



Universiteit Twente  
*de ondernemende universiteit*

# Public Key Encryption with Prefix Keyword Search

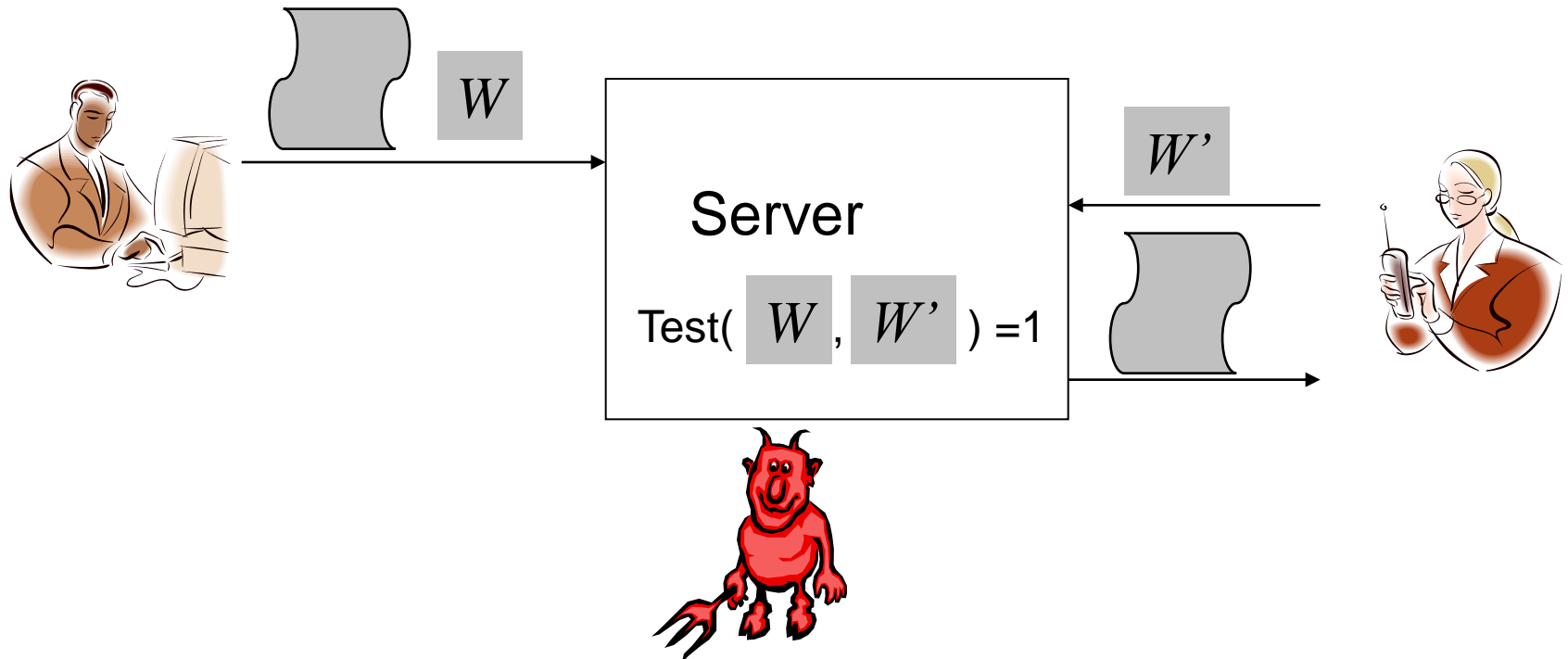
Saeed Sedghi  
SEDAN Workshop  
October 7 2009

# Outline

- Public key encryption with keyword search
- Prefix search
- Scheme
- Conclusion

# Public key encryption with keyword search

Dan Boneh, Giovanni Di Crescenzo, Rafail Ostrovsky, Giuseppe Persiano



# PEKS algorithms

- **Keygen**( $s$ ): Given a security parameter  $s$ :
  - Master secret key:  $msk$
  - Public parameters:  $param$
- **SearchableRepresentation**( $W, param$ ) :  $S_W$
- **Trapdoor**( $W', msk$ ):  $T_{W'}$
- **Test**( $T_{W'}, S_W$ ) = 1 if  $W = W'$

