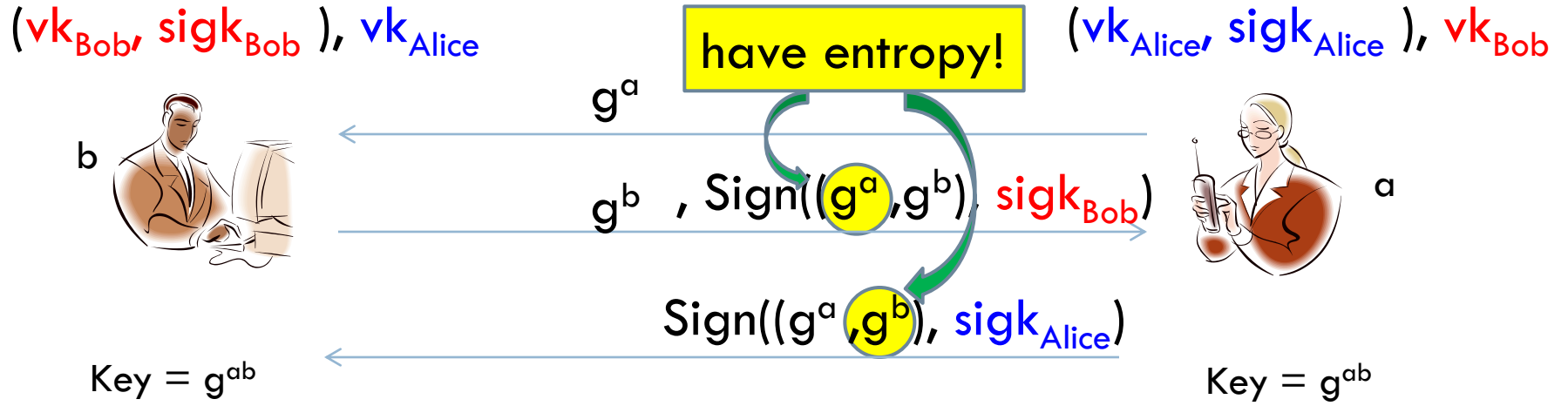
- Valid/Invalid Indistinguishability (given sk): follows from DDH
- Smooth: $Decap(c^*, sk) = g^{ax}h^{by}$ random given g^x, h^y and $pk = g^a h^b$
  - Need an extractor to get L-leakage-smoothness for $L \approx \log(p)$.

- Generalizes to $t > 2$ generators: $sk = (a_1,\ldots,a_t)$, $pk = \Pi_i (g_i)^{a_i}$
  - No extractor needed! L-Leakage-smooth for $L \approx (t-2)\log(p) = (1-2/t)|sk|$

Roadmap of This Survey

- Relative Leakage Model
  - Password Authentication and OWFs
  - Identification Schemes
  - Signature Schemes
  - Encryption Schemes (and IBE)
  - ☑ Authenticated Key Agreement (AKA)
- Bounded Retrieval Model
  - From Relative to Absolute leakage

Authenticated Key Agreement (AKA)



- Alice and Bob agree on shared **session-key**, secret from adversary
- Need: public-key infrastructure (e.g., signing/verification keys)
- Past session-key secure, even if adv. learns all signing keys in future
- LR-AKA: leakage of signing keys $\Rightarrow$ future session-keys secure
- [ADW09]: above protocol is LR-AKA, if use LR-Signatures
  - In fact, Entropic Unforgeability enough (important in BRM)
- [DHKW09]: new LR-AKA from LR-PKE

Roadmap of This Survey

- Relative Leakage Model
  - Password Authentication and OWFs
  - Identification Schemes
  - Signature Schemes
  - Encryption Schemes (and IBE)
  - Authenticated Key Agreement (AKA)
- Bounded Retrieval Model
  - ☑ From Relative to Absolute leakage