# Anonymous identity based encryption

- *Searchable Encryption Revisited: Consistency Properties, Relation to Anonymous IBE, and Extensions*

Abdallah et al



- Any anonymous identity based encryption (IBE) scheme can be used as a PEKS scheme.
- Anonymous identity based encryption: an IBE scheme which hides the identity of the receiver.
- Message to be encrypted is 1 and identity is replaced by keyword
- There are quite many PEKS schemes

# PEKS with Prefix Test

- In many practical cases client wants to perform a prefix keyword test
  - Example: retrieve encrypted documents that contain “take”, “takes” and “taken” via a trapdoor built by “take”.
- Existing PEKS schemes (Anonymous IBE) are capable of equality search.
  - **Keygen**( $s$ ): Given a security parameter  $s$ :
    - Master secret key:  $msk$
    - Public parameters:  $param$
  - **Searchable-Encryption**(“Takes”,  $param$ ) :  $S_{Takes}$
  - **Trapdoor**(“Take”,  $msk$ ):  $T_{Take}$
  - **Test**( $T_{Take}$ ,  $S_{Takes}$ )  $\neq 1$

# Why PEKS is suitable equality search only

$G$ : a multiplicative group of order  $q$

$g$ : group generator of group  $G$

$e: G \times G \longrightarrow G_T$

$e(g^x, g^y) = e(g, g)^{xy}$ ,  $(g^x, g^y) \in G^2$

$H_1: \{0,1\}^* \longrightarrow G$ ,  $H_2: G_T \longrightarrow \{0,1\}^p$

- Boneh et al PEKS:

**Keygen**( $s$ ):  $msk: a \in \mathbb{Z}_q$ ,  $param: (q, G, G_T, g^a), H_1, H_2, e(\dots)$

**Searchable-representation**( $w, g^a$ ):

$S_w = [g^r, H_2(e(g, H_1(w))^{ar})]$

**Trapdoor**( $w, a$ ):  $T_{w'} = H_1(w')^a$

**Test**( $T_{w'}, S_w$ ): output 1 if  $[g^r, H(e(T_{w'}, g^r))] = S_w$

# Solution Directions

- Trivial solutions:
  - Client sends trapdoor of all the possible keywords
  - Extend PEKS to a character based searchable encryption
- Using range queries on encrypted data techniques ( Hidden vector encryption)
  - Trapdoor is built for keyword  $W^*$
- Problem:
  - Not efficient: Decryption cost depends on #characters in trapdoor
  - Revealing #characters in trapdoor is revealed

# Prefix Keyword Search Scheme

# Preliminaries

$G$ : a multiplicative group of order  $n$

$n = pq$  for two large primes  $p$  and  $q$

$g$ : group generator of group  $G$

$e: G \times G \longrightarrow G_T$

- Decision Linear Diffie-Hellman problem: Given a tuple  $(g^{z_1}, g^{z_2}, g^{z_1 z_3}, g^{z_4}, Z)$  for random exponents  $(z_1, z_2, z_3, z_4) \in \mathbb{Z}_p$  it is hard to distinguish between  $Z = g^{z_2(z_4 - z_3)}$  and a random  $Z \in G$ .

# Prefix search without random oracle

- **keygen**( $s$ ):  $msk = (\alpha, p, q)$   $p$  is order of group  $G$
- Pick  $(u, u_1, \dots, u_L) \in Z_n$ . Pick  $\alpha \in Z_n$

$$Pk = ((g^u, g^{\frac{p}{q^L} u_1}, \dots, g^{\frac{p}{q^{L-(i-1)}} u_i}, \dots, g^{\frac{p}{q} u_L}), g^\alpha, \mathbb{G}_1, \mathbb{G}_2, e(\cdot, \cdot))$$

- **Searchable-Representation**( $W, pk$ ):  $W = (w_1, \dots, w_l)$
- Pick  $(r_1, r_2) \in Z_n$

$$S_W = (g^{\alpha r_1}, g^{r_2}, (g^u \prod_{i=1}^l g^{\frac{p}{q^{L-(i-1)}} u_i w_i})^{r_1+r_2})$$

- **Trapdoor**( $W', \alpha$ ):  $W' = (w'_1, \dots, w'_m)$ , Pick  $s \in Z_n$

$$T_{W'} = ((g^u \prod_{i=1}^m g^{\frac{p}{q^{m-i}} u_i w'_i})^s, (g^u \prod_{i=1}^m g^{\frac{p}{q^{m-i}} u_i w'_i})^{\alpha(s)}, g^{\alpha q^{L-(m-1)} s})$$

- **Test**( $T_{W'}, S_W$ ): Let  $S_W = (C_1, C_2, C_3)$ . Let  $T_{W'} = (T_1, T_2, T_3)$ .  
check if:  $e(C_1, T_1)e(C_2, T_2) = e(C_3, T_3)$

# Correctness

$$e(C_1, T_1) = e(g^{\alpha r_1}, (g^u \prod_{i=1}^m g^{\frac{p}{q^{m-i}} u_i w'_i})^s) = e(g, g)^{\alpha r_1 u(s)} \prod_{i=1}^m e(g, g)^{\alpha r_1 \frac{p}{q^{m-i}} u_i w'_i(s)}$$

$$e(C_2, T_2) = e(g^{r_2}, (g^u \prod_{i=1}^m g^{\frac{p}{q^{m-i}} u_i w'_i})^{\alpha(s)}) = e(g, g)^{\alpha r_2 u(s)} \prod_{i=1}^m e(g, g)^{\alpha r_2 \frac{p}{q^{m-i}} u_i w'_i(s)}$$

$$e(C_1, T_1) e(C_2, T_2) = e(g, g)^{\alpha u (r_1 + r_2)(s)} \prod_{i=1}^m e(g, g)^{\alpha (r_1 + r_2) \frac{p}{q^{m-i}} u_i w'_i(s)}$$

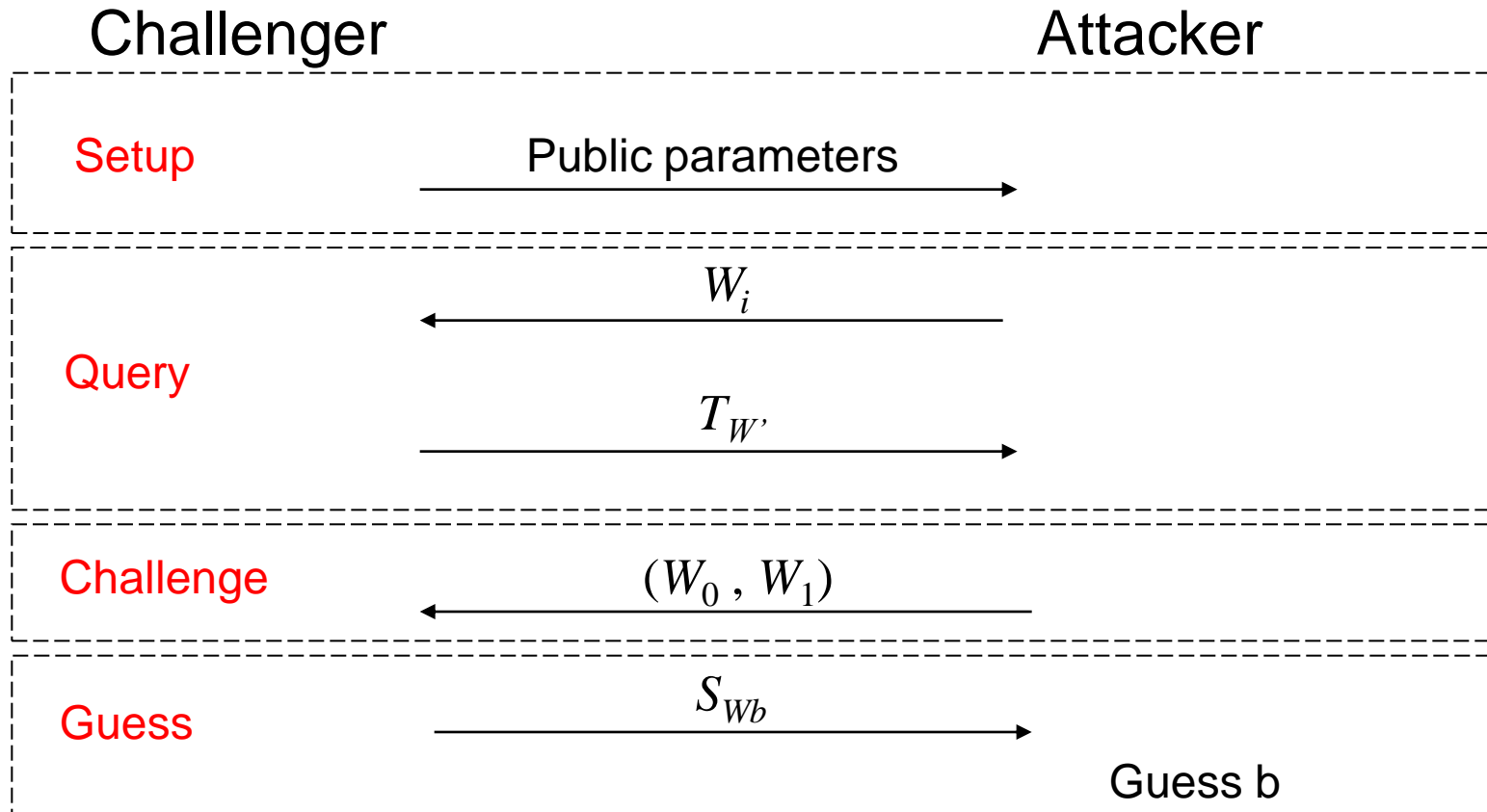
$$e(C_3, T_3) = e(g, g)^{u \alpha(s) (r_1 + r_2)} \prod_{i=1}^l e(g, g)^{\frac{p}{q^{L-i-1}} q^{L-(m-1)} (r_1 + r_2)(s) u_i w_i} =$$