Bounded Retrieval Model

- Adversary can learn *any efficiently computable function* $g : \{0,1\}^* \rightarrow \{0,1\}^L$ of the secret key. L = Leakage Bound.
  - Increase sk size to allow L bits of leakage.
  - All other params don't depend on L!
- All existing BRM schemes built from relative-leakage scheme in 3 steps:
  1. Leakage Amplification (via Parallel-Repetition)
  2. Efficiency via Random-Subset Selection
  3. Adding a Master Public Key
- Steps 1. and 2. critically use information-theoretic techniques
- Simplest example: Password Authentication (PA)
  - See [ADW09, ADN+09] for ID, Sigs, Enc, IBE schemes

sk

leak

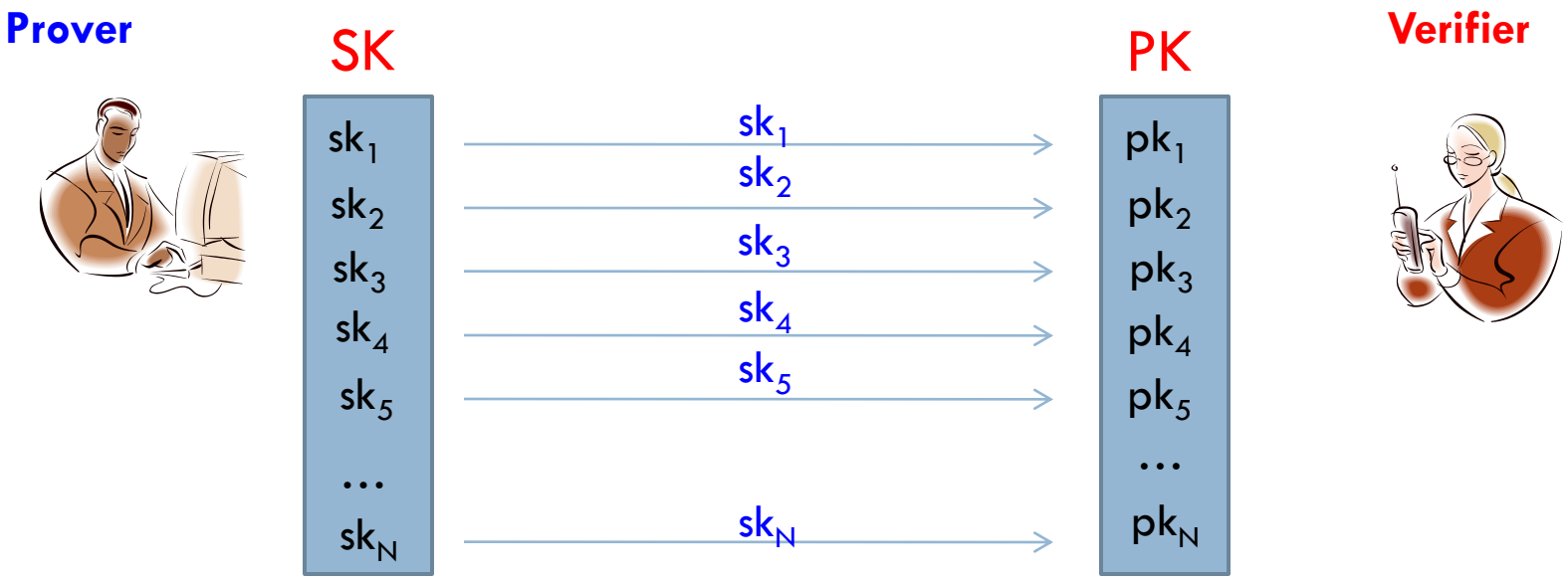
Template for BRM Schemes:
1. Leakage Amplification (via Parallel-Repetition)

- **Given**: scheme X resilient to L bits of leakage and L' > L
- **Goal**: construct scheme X' resilient to L' bits of leakage.
- Answer 1: Inflate security parameter λ until $L(\lambda) > L'$.
- Answer 2: Parallel-Repetition: run N independent copies of X
  - Choose N pairs $(pk_1,sk_1), \ldots, (pk_N,sk_N)$

    Set $PK = (pk_1,\ldots,pk_N)$, $SK = (sk_1,\ldots,sk_N)$
  - PA case: $pk_i = f(sk_i)$; to authenticate, send all N keys $sk_1,\ldots,sk_N$

Template for BRM Schemes:

1. Leakage Amplification (via Parallel-Repetition)



- **Intuition**: Scheme should tolerate L' = NL bits of leakage.
  - If leakage on SK is < NL bits then leakage on some sk_i is < L bits
  - **Wait!** How to reduce NL bit leakage of X' to L bit leakage of X?