$$e(g, g)^{u (r_1 + r_2)(s)} \prod_{i=1}^m e(g, g)^{\frac{p}{q^{m-i}} (r_1 + r_2)(s)} \prod_{i=m+1}^l e(g, g)^{p q^{i-m} (r_1 + r_2)(s) u_i w_i}$$

Since  $e(g, g)^{pqx} = e(g, g)^{nx} = 1$  for any integer  $x$ :

$$e(C_3, T_3) = e(g, g)^{\alpha u (r_1 + r_2)(s)} \prod_{i=1}^m e(g, g)^{\alpha \frac{p}{q^{m-i}} (r_1 + r_2)(s)}$$

# IND-CPA Security



If  $\Pr[b=b'] = \frac{1}{2} + \varepsilon$ , scheme is IND-CPA secure

# Security analysis

- The searchable representation:  $(g^{\alpha r_1}, g^{r_2}, (g^u \prod_{i=1}^l g^{\frac{p}{q^{L-(i-1)}} u_i w_i})^{r_1+r_2})$

is indistinguishable from  $(g^{\alpha r_1}, g^{r_2}, Z)$ ,  $Z$  is random from  $G$

- Decision Linear Diffie-Hellman problem: Given a tuple  $(g^{z_1}, g^{z_2}, g^{z_1 z_3}, g^{z_4}, Z)$  for random exponents  $(z_1, z_2, z_3, z_4) \in \mathbb{Z}_p$  it is hard to distinguish between  $Z = g^{z_2(z_4 - z_3)}$  and a random  $Z \in G$ .

# Conclusion

	Cipher-text cost	Trapdoor Cost	Search cost	Revealing # letters in trapdoor	Capability
Trivially extended Anonymous IBE and PEKS	$O(l)$	$O(m)$	$O(m)$	Yes	Prefix Search
Waters range query	$O(l)$	$O(m)$	$O(m)$	Yes	Subset, range, comparison query
Dimensional range query	$O(l)$	$O(m)$	$O(m)$	Yes	Subset, range comparison query
Our scheme	$O(l)$	$O(m)$	$O(1)$	No	Prefix Search

Questions?