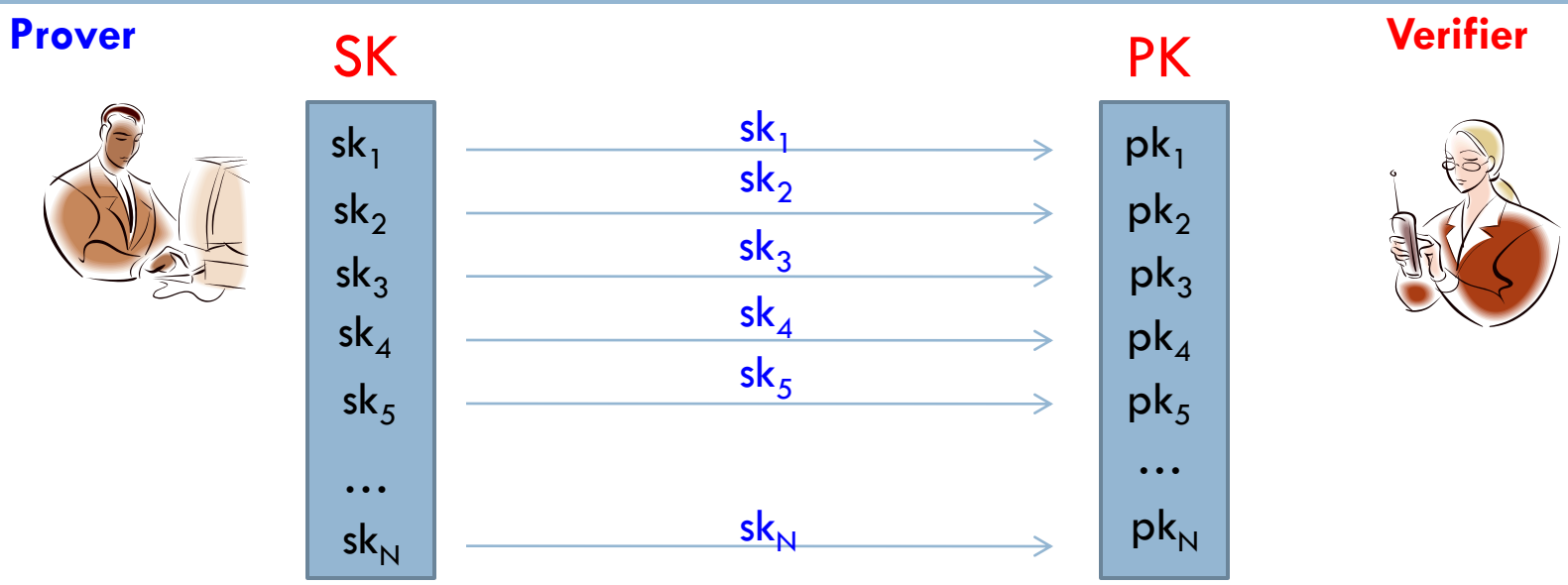
Template for BRM Schemes:
1. Leakage Amplification (via Parallel-Repetition)

- **Q: Does parallel-repetition amplify leakage-resilience?**
- **A1:** No *general* black-box reduction is possible ☹.
- **A2:** Works if original scheme has "extra properties".
  - Happens to be true for ID, Sig[illegible]9]
  - Interestingly, the extra-prope[illegible]!
- PA Case: let $F(x_1,\ldots,x_N) = (f(x_1),\ldots,f(x_N))$, wi[illegible] $f: n \to k$
  - If f is L-LR-OWF, cannot prove anything about F ☹
  - If f is SPRF from n to k ($\Rightarrow$ L-LR-OWF for $L \approx n-k$), then F is SPRF from Nn to Nk ($\Rightarrow$ L'-LR-OWF for $L' \approx Nn-Nk = NL$)

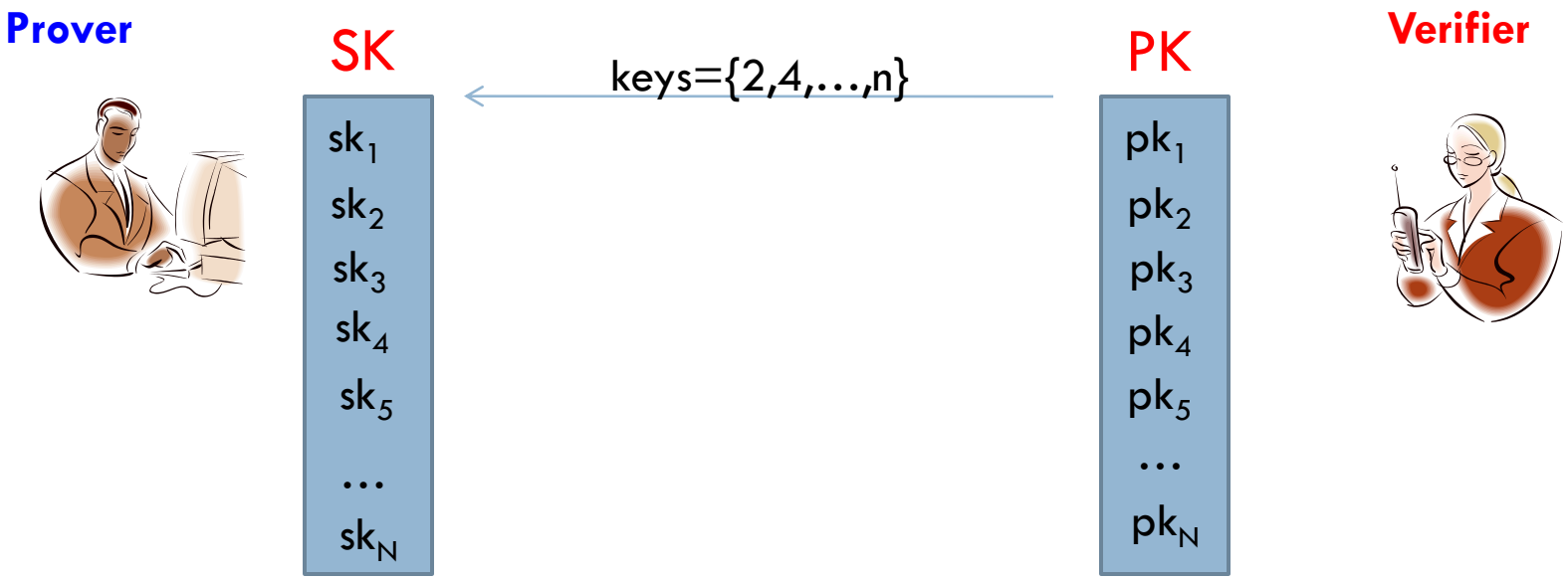
old leakage x N !

Template for BRM Schemes:

2. Efficiency via Random-Subset Selection

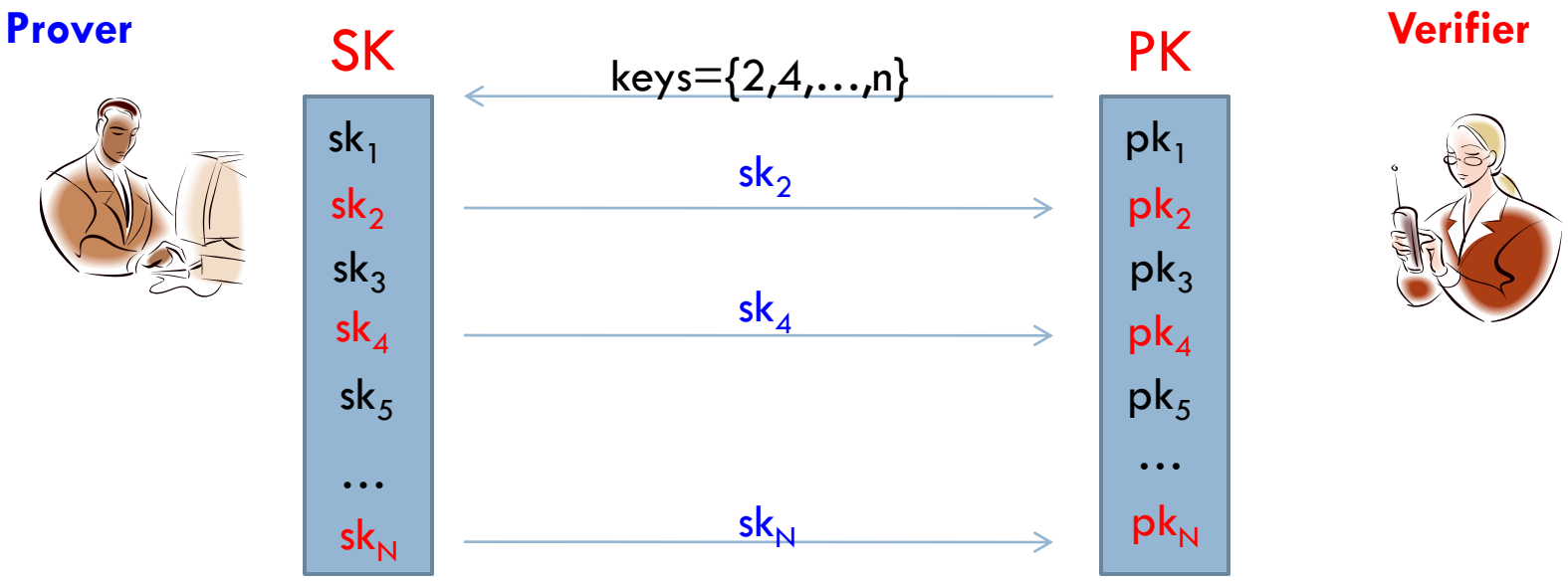


Template for BRM Schemes:
2. Efficiency via Random-Subset Selection



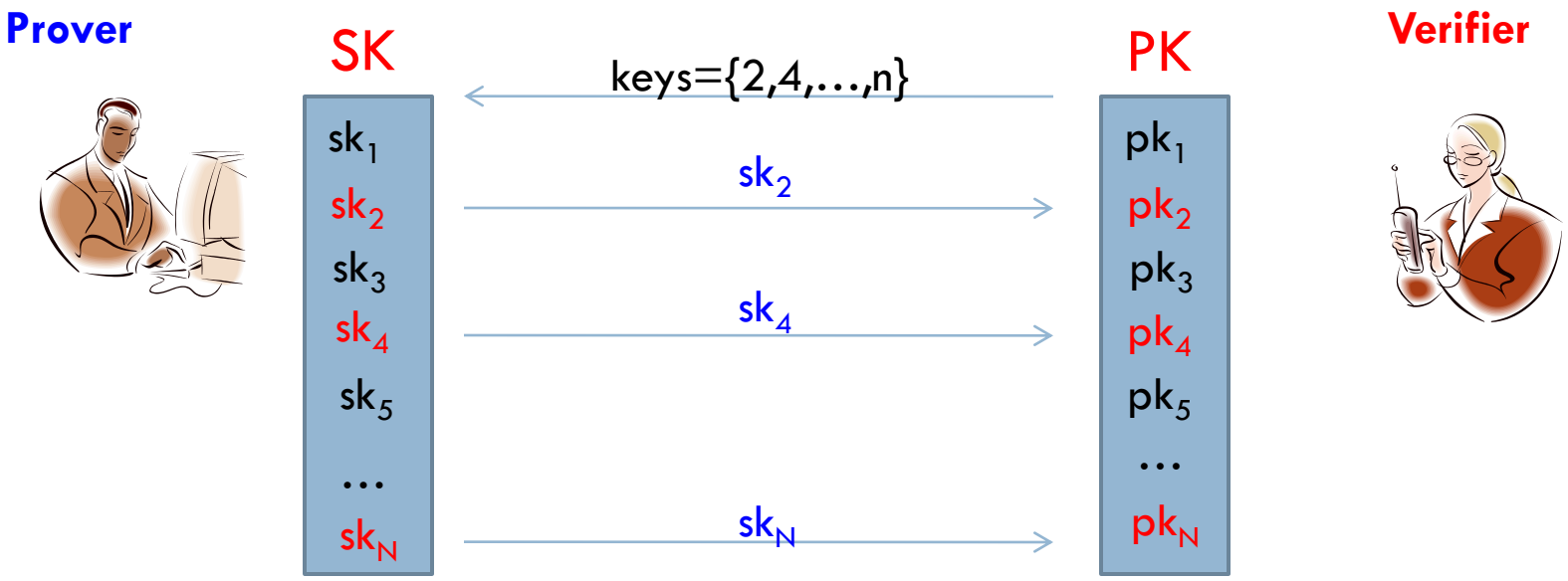
- Let Verifier choose $t=O(\lambda)$ random key-pairs and only use these

Template for BRM Schemes:
2. Efficiency via Random-Subset Selection



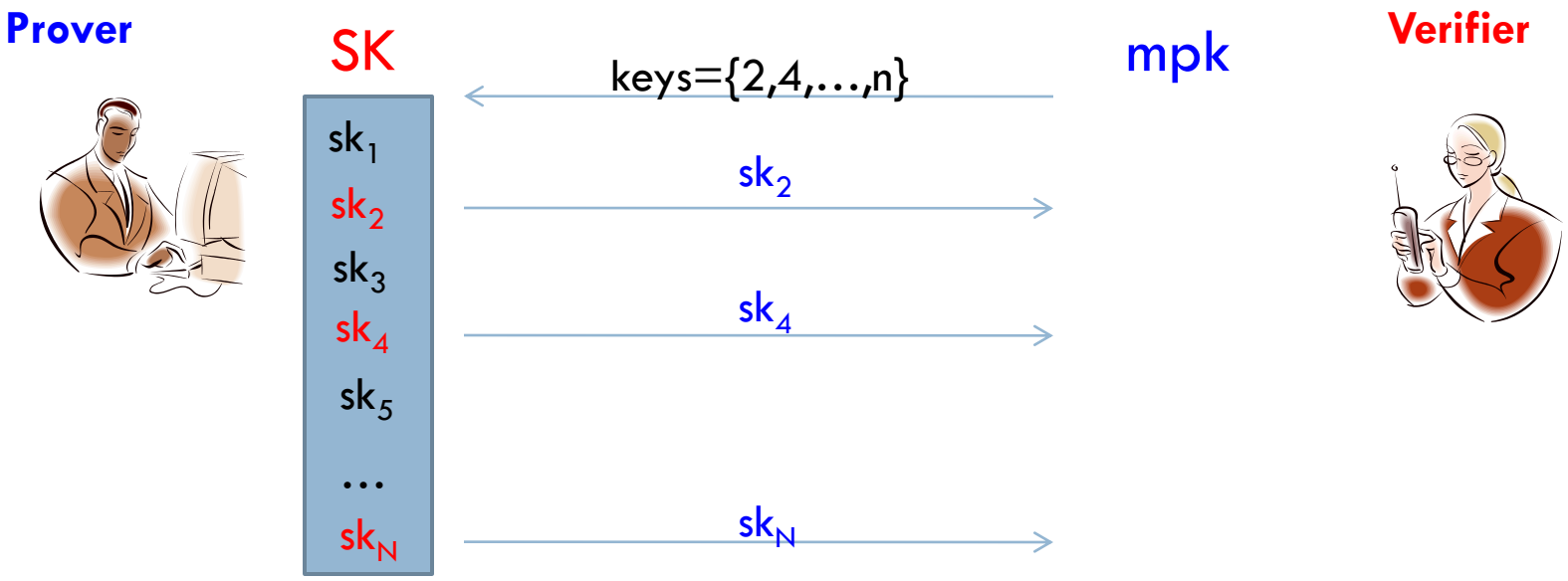
- Let Verifier choose $t=O(\lambda)$ random key-pairs and only use these
- Entropy Preservation Lemma: If Entropy(SK) given (PK, Leakage) is high, then Entropy($\{sk_i \mid i \in keys\}$) given (PK, Leakage, keys) is "high"
- Essentially reduces analysis to leakage-amplification

Template for BRM Schemes:
3. Adding a Master Public Key



- Last problem: $|PK| = O(N)$ still large ☹
- Use ID-based Techniques:
  - One short master public key mpk; view 1,…,N as "identities"

Template for BRM Schemes:
3. Adding a Master Public Key



- Last problem: |PK|=O(N) still large ☹
- Use ID-based Techniques:
  - One short master public key mpk; view 1,…,N as "identities"
- Authentication Applications: delegation by sigs [ADW09]
- Encryption Applications: IBE tools [ADN+09]

Summary

- Leakage-Resilient Crypto: primitives that provably allow leakage of secret key
  - Assume leakage is arbitrary but incomplete
- Relative Leakage vs. BRM
- Very active field, lots of work !
  - Many open questions too (e.g., efficiency)
- Information-Theoretic Tools used often

Thank You!

Questions?

LR-Friendly Reductions

information-theoretic techniques

- Set pk = SPRF(sk), where $|pk| \ll |sk| - L$
  - Many valid secret keys for a given public key pk
- Know <u>one</u> valid secret key sk in the reduction
  - Appears the only way to simulate the leakage oracle !
- Argue that adversary's A other legal queries do not give "too much" information about sk
  - Thus, Entropy(sk) given (pk, leakage, queries) still high
- Unpredictability Primitives (PA, OWF, ID, Sig, MAC, …):
  - Argue A must compute a valid secret key sk' to succeed
  - Break SPR since likely sk' ≠ sk
- Indistinguishability Primitives (encryption, IBE, AKA, …):
  - Design the "one-time pad" = Randomness Extractor(sk